UiT The Arctic University of Norway

Faculty of Engineering Science and Technology
Department of Computer Science and Computational Engineering

Clustering of clinical multivariate time-series utilizing recent advances in machine-learning

Asal Asgari

DTE-3900 Master's thesis in Applied Computer Science May 2023

Abstract

The purpose of this thesis is to set the groundwork for future research on developing a machine-learning based anomaly detection system for hospitalized patients. Our first step was to study and analyze the project's needs, background, and literature examining similar criteria. In the second step, we interviewed medical experts and researchers. Based on our research and the suggestions received in our interviews, we explored methods that could be utilized to approach the issue based on the data we collected. The results of these approaches were then discussed.

According to the results, the K-means algorithm, which utilizes principle components to cluster, obtained the highest quality. We then discussed how other algorithms have been influenced more by the shape of the data than by the values of the data. Afterward, we made some suggestions about how this research could be approached in the future as we move forward.

Acknowledgements

I would first like to thank my supervisors Helge Fredriksen and Bernt Arild Bremdal for their guidance, feedback, and resources.

I am also grateful to my parents for supporting me and making it possible for me to move to Norway and pursue higher education.

Lastly, I am thankful to my classmates Håkon Berg Borhaug, Johanne Holst Klæboe and Joachim Kristensen for all their help during my master's.

Contents

List of Figures

List of Tables

1 Introduction

This chapter discusses the context of the thesis, the objectives, and an overview of the project, as well as some of its underlying fields and disciplines. Later in the section, we discuss the existing technology and the state of the art.

1.1 Background

The necessity of maintaining good health is paramount to leading a fulfilling life. Despite this, illnesses and other health problems can strike abruptly and without warning, making it imperative that you recognize them and take immediate action. The earlier a sickness is detected, the more likely it is to be treated, preventing an illness from becoming worse and lowering the probability that new complications will occur (1). Consequently, a uniform method of early detection would greatly improve efficiency and allow the health sector to address the issue in a timely manner.

In 2012 The Royal College of Physicians addressed this issue by developing a standardized and uniform early warning system to assess patient illnesses and deterioration by assigning a score to the routinely recorded physiological parameters in hospitalized patients (2). The National Early Warning Score (NEWS) is measured by evaluating six physiological parameters where each parameter is assigned a score based on the deviation from the normal range for that parameter. The patient's severity can be determined based on the aggre-

gated scores from these multiple physiological parameters, and the appropriate action can be taken depending on the identified level of risk (3). The vital signs measured by NEWS typically include:

- Respiratory rate
- Oxygen saturation
- Pulse rate
- Blood pressure
- Temperature
- Level of consciousness

It is recommended by the Norwegian Directorate of Health that all hospitals in Norway should implement NEWS as part of their routine clinical practice. While the recommendation was not mandatory, it was widely adopted and many hospitals in Norway implemented NEWS.

1.2 Relevance

In direct engagement with medical experts, they pointed out that despite the good intentions NEWS does not fulfill all of their needs, and we discussed why this project is significant for them and why they need an upgraded system even though they are already implementing NEWS.

They explained that while NEWS has been widely adopted across many healthcare systems, it is not without its limitations. They cited a number of issues with the existing system, including excessive noise notifications and an added workload for doctors and nurses. For instance, the current system does not take into account the context of the patient's condition, as vital signs can change based on factors such as movement or position which can lead to false alarms or unnecessary notifications.

Moreover, different individuals may have different normal ranges for vital signs depending on their health status and medical history. For instance, a healthy individual would typically have an oxygen saturation level above 96, but for someone with COPD (Chronic Obstructive Pulmonary Disease), an oxygen level around 90 may be considered normal, while a level below 75 is alarming. Therefore, determining whether a particular vital sign reading is dangerous for a person depends on their individual situation, including factors such as age and existing medical conditions which again contracts with the concept of having a unified scoring model such as NEWS for every patient.

Additionally, when a patient is under control for a particular condition, doctors take action toward solving the problem. As that care might take time to improve the patient's condition, the system may send alerts every few hours to nurses who have already taken the right action and are aware of the patient's situation, resulting in "noise" notifications that can be a distraction.

In addition, NEWS is trained on end results and predicts if a patient needs to go to the Intensive Care Unit or is likely to die within 24 hours (3). However, this may not provide much additional information to doctors who are already checking in with patients daily and are aware of their condition. As a result, it was decided to seek help from a computer science approach to improve the system's effectiveness.

1.3 Objective

We propose an investigation toward the development of an anomaly detection system for hospitalized patients on previously recorded time series of vital signs and patient characteristics. This study will examine whether it is possible to capture characteristic patient profiles based on a variety of clinical conditions that can be verified, and this work will be the basis for future research into an anomaly detection system. We have been provided datasets by Nordlandssykehuset which consist of a set of multivariate, irregularly sampled time series of variable lengths. The dataset contains measurements such as Systolic and Diastolic blood pressure, temperature, respiration rate, oxygen saturation, and pulse. We also have access to the gender and the level of consciousness of each patient and the ward that they are hospitalized in. The dataset lists 6 types of consciousness levels that patients could experience:

- Våken (Awake)
- Reagerer på tiltale (Responds to charges)
- Reagerer ved smertestimulering (Responds to pain stimulation)
- Nyoppstått forvirring (Emerging confusion)
- Reagerer ikke på tale eller smertestimulering (Does not respond to speech or pain stimulation)
- Våken og orientert (Awake and oriented)

The clinical data we are working with has a few characteristics which make it challenging to train a machine-learning model. As an example, we have 6 dimensions or physiological measurements with every registration. This means that our training process is more complex due to the multivariate nature of our

data.

Vitals are normally recorded every 8 hours; however, when NEWS detects patients at higher risk, they are recorded more frequently. Therefore, the rate at which measurements are taken varies based on the patient's level of risk, so patients are not equally spaced in their registrations.

Additionally, we do not have an equal number of registrations for each patient. Number of sample registrations spans from 1 to 91, which could also contribute to sample selection bias (4).

Data are also subject to varying levels of uncertainty, including errors introduced by clinical staff and technicians during manual recording, measurement noise, and system failures.

As a result of our interview with medical experts, we gained a better understanding of the motivation behind the project. We continued our process by examining the theoretical components and reviewing the relevant literature. Literature review aims to gain insight into how relevant papers use tools and methods that may be useful to us.

1.4 The State of the Art

Multivariate time-series data analysis is a challenging and active research area with growing literature. However, the unique characteristics of real-world datasets, such as irregularly spaced data, multivariate time series with unequal lengths, and missing data, have made the analysis of such data even more challenging. In this section, we review some of the recent research work that addresses the challenges of analyzing such datasets.

We drew inspiration from other papers that tackled similar problems to our own research, such as the "Unsupervised Pattern Discovery in Electronic Health Care Data Using Probabilistic Clustering Models" paper (5). While the authors of this paper dealt with sparse, high-dimensional, and noisy data collected from a pediatric intensive care unit over 10 years, they only retained the first 24 hours of admission for each patient. However, our data differ in that they had access to almost all measurements taken within 24 hours of admission, with data points spaced every hour. The authors of the paper present a probabilistic clustering model that employs an empirical prior based on a similarity kernel, which encourages the mean parameter of each cluster to be smooth over time.

Another paper introduced a method for classifying multivariate time series

using dynamic time warping (DTW), specifically for human activity recognition using sensor data (6). The authors developed a template selection algorithm that assessed the discriminative ability of each potential template, selecting the most informative ones for use in the DTW algorithm. They showed that this method enhances DTW-based classifiers' performance on multivariate time series data.

Missing data is a common issue in many datasets, including multivariate time series data. "A Gentle Introduction to Imputation of Missing Values" is a review paper that provides an overview of the methods used to handle missing data in statistical analysis (7). In their discussion, the authors review various imputation methods and how imputation can be used to deal with common types of missing data, such as missing at random (MAR), missing completely at random (MCAR), and missing not at random (MNAR).

"Application of Deep Interpolation Network for Clustering of Physiologic Time Series" (8) discussed a similar situation to ours with sparse and irregularly sampled time series data on physiological variables. A deep interpolation network was proposed to extract latent representations from time-series vital signs measured within six hours of hospitalization which estimates missing values through the fitting of cubic polynomial curves. These interpolated time points were then clustered with K-means, and they had access to labels for the diagnosis of patients, which helped them prove the quality of the clusters.

Another article with a similar focus to our thesis explores the analysis of trajectory similarities in time series. The authors propose an indexing method that consists of a two-level index structure (9). The first level index is based on clustering, while the second level index is based on distance measure selection. This indexing method accommodates multiple distance measures, such as Longest Common Subsequence (LCSS), Dynamic Time Warping (DTW), and Euclidean distance. The authors' primary contribution is the ability to support all of these measures without having to restructure the index, making their approach more efficient and practical.

In "Reservoir Computing Approaches for Representation and Classification of Multivariate Time Series," the authors propose a Reservoir Computing classifier based on the Reservoir model-space representation for the classification of multivariate time series (10). The challenges in this paper include the high dimensionality of the data, the complex temporal dependencies between variables, and the need for real-time processing.

Another paper introduces a kernel method called Time Series Cluster Kernel (TCK), for computing similarity between multivariate time series that handles missing data without resorting to imputation (11). Missing data and incomplete

labels, common in medical time series, are accommodated using a probabilistic model, improving clustering accuracy.

1.5 Strategy

We conducted a literature review, interviewed some of the authors of these papers, and based on the advice we received, we concluded that our first step would be to preprocess the data by calculating the minimum, maximum, standard deviation, and mean for each patient's time series, and then we utilized a dimension reduction method such as principal component analysis to retain the most important dimensions in the datasets and to understand if there are any distinct patterns in this feature space.

Analyzing the pre-analysis results, we looked at methods for approaching our project. We found that in studies that encountered similar problems working in this setting, either supervised (10) or unsupervised learning utilizing clustering techniques (5; 8) approaches were used. Next chapter, we will mention that supervised learning requires labels to train a model. Since we don't have access to labels, we chose to pursue unsupervised learning instead. Taking inspiration from the literature review papers, we will explore imputation on our data and dynamic time warping as well as other similar distance metrics and kernels for clustering multivariate series. Different clustering algorithms, such as K-means and hierarchical clustering, will be explored to determine if they can be used with our data type and if they identify distinct clusters, as well as whether patient characteristics affect clustering.

1.6 Theory

The purpose of this section is to define some of the methods that have been discussed in the previous section as well as the methods that we intend to use for our study.

1.6.1 Machine Learning

The development of machine learning can be traced back to 1950 where Alan Turing proposed to consider the question "Can machines think?" (12). Turing introduces the concept of the Turing Test, which proposes that a machine can be considered intelligent if it can successfully convince a human interrogator that it is human through written conversation (13). In 1959, Arthur Samuel

defined machine learning as, "Field of study that gives computers the ability to learn without being explicitly programmed". He introduced a pioneering approach to machine learning through creating a program that could learn to play the game of checkers againts itself and improve its strategy based on the outcomes of those games (14).

Essentially, machine learning is a subset of artificial intelligence that enables computers to emulate human intelligence by learning from examples, surrounding environments, and past experiences, such as playing checkers or chess (14), pattern recognition (15), or medical applications (16).

There are four main categories of machine learning based on the type of data labeling: Supervised, Unsupervised, Semi-supervised and Reinforcement learning (13). Supervised learning involves training algorithms on labeled data, where the desired output for each input is known. Unsupervised learning involves discovering patterns and structures in unlabeled data (17). A semi-supervised method uses unlabeled data to augment labeled data in a supervised learning paradigm (18) and Reinforcement learning is a type of machine learning where an agent learns to make decisions in an environment by interacting with it and receiving feedback in the form of rewards or punishments (19).

1.6.2 Time-series Clustering

Clustering is a machine learning technique used to place data elements into related groups without advanced knowledge of the group definitions. Each cluster consists of objects that are similar between themselves and dissimilar to objects of other groups (20). A special type of clustering is time-series clustering. A sequence of continuous, real-valued elements containing explicit information related to timing is called a time-series (21). In essence, a time series is dynamic data since it changes in values as a function of time, so the value(s) of each point are observations made in chronological order (22). Clustering of time-series data is used to either discover interesting patterns in the dataset (23) or to find outliers to help make predictions and recommendation (22) (24).

K-means

K-means (25) is a simple and effective unsupervised clustering algorithm that is easy to understand and implement. K-means assigns data points iteratively to the closest cluster centers, then recalculates the cluster centers based on the updated assignment. The algorithm is initially set up by placing centroids randomly, labeling the instances, and updating the centroids iteratively until the centroids stop moving. A given dataset can be effectively divided into K

clusters based on its similarity. The algorithm converges in a finite number of steps which usually are quite small (26).

Hierarchical clustering

Hierarchical clustering is a method for recursively dividing a dataset into smaller clusters (27). The data points are divided into individual clusters and, at each step, the closest clusters are merged. This is shown by a tree-like diagram called a dendrogram, with each leaf representing a data point and each internal node representing a cluster containing its descendant leaves and the order in which the clusters were merged. Arrangement of the clusters can be used to determine which leaves are most similar to each other, while height of the branch points can indicate their differences: the taller the branch point, the greater the distance/difference between the two (28).

When using hierarchical clustering, each object starts as a single cluster, and then using a linkage method, they are merged together. The linkage methods work by calculating the distances or similarities between all objects. There are six different linkage methods that are used for forming these clusters, including single, complete, average, weighted, median, centroid and ward. Single methods combine cluster entries with the smallest minimum distance between them. In spite of single methods having a faster computation time, it performs poorly if noise is present (29). The complete linkage method merges two clusters with the shortest maximum distance between their members; however, it does not perform well in cases of outliers (29), whereas a centroid method merges two clusters with the shortest centroid distance between their members. Ward's linkage involves merging two clusters by calculating their error sum of squares (ESS) and the Average linkage method uses the lowest average distances for merging the clusters (30). Unlike average linkage, weighted average considers inter-cluster distances inversely proportional to the number of objects in each class. The distance between two clusters in the median method is defined as the median distance between all pairs of points in the two clusters.

DBSCAN

DBSCAN (Density Based Spatial Clustering of Applications with Noise) (31) is a clustering algorithm introduced in 1996 as a tool for handling large noisy datasets for which there is not much knowledge of the domain to determine parameters. An instance is considered a core instance if it has at least a defined number of minimum sampled instances within an Epsilon distanced neighborhood. Every instance in a core instance's neighborhood belongs to the same cluster. If an instance is not a core instance in its neighborhood and does not

have one in its neighborhood, then it is considered anomalous (26).

1.6.3 Clustering Evaluation Metrics

Clustering evaluation metrics can be used for many reasons, including determining the correct number of clusters, assessing how well the results of a cluster analysis fit the data, and determining that the cluster structure is non-random (28). Two aspects can be used in the unsupervised evaluation of cluster quality: cluster cohesion, which assesses how closely related objects within a cluster are, and cluster separation, which measures how distinct a cluster is from others (28).

Among the many evaluation techniques, two metrics are used in this thesis:

Silhouette Score

Silhouette scoring begins with calculating the average distance between an object in a cluster and every other object in the cluster(a). Then we calculate the average distance between the mentioned object and each other object in the other clusters that do not contain our object(b). The silhouette coefficient for the object is $s = \frac{b-a}{max(a,b)}$ (28).

A cluster's silhouette score is the average of its silhouette coefficients. Silhouette score lies between -1 and 1, the closer the value is to 1, the better matched the data points are to their own clusters (28).

Davies-Bouldin Index

Davies-Bouldin Index(DBI) (34) is calculated as the average of each cluster's similarity score with its most similar cluster. Each cluster is given a DBI score, which is averaged over all clusters. The Davies-Bouldin index is defined as:

$$DBI = \frac{1}{k}\sum_{i=1}^{k} max(R_{ij})_{i\neq j} \tag{1.1}$$

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \tag{1.2}$$

where s_i and s_j is the average distance between each point of cluster i and j and the centroid of that cluster and d_{ij} is the the distance between cluster centroids i and j (35).

A lower value indicates better clustering, with zero being the minimum score (36).

1.6.4 Similarity Measures

The process of determining the degree of similarity between different time series and identifying the most similar time sequences is fundamental to time series data mining (37). Similarity measures are typically used to compare two time series and determine how much they resemble each other. This could be based on their shape, pattern or behavior (38).

Time-series applications use a variety of similarity measures, and we describe some of them in this thesis.

Dynamic Time Warping

Dynamic Time Warping (DTW) (39) is a well known algorithm that was proposed around 1970 in the context of speech recognition and aligning two speech patterns spoken at different rates. This algorithm is for finding similarity and optimal alignment between two time dependent sequences that may vary in time and speed (40).

DTW operates by constructing a matrix of accumulated distances between the two time series. Each element of this matrix represents the distance between two points i and j in respectively in timeseries A and B, with the distance between points i and j given by d(i,j) (41). DTW works by finding the optimal alignment between two time series by warping their time axes and by minimizing the cumulative distance along a path through the distance matrix (40). A DTW is considered an elastic measure that can be applied to applications involving time shifts and different lengths (42).

Soft-DTW

Soft-DTW (43) is a differentiable approximation to DTW that can be optimized efficiently using gradient-based methods. DTW is usually computed using dynamic programming as a method of minimizing the alignment cost of two time series. Soft DTW, unlike traditional DTW, finds a distribution of paths instead of finding an optimal path through the warping space. This makes it more robust to noise and able to capture alignment uncertainty.

Global Kernel Alignment

Global Kernel Alignment is a similarity measure that elaborates on soft-DTW for comparing time-series (42). The advantage of global kernel alignment is that it can capture the overall similarity between two sequences, even if they have different lengths or contain different patterns. Global kernel Alignment is connected to DTW through this formula:

$$k(x, y) = exp(-\frac{softDTW_{\gamma}(x, y)}{\gamma}) \tag{1.3}$$

where γ is the hyper-parameter controlling softDTW smoothness (36).

Euclidean

The Euclidean distance is a measure of the distance between a pair of samples in an n dimensional space where $d(q, p) = \|q - p\| = \sqrt{(\sum_{i=0}^{n}(q_i - p_i)^2)}$ (44).

1.6.5 Dimensionality Reduction

There are many Machine Learning problems that have a large number of features which could lead to a complex model and make the training process much longer than necessary. Creating a low-dimensional representation of a high-dimensional dataset is known as dimensionality reduction (45). Apart from speeding up training, dimensionality reduction is also extremely useful for data visualization, since it allows a high-dimensional training set to be visualized in a condensed way, allowing one to visually identify patterns, such as clusters, and often gain valuable insights (26).

Principle Component Analysis

Principle Component Analysis (PCA) was first introduced in 1901 where Pearson was looking for an algorithm to summarize and analyze the patterns in complex dataset (46). As a machine learning method, PCA has essentially the same objective: reduce dimensionality while preserving as many variation characteristics as possible. The process of reducing dimensionality in PCA does not require any labels, thus making it considered an unsupervised method. A reduction is accomplished by transforming the data into an additional set of variables, the principal components. These variables are uncorrelated and sorted so that the first few retain most of their original variance (47).

t-Distributed Stochastic Neighbor Embedding

The t-Distributed Stochastic Neighbor Embedding (t-SNE) (48) technique is a dimensionality reduction technique ideally suited to visualization of high-dimensional datasets (26). To map high-dimensional data points to low-dimensional space while maintaining pairwise similarities, a probabilistic approach using a Gaussian kernel is applied in this method (48). As a result, a low-dimensional embedding of the data points is created that preserves their pairwise similarities using gradient descent to minimize divergence between the two probability distributions (49).

2

Tools and Methods

In this chapter, we describe the tools and methods we used for the project. First, we discuss the tools and libraries we used for the project. The next section describes the processing steps, datasets, and implementation details, as well as the decision-making process.

2.1 Tools

This project was developed using Python (50) version 3.7. Several different software libraries were investigated in order to find the optimal clustering algorithm, such as Scikit-learn (35), TSlearn (36), and SciPy (30). We used Python Numpy (51) and Pandas (52) for analyzing data, Plotly (53) and Matplotlib (54) for visualizing, and Yellowbrick (55) to calculate metrics for clustering and visualization.

2.1.1 Tool Selection

There are many programming languages that can be used for machine learning development such as Python, C++, Java, R, and MATLAB. A lack of experience with some of these languages prevented them from being selected for this thesis, and the project's short timeline would have prevented learning a new programming environment. In addition, C++ and Java don't provide tools for

developing clusters with the metrics we'd like to apply. For example, dynamic time warping and global alignment based K-means are not implemented in any of these languages and would need to be developed from scratch if we chose them, whereas Python offers a variety of libraries that support our data format and the algorithms we're interested in.

The selected Python libraries were the most suitable in terms of support for our datatypes and the development of the algorithm we wanted to move forward with based on testing and experimenting with a variety of libraries.

2.2 Preprocessing

A few preprocessing steps were applied to the data to create a more uniform dataset. Since our goal is to study patients' health trajectories and cluster them according to their trajectory, we decided to delete patients with fewer than three registrations. This decision was based on discussions with supervisors. If there were just one or two records of the patients, meaning the patient had been discharged or deceased, the dataset did not contain enough data to consider their trajectory.

Patients in our dataset were divided into 8 types of wards, including Medicine, Surgical, Neurology, Intensive Care Unit (ICU), Orthopedics, Observation Unit, Psychiatry, Obstetrics, and Gynaecology (Obst/Gyn). Additionally, several patients were not included in any of these categories. There were only 2 to 4 patients identified in certain wards, such as Psychiatry and Obst/Gyn. Due to the small amount of data from those wards, we decided to ignore them during clustering since we couldn't get an effective cluster indicator from them.

We were also provided with another dataset without access to each patient's ward. We merged these two datasets taking into account the missing wards classified under unknown categories.

Dataset	**Holds information about**
Type A	Vital values, WardID, Gender, Level of consciousness
Type B	Vital values, Gender, Level of consciousness

Table 2.1: The difference between the two datasets provided by the Nordlandssykehuset

Moreover, as previously mentioned, patients have different amounts of registration, so we determined to only consider 48 hours of data for each pa-

tient. This decision was inspired by several similar studies conducted in this field (5).

Some patients in the dataset had pauses in their time series, indicating that they had either been moved to another ward or returned to the hospital after being discharged for a time. We decided to only keep the first 48 hours of admission when such a pause was detected. Our dataset contains 5619 patients with a total of 29506 registrations.

Ward	Dedicated ID
Medicine	2
Surgical	3
Neurology	5
ICU	6
Orthopedics	7
Observation Unit	8
Unknown	0

Table 2.2: Type of ward and the ID dedicated to that ward for simplification in the analysis process

	PatientID	Timestamp	WardID	Systolic	Diastolic	O2	Pulse	Temp	Rf	Gender	Conciousness
1	355175	2022-02-23 18:37:39	7	118.0	64.0	96.0	93.0	37.5	16.0	1	Våken
2	355175	2022-02-23 19:31:53	7	125.0	63.0	97.0	87.0	37.6	21.0	1	Våken
3	355175	2022-02-24 05:28:08	7	116.0	59.0	98.0	66.0	36.7	18.0	1	Våken
4	355175	2022-02-24 18:50:30	7	117.0	58.0	96.0	65.0	36.7	17.0	1	Våken
5	355175	2022-02-25 04:48:48	7	98.0	57.0	99.0	60.0	36.1	17.0	1	Våken
...	...	...	...	...	...	...	...	...	...	...	...
9797	1765581	2022-02-27 05:40:00	2	111.0	81.0	98.0	60.0	36.2	17.0	0	Våken
9798	1765581	2022-02-27 20:38:42	2	125.0	80.0	96.0	70.0	37.0	18.0	0	Våken
9799	1765607	2022-02-27 02:09:58	7	129.0	63.0	99.0	97.0	36.1	18.0	0	Våken
9800	1765607	2022-02-27 13:44:39	7	116.0	69.0	97.0	78.0	36.2	12.0	0	Våken
9801	1765607	2022-02-27 21:13:31	7	113.0	70.0	99.0	69.0	36.5	18.0	0	Våken

9750 rows × 11 columns

Figure 2.1: Dataset format with patient vitals, gender, level of consciousness, and the time each record was made

2.3 Non-temporal Analysis

The non-temporal analysis involved calculating the standard deviation, minimum, maximum, and mean of each patient's recorded vitals by writing a Python

script that calculates these data for each vital on a patient's trajectory. These values were used to determine if an outlier case is clearly evident from these analyses or whether we found a clear pattern that could guide us in how to proceed next.

Figure 2.2 shows the calculated metrics for systolic blood pressure for each patient's registries. These metrics have been calculated for all psychological vitals available.

Additionally, we used PCA to analyze these calculated metrics by passing a table with 18 columns for each row representing a single patient. As part of the analysis, we computed the mean, minimum, and maximum values for each of the six vital signs, and scaled all the data so that it fit into one unit before passing it into the PCA model using the StandardScaler from the Scikit-learn library processing module. A total of 18 components were passed to the PCA model, which represented the number of features available in the dataset. In order to avoid any duality in the analysis, we deleted the standard deviation as an input feature, since the stochastic properties may already be represented by PCA.

In the next chapter, we will discuss the analysis gathered from these calculations.

	WardID	Gender	std	min	max	mean
355175	7	1	10.034939	98	125	114.800000
515472	8	0	6.000000	131	143	137.000000
668834	3	0	4.582576	129	138	134.000000
744434	2	0	23.177575	108	166	142.800000
831591	2	0	8.238858	101	126	111.666667
...	...	...	...	...	...	...
1765534	3	0	7.234178	113	126	117.666667
1765553	3	0	16.563011	115	148	130.666667
1765555	3	0	15.526322	83	124	107.333333
1765581	2	0	8.717798	109	125	115.000000
1765607	7	0	8.504901	113	129	119.333333

Figure 2.2: Table of Calculated the standard deviation(std), minimum(min), maximum(max), and mean values of systolic blood pressure for each patient's trajectory

2.4 Clustering

A number of approaches have been incorporated into our early warning system in order to identify outliers or patients with similar trajectory. Taking into ac-

count our unique data format, we decided to compare three different clustering algorithms with different computed formats for our data and different similarity measures. In order to evaluate the quality of each of these approaches and compare them to each other, we used clustering evaluation metrics.

2.4.1 Dataset

In order to compare how different methods perform, we experimented with different formats for our data because we have unequal amounts of recorded registration arrays, and many algorithms do not support this format.

Initially, we used data derived from the non-temporal analysis. Dataset consists of patients as rows, and principal component measures as columns representing the first five that exhibited the most variation. In this dataset, observable features are translated into latent spaces. An abstract, multi-dimensional space encoding a meaningful internal representation of externally observed events is called a latent space. Similar samples in the external world are positioned close to each other in the latent space.

The second approach was to use the raw data as it was and format it to a numpy 2D array, where each patient is an array with arrays of registrations. This type of dataset is not compatible with most clustering algorithms and we use this dataset to compute a similarity matrix and cluster the data using hierarchical clustering and DBSCAN based on the computed matrix.

Our last approach involved interpolating our data so that every patient had the same amount of information. Each patient was extended to 45 entries, and we compared the quality of clusters using interpolation with other approaches.

For easier reference throughout the thesis we refer to these types of datasets respectively as **Type 1**, **Type 2**, and **Type 3**.

2.4.2 K-means

Several libraries have implemented K-means, such as Scikit-learn and TSlearn. TSlearn K-means is specifically designed to cluster time series data, whereas Scikit-learn K-means is a general-purpose clustering algorithm that can handle a wide range of data types (35; 36). K-means has a linear time complexity of O(NTK), where N stands for the total number of data sets, K stands for the total number of partitions, and T stands for the total number of iterations (56). As a result, it handles large datasets efficiently and is relatively faster than other

algorithms. The TSlearn K-means also offers flexibility, since it can be used with different distance metrics, such as Euclidian, DTW, and softDTW (36).

For clustering our data with K-means, we used Type 1 and Type 3 datasets. Type 2 datasets cannot be used with K-means since K-means cannot accept unequal length entries. As Type 1 data is not a time series, we clustered it using Scikit-learn K-means since it is in a non-temporal latent space from the principle component. For Type 3 data, which is a time series, we clustered it using TSlearn TimeSeriesKmeans with Dynamic Time Warping and Soft Dynamic Time Warping distance metrics.

The K-means model was configured to fit our data using a number of parameters and settings. It is important to remember that K-means requires the number of clusters defined before clustering. To determine the optimal number of clusters for each dataset, we used the silhouette score since we did not have enough information about the data itself to determine what number of clusters was appropriate.

Type 1 dataset clustering setting

Scikit-learn's K-means algorithm uses the Euclidean metric for clustering. This is suitable for this specific dataset as it has a uniform format that can be used with this metric. As for other parameters, we can also choose the maximum number of iterations the K-means algorithm runs through before it stops updating with a parameter called max_iter, which was chosen to be 300.

To determine what number of clusters is most appropriate for the data, we also ran this algorithm with a silhouette score. The highest silhouette score was obtained when the algorithm had only two clusters, so we proceeded with two clusters as shown in the figure 2.3.

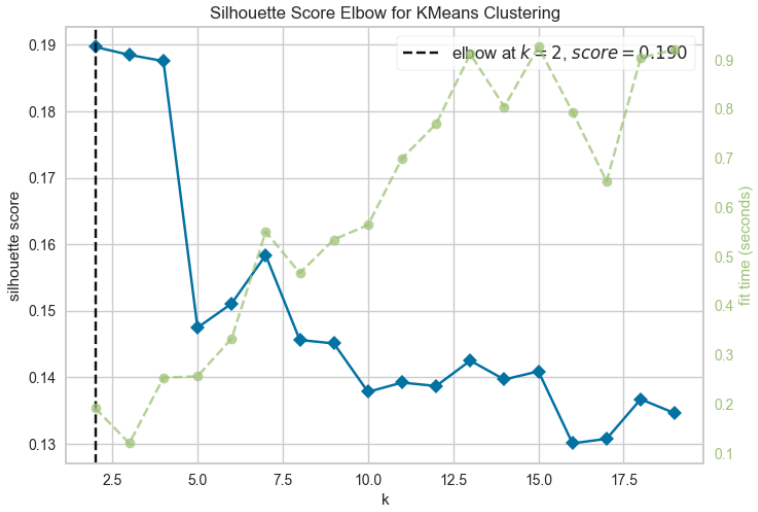


Figure 2.3: Figure of the elbow method using the Silhouette score suggesting the optimal number of clusters for the Type 1 dataset

Type 3 dataset clustering setting

This dataset contains interpolated data that we want to cluster based on their timeseries similarities with DTW and soft-DTW. Based on the Sillohuete score, 2 clusters were selected for this data set. Different metrics values have been tested and the selected ones resulted in higher quality clusters. By computing the Davies-Bouldin score of each cluster made by these various metrics and comparing it with each other, we determined the highest quality cluster by choosing the one resulting in the lowest Davies-Bouldin score.

The chosen metrics for both DTW and soft-DTW were 30 in max_iter and 20 barycenter iterations. Barycenter iterations determine how many iterations are specified for computing cluster barycenters. Barycenter computations update the cluster centroid over time. We can also choose a gamma value for the model if using the soft-DTW metrics which control the softness of the dynamic time warping (DTW) constraint. Higher gamma values result in more rigid alignment, whereas lower values allow for more flexibility in alignment. However, too small values can cause the algorithm to get stuck in local minima (43) the chosen gamma value for this data set was 0.1.

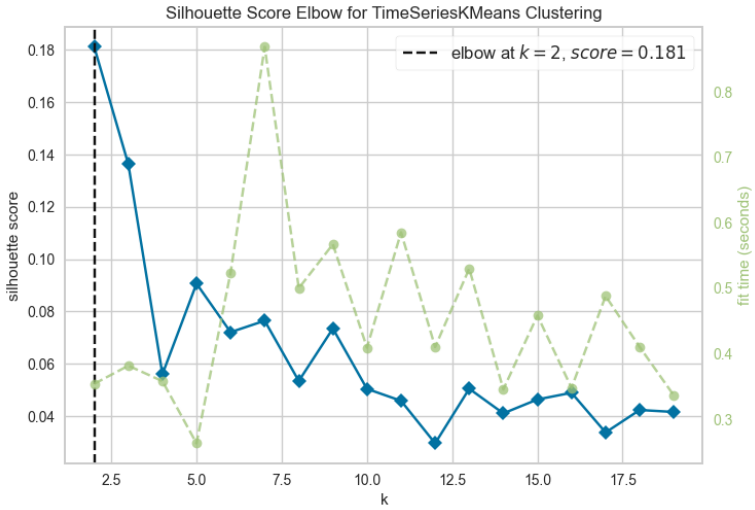


Figure 2.4: Figure of the elbow method using the Silhouette score suggesting the optimal number of clusters for the Type 3 dataset

2.4.3 Hierarchical Clustering

The advantage of hierarchical clustering for us is its ability to accept precomputed similarity matrices. The unique shape of our data made it difficult to find an algorithm that could utilize the data in its original shape. It was important to also find a solution without the use of imputation. This was made possible through hierarchical clustering and the TSlearn library metrics. However, we decided to test both imputed and raw data with this method.

The TSlearn Library does not implement hierarchical clustering but offers support for calculating similarity between time series with various metrics. Some of the libraries that support hierarchical clustering and we have tested for this method are Scikit-learn, SciPy and HDBSCAN.

When using Scikit-learn with a precomputed similarity matrix, there is limited support for the linkage method and only the single or complete linkage method can be used. As we discussed in the Theory section, single and complete linkages are not the most optimal methods when outliers and noise are present.

HDBSCAN (Hierarchical Density Based Spatial Clustering of Applications with Noise) (32) is a hierarchical clustering algorithm extending DBSCAN. While testing HDBScan with different metrics the algorithm only produced one cluster which was not an optimal result to move forward with. We chose to move

forward with our method with the SciPy library due to its flexibility and support for a precomputed similarity matrix.

There are a few metrics and functions that must be tuned when using SciPy clustering, such as preparing the data to be trained on in our case a similarity matrix, choosing a linkage method, and training the model using the fcluster function.

To begin with, we calculated a similarity matrix using TSlearn.metrics. Similarity matrices represent the similarities between pairs of objects or in our case, time series, using metrics like DTW, Soft-DTW, and GAK. The diagonal entries in a similarity matrix are zero, which indicates that each object is identical to itself. Due to the fact that the similarity between objects i and j is the same as that between j and i, the matrix is symmetrical.

We used DTW, Soft-DTW, and GAK because they are suitable for handling multivariate data and data of unequal length (36). In spite of this, we cannot support unequally spaced samples, and the algorithm assumes that all samples are equally spaced. After calculating each of these (dis)similarity metrics using the TSlearn library we passed the output on to hierarchical clustering to determine how many clusters we had. A positive aspect of the SciPy library's hierarchical clustering is that it can decide how many clusters it needs without being informed beforehand. We will compare the performance of these similarity metrics among the type 2 and type 3 datasets.

Despite trying all the linkage methods mentioned in Chapter 1, only the Ward method produced distinctive clusters compared to the threshold. While using other methods, a small change like 0.05 in the threshold changed the number of clusters from 2000 to 1. However, the Ward method was relatively smooth as I adjusted the threshold.

By using the fcluster function, it is possible to obtain the number and labeling of clusters based on the linkage method and some parameters need to be defined. Fcluster takes two main arguments: a linkage matrix and a threshold value of t. This threshold depends on the criteria chosen to extract the clusters, specified by the criterion argument. There are five different criteria available with the Fcluster function, including inconsistent, distance, maxclust, monocrit, and maxclust_monocrit (30). The inconsistent method uses the inconsistency coefficient, which measures the difference between the height of a link in a dendrogram and the average height of the links at the same level. Our experiments showed that this method did not work well on our data and was too sensitive to thresholds, and the number of clusters varied wildly between 1 and 3000, so we chose not to use it. The threshold at which a cluster will form in monocrit is determined by a user-defined monotonic function of the cluster statistics, but

defining such a monotonic function requires more domain expertise, and the clusters may not be easily interpreted. The maxclust_monocrit method combines the last two methods, so it is also incompatible with our needs. Maxclust tries to find clusters smaller than t, which is not suitable for our case since we do not want to set a default number of clusters for our algorithm. The fcluster is formed using the distance method, which groups all pairs of points whose distance is less than or equal to t into the same cluster. The value of t was chosen based on the dendrogram plot, which we interpreted.

2.4.4 DBSCAN

The DBSCAN model is implemented using Scikit-learn, and there are a few parameters that can be set for training, including min_sample, epsilon, and metric. In our case, we chose a precomputed metric representing our similarity matrix with a min_sample parameter of 100 and an Epsilon of 0.5. This means that clusters will only form if there are at least 100 samples within 0.5 distance of each other. In a similar manner to hierarchical clustering, we need not specify the number of clusters, which is a major advantage for us. Based on types 2 and 3 datasets, we pass similarity matrices built with GAK, DTW, and Soft-DTW metrics using TSlearn library to the DBSCAN model.

2.5 Methods pipeline

Figure 2.5 shows a summary of how the methods and datatypes interact.

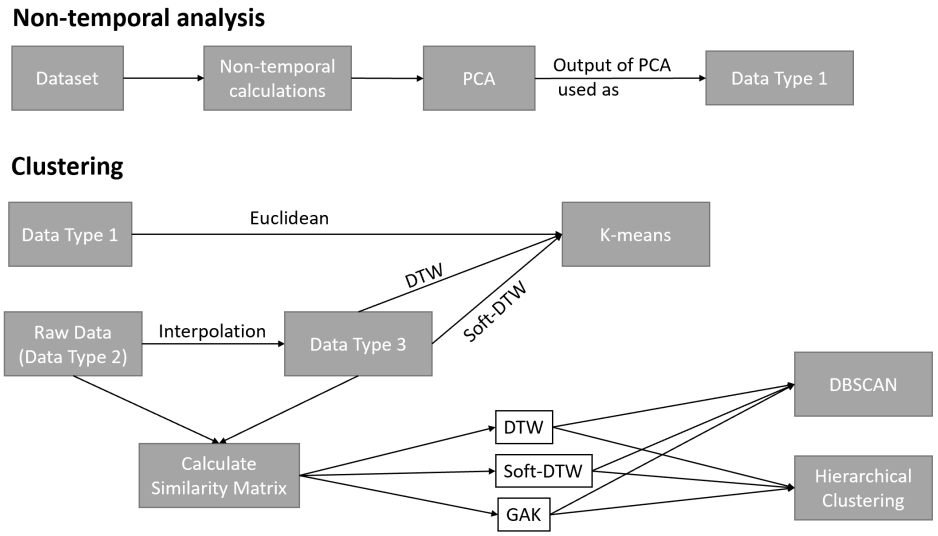


Figure 2.5: Pipeline summary of the methods

3

Results and Discussion

In this chapter, we will examine the results of the experiments described in chapter 2. The first section will present the findings, and the second section will discuss what these findings could mean for the project as a whole.

3.1 Results

The results of the methodology described in chapter 2 are presented in this section. In the first section, we present the results of the non-temporal analysis, while in the second section, we examine the quality of the algorithms used for clustering.

3.1.1 Non-temporal Analysis

As part of a non-temporal analysis, we examined the outlier patients in each vitalia within each ward. There were four categories of data analyzed: the mean, the minimum, the maximum, and the standard deviation. Outlier patients are those with significant differences in each of the statistical categories for vitals compared to the general population.

As mentioned in the previous chapter and in table 2.1, we had access to two different datasets. Type A contained information about the wards in which

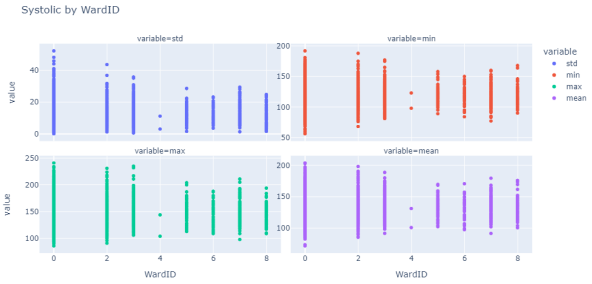
patients were hospitalized and Type B without any information regarding the wards. To identify any connections or outliers within these known wards, we decided to initially analyse the dataset with ward information.

Our first step was to evaluate whether outlier patients had any specific differences or patterns based on their consciousness level. In light of the fact that consciousness level was the only modifying label, we wanted to figure out if we could determine the existing pattern before showing the outliers to the medical expert from a label point of view. The second step was to show Dr. Nymo, Our superviser in the Nordlandssykehuset, some of these outliers and ask him if they indicated a visible pattern that can be used to guide our next steps.

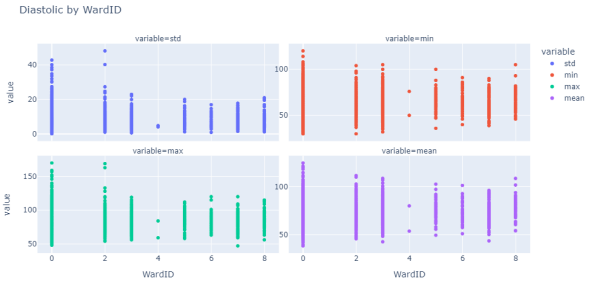
Each of the figures in Figures 3.1 corresponds to a specific vital, and the vertical lines correspond to wardIDs from 0 to 8 as shown in the table 2.2.

Based on these plots 3.1, we extracted outliers from known wards and compared their consciousness levels. However, we found no correlation between their level of consciousness and outliers due to the fact that they mostly shared the same level of consciousness "awake", so we could not see any identifying patterns that could improve our analysis.

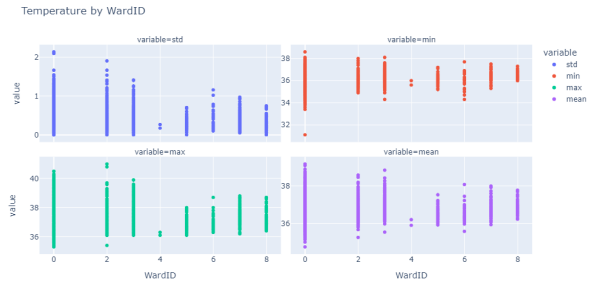
The next step was to analyze features with principal component analysis. Figure 3.2 shows the percentage of variance explained by each principal component in a Principal Component Analysis (PCA). From the scree plot, it can be seen that the first five components cover the 80% of variance, while from the sixth to the eighteenth components, the variance of each component is 5% and below.



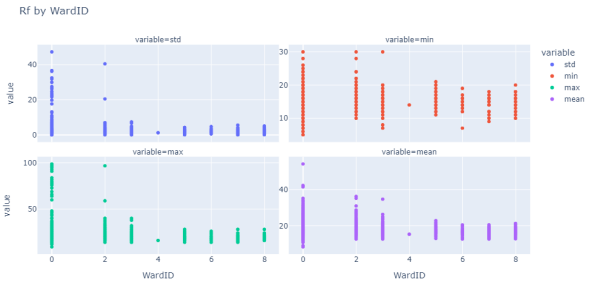
(a) Systolic blood pressure



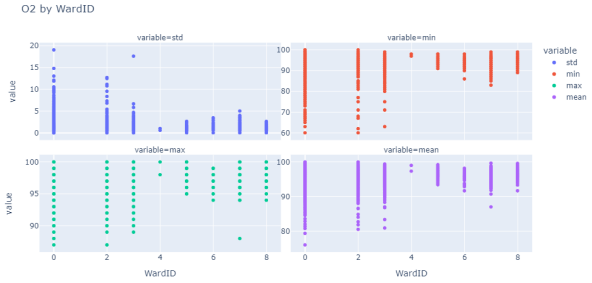
(b) Diastolic blood pressure



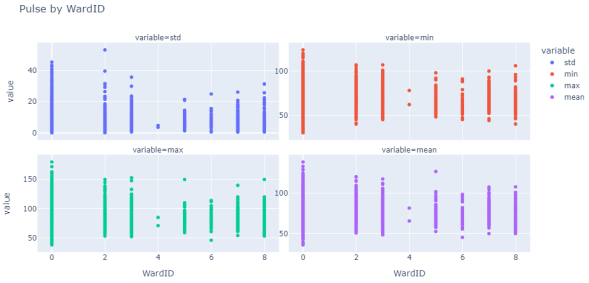
(c) Temperature level



(d) Respiratory rate



(e) Oxygen saturation level



(f) Pulse rate

Figure 3.1: Figures of each vital level measured by std, min, max, and mean value for each patient divided by ward ID

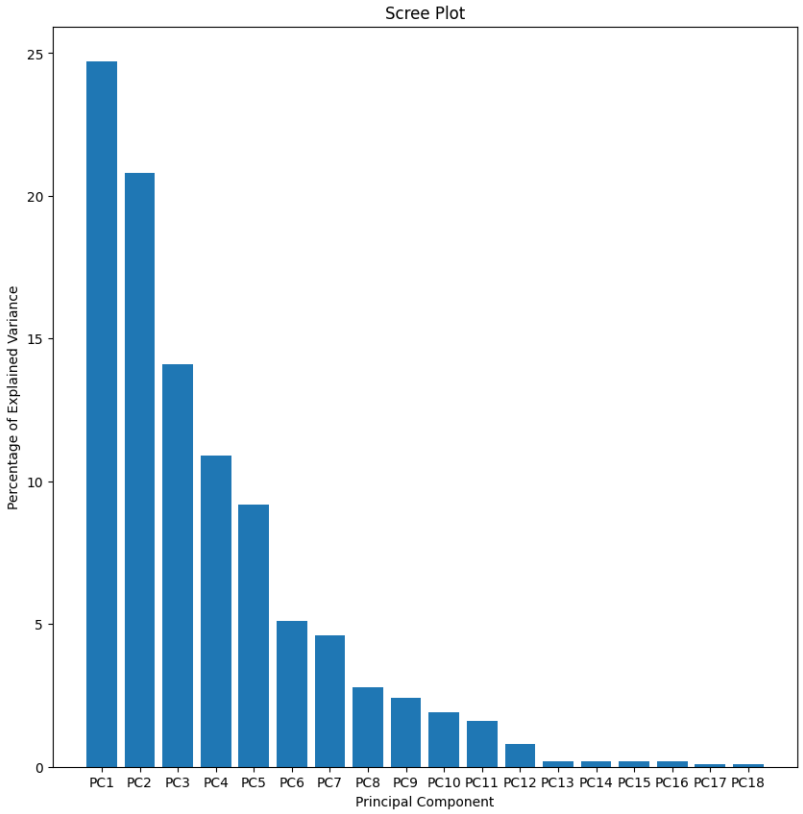
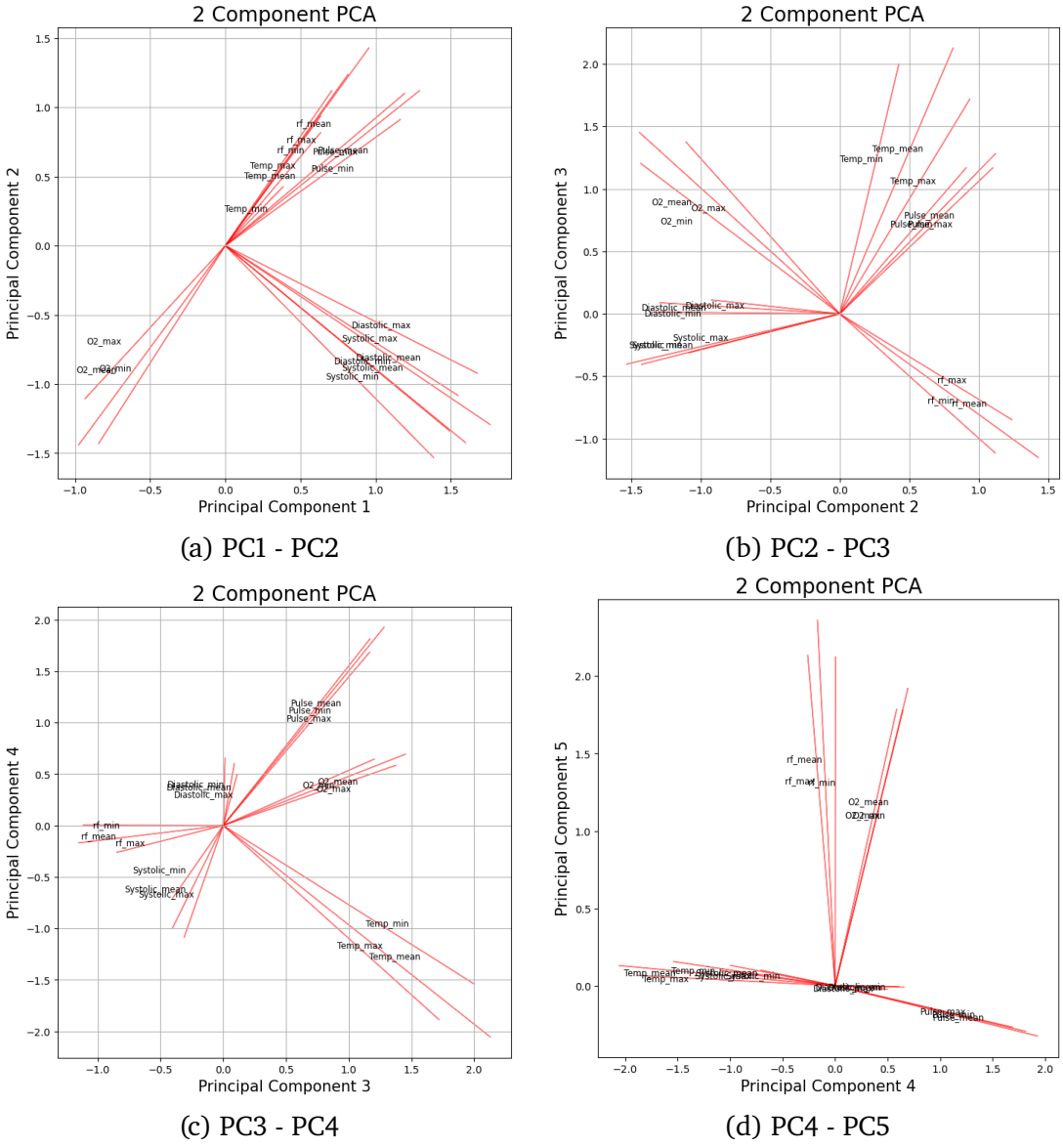


Figure 3.2: PCA analysis - 18 component PCA

Figure 3.3(a) illustrates the variances of each feature on PC1 and PC2. As can be seen from this plot, systolic and diastolic values tend to be more closely correlated with one another, and temperature and RF values are also closely correlated with one another where the temperature has lower variance than RF, and both temperature and RF are correlated with pulse values and all tend to be negatively correlated with oxygen levels.

On the other hand, the variance of each feature on PC2 and PC3 3.3(b) shows temperatures and RF are uncorrelated. Rf is negatively correlated with O2, which is also present in figure 3.3(a). Systolic and diastolic pressures are also correlated.



(a) PC1 - PC2

(b) PC2 - PC3

(c) PC3 - PC4

(d) PC4 - PC5

Figure 3.3: PCA analysis - Biplot variance of each feature

We can see in figure 3.3(c) that diastolic and systolic have less variance and are no longer correlated as in the first two plots. Neither temperature nor pulse is correlated. We can see from the angle of all the features in this plot that no two are closely related, though systolic and RF seem to be related in a relatively weak way.

We can observe in Figure 3.3(d) that temperature and systolic seem to have

a correlation, whereas systolic and pulse seem to be negatively correlated, o2 and RF seem to be correlated, and diastolic is correlated with the pulse, with diastolic having a lower variance.

In figure 3.4, you can see where patients stand on PC1 versus PC2, showing that all patients fall into the same blob without distinguishable clusters. We studied and compared the middle points on the PCA plot and some of the outliers in the hope of them providing information about time series characteristics and observing if any apparent patterns emerged.

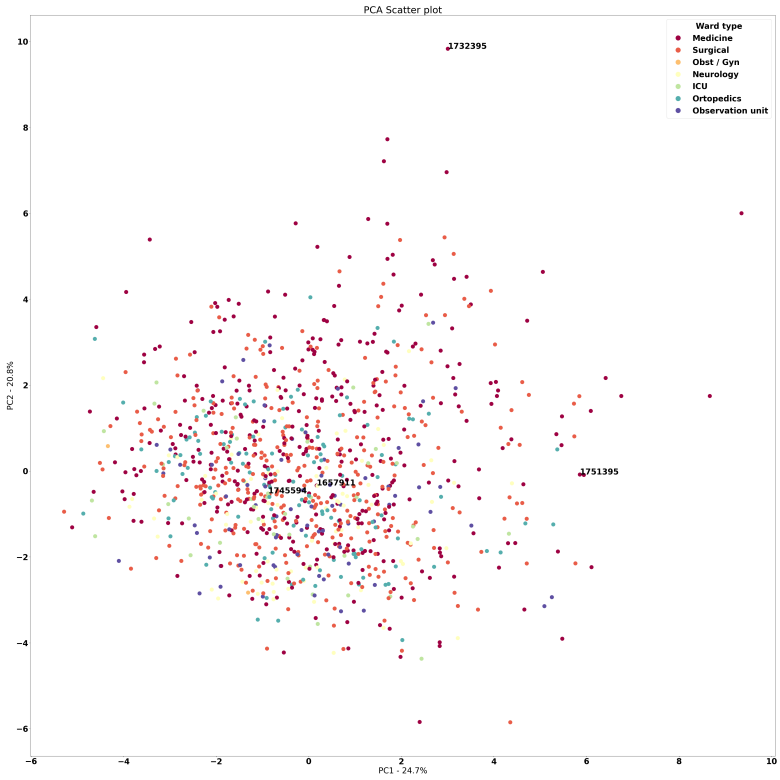


Figure 3.4: PCA analysis - PC1 vs PC2 - The patients highlighted in the figure are those who are referred to and plotted in figures 3.5 to 3.8

The common characteristic of outliers was a high amount of RF, but all of these outliers also had the same level of consciousness as "awake". Some of these outliers were plotted out and sent to Dr. Nymo for feedback. He mentioned that most outliers are caused by errors in registering one of the values, as for example the RF value in figure 3.6. Another outlier was noticeable because of their low oxygen level, figure 3.5, and Dr. Nymo mentioned that this particular individual might be ill. Also patients in the ICU, figure 3.7, are supposed to be seriously ill, but their registration does not stand out in the PCA analysis

and it is similar to a stable patient in the middle of the plot 3.8. Regarding ICU patients, Dr. Nymo mentioned that their vitals might be artificially maintained based on the medications they're on.

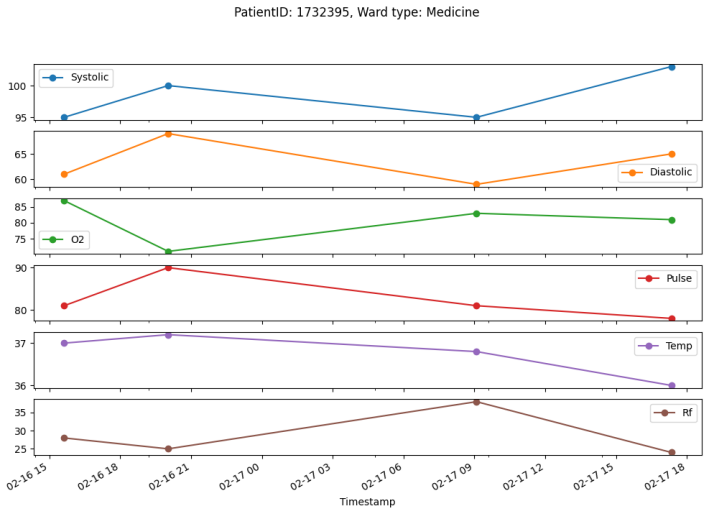


Figure 3.5: Patient 1732395 - Outlier and might be sick based on the low value of O2

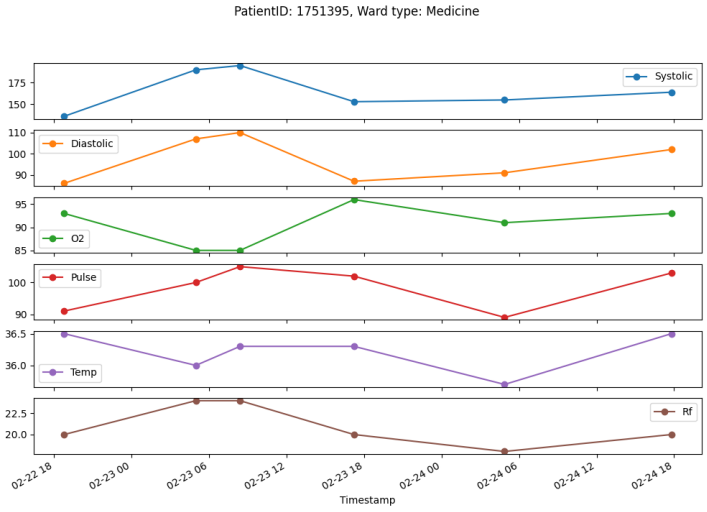


Figure 3.6: Patient 1751395 - Outlier because of invalid and high values of RF due to error in registeration

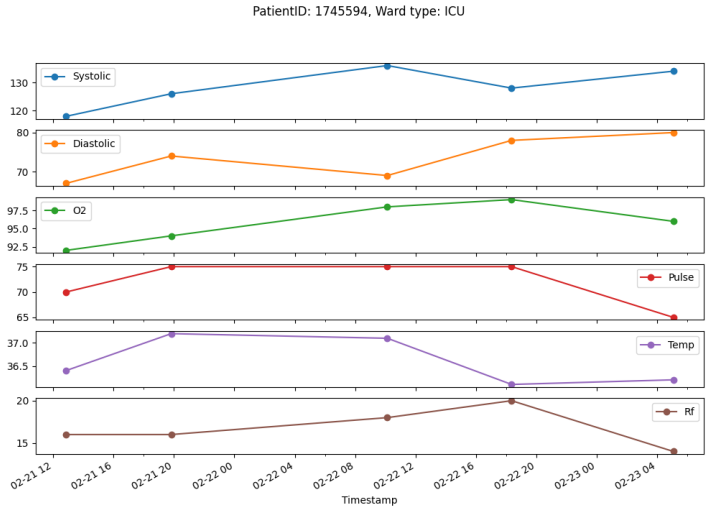


Figure 3.7: Patient 1745594 - ICU - Vital range seemed normal and the patient was close to the center of the PCA cluster

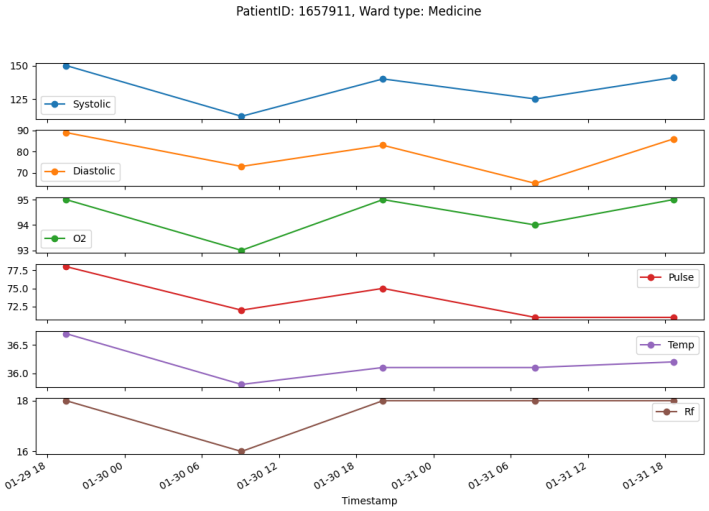


Figure 3.8: Patient 1657911 - A normal patient close to the center of cluster

3.1.2 Clustering

The dataset used in the clustering section is a merged dataset from Type A and Type B datasets, as described in 2.1.

Kmeans

In section 2.4.2, we describe how we trained our K-means model for two different datatypes, Type 1 which consists of principal component analysis from the non-temporal analysis, and Type 3 which contains interpolated data with 45 registrations per patient.

In table 3.1, the results of these datasets are compared with the referred metrics. As shown in the table, when using the K-means algorithm with Euclidean distance as a metric for type 1 dataset we had the highest clustering quality.

DataType	Metric	DBI Score
Type 1	Euclidean	1.9134
Type 3	DTW	2.8181
Type 3	Soft-DTW	2.9065

Table 3.1: Davies-Bouldin Index score trained with each similarity metric using the K-means algorithm

Type 1 dataset with Euclidean K-means		
Number of Patients	**Cluster 1**	**Cluster 2**
Total	2586	3033
Male	1542	1818
Female	1573	1769
Completely awake	2435	2948
Consciousness level other than awake	151	85
Reagerar på tiltale	116	66
Reagerar ved smertestimulering	30	11
Nyoppstått forvirring	35	11
Reagerer ikke på tale eller smertestimulering	9	4
Våken og orientert	0	2

Table 3.2: Distribution of patients among two clusters in the chosen K-means algorithm

Apart from the vital level, we also had access to each patient's consciousness level and gender in the dataset. In table 3.2, we extracted patient details on their given cluster and the characteristics we had access to when clustered by K-means and Euclidean distance using Type 1 dataset.

From table 3.2, we can observe that the two clusters show an even distribution of females and males in each group. The number of patients who were completely awake throughout their stay in Cluster 2 was 2948. The number of patients in a similar situation in Cluster 1 was 2435. The category completely awake means that these patients had their consciousness level registered as "Våken" in every one of their submitted registrations.

We also counted the number of times each patient had a different state of consciousness than awake and listed them. The consciousness level in these patients might vary between all the categories, and they are not necessarily experiencing the same level of consciousness throughout all their registrations. They could be awake in one registeration and "reagerar på tiltale" in another. There were almost twice as many patients in cluster 1 with a consciousness level other than awake. More than 1.75 times as many patients were in the "reagerar på tiltale" category, which means the patients are partially unconscious, but respond when spoken to, and fall back and forth between conscious and unconscious. In Cluster 1, there were also more than twice as many patients in "Reagerar ved smertestimulering". In this category, the person is unconscious but reacts to physical stimuli. The same goes for the "Nyoppstått forvirring" category, which means that the patient is awake, but unclear and confused when questioned. Additionally, two times as many patients reported "Reagerer ikke på tale eller smertestimulering" in cluster 1 indicating a profoundly unconscious state.

Figure 3.9 plots the mean and standard deviation of each vital among all patients in each cluster in their first 24 hours of hospitalization. In the actual data, the patients did not have the same amount of registration, so this plot is based on interpolated data since the scales could not be compared. We can observe that in all vitals except oxygen level and temperature, patients in cluster 1 displayed a higher value compared to patients in cluster 2. Nevertheless, patients in cluster 1 had a lower level of blood oxygen saturation, which from our early discussions with medical experts indicates a higher risk of illness. Moreover, both clusters experienced the same mean temperature values.

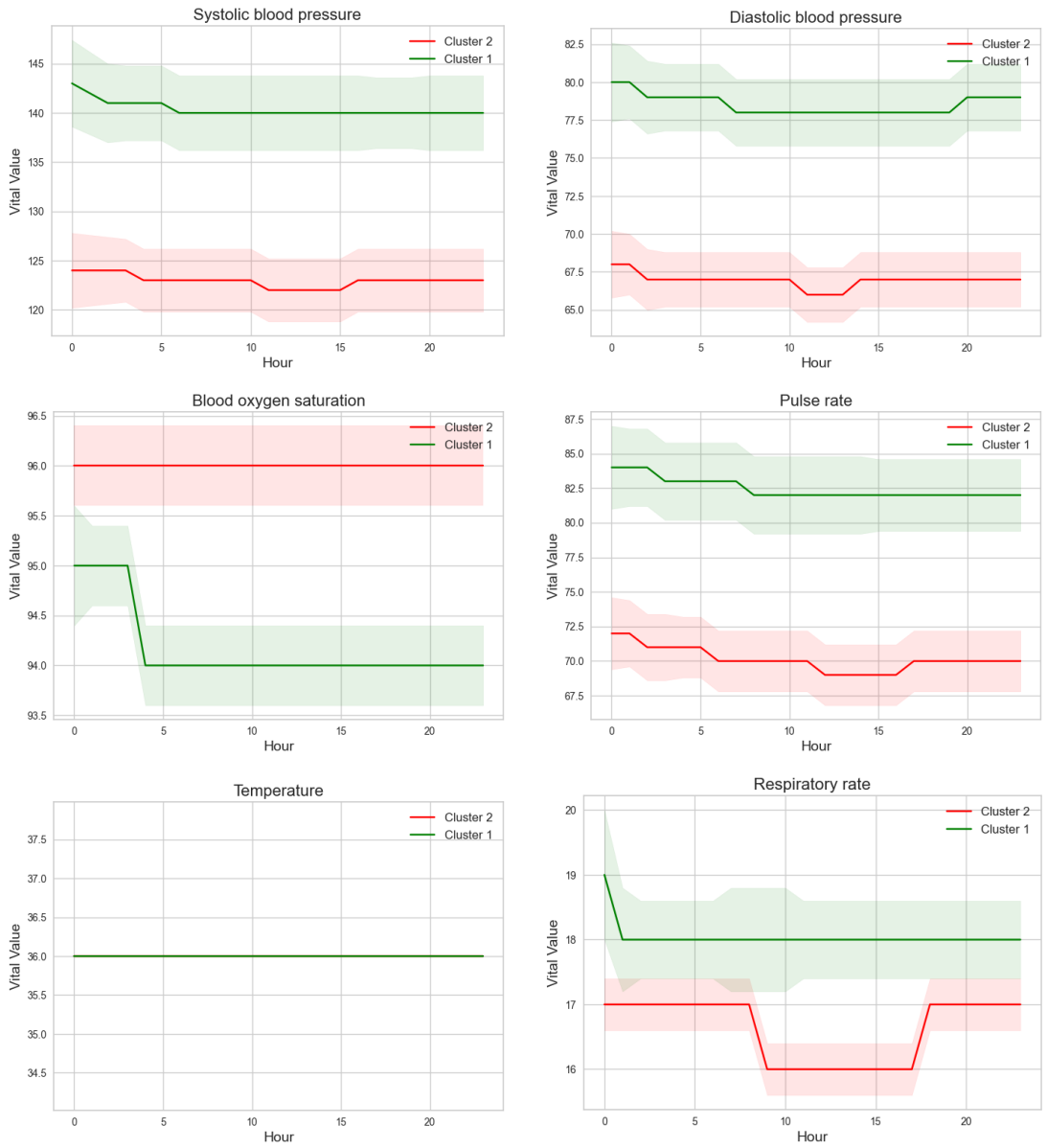


Figure 3.9: Mean and standard deviation distribution of each vital sign among clusters during the first 24 hours of admission

Hierarchical clustering

Section 2.4.2 describes how we trained our hierarchical model on two different datasets. These datasets are Type 2 which consists of our raw data, and Type 3 which contains interpolated data with 45 registrations per patient. In the table

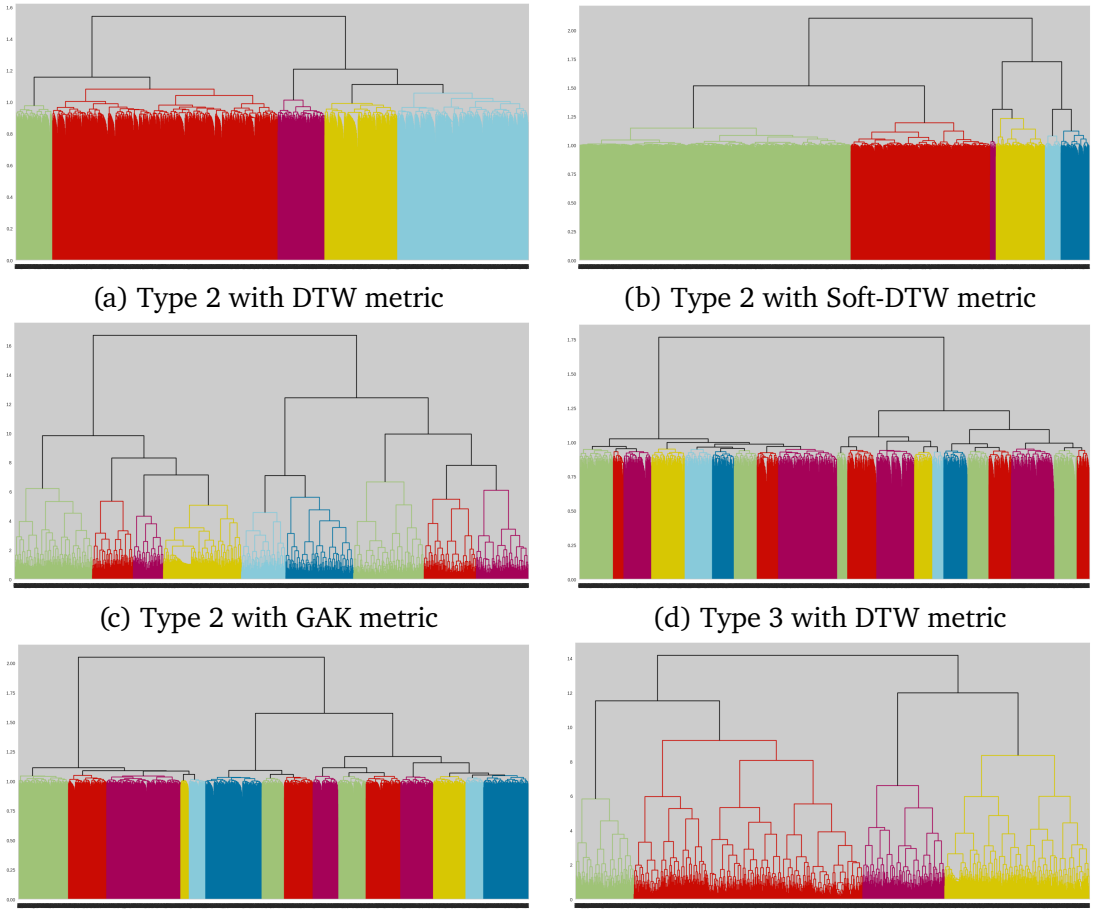
3.3, the results of these datasets are compared with the referred metrics. Table 3.3 shows that different data sets reached different Davies-Bouldin scores and suggested different cluster numbers, based on the metric. Based on the results listed in this table, we were able to achieve the best clustering quality using a hierarchical algorithm with global alignment kernel as a distance metric for type 3 datasets.

In figure 3.10, we see the dendrogram produced for each dataset based on the given metric, which indicates how patients are clustered and how many clusters each method generated. Hierarchical clustering can create more or fewer clusters depending on how the thresholds are set. According to 3.10(a), five clusters were found for this dataset when the threshold was set at 1.1. However, we tested the dataset with a range of threshold amounts. For example, we got three clusters at a threshold of 1.15, while 12 clusters were found at a threshold of 1. Choosing the current threshold and number of clusters was based on the fact that we got a lower Davies-Bouldin score for this value, which indicates a higher level of cluster quality, and this decision was extended to all other datasets and metrics.

According to Table 3.3, when using Soft-DTW and DTW, clusters formed with dataset type 2 is generally of higher quality than clusters formed with dataset type 3. Furthermore, Soft-DTW had better clustering quality than DTW for type 2 datasets, and global alignment had better clustering quality than the two other metrics. In general, the global alignment kernel metric yielded better results in both datasets, with the type 3 dataset producing the best results.

DataType	**Metric**	**Clusters**	**DBI Score**
Type 2	DTW	5	28.3733
Type 2	Soft-DTW	6	14.4297
Type 2	GAK	9	5.6746
Type 3	DTW	20	58.5925
Type 3	Soft-DTW	15	62.2391
Type 3	GAK	4	2.8295

Table 3.3: Davies-Bouldin Index score trained with each similarity metrics using hierarchical clustering



(a) Type 2 with DTW metric

(b) Type 2 with Soft-DTW metric

(c) Type 2 with GAK metric

(d) Type 3 with DTW metric

(c) Type 3 with Soft-DTW metric

(d) Type 3 with GAK metric

Figure 3.10: Dendrograms of clustering each type of dataset with DTW, Soft-DTW, and GAK as similarity metrics

The distribution of patients clustered by hierarchical clustering and GAK metric is shown in Table 3.4. In table 3.4, we observe an approximately even distribution of male and female patients in all four clusters. Additionally, cluster 4 holds most of the patients and 97% of patients in this cluster were completely awake during their hospitalization. There were 2% percent of patients in cluster 1 who appeared to be having a different level of consciousness from awake, 7% percent in cluster 2, 3% percent in cluster 3, and 3% percent in cluster 4. Overall, Cluster 2 had the most patients appearing to have different consciousness states in all categories. Cluster 4 had the second highest amount of patients experiencing a varied level of consciousness other than awake. There were no patients in cluster 3 in the “Reagerer ikke på tale eller smertestimulering” category meaning that there were no patients that were deeply unconscious in

this category.

Type 3 dataset with GAK Hierarchical clustering				
Number of Patients	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**
Total	681	1616	635	2687
Male	393	995	416	1556
Female	432	936	347	1627
Completely awake	662	1493	614	2614
Conciousness level other than awake	19	123	21	73
Reagerar på tiltale	16	96	14	56
Reagerar ved smertestimulering	6	20	2	13
Nyoppstått forvirring	3	22	4	17
Reagerer ikke på tale eller smertestimulering	2	8	0	3
Våken og orientert	0	0	1	1

Table 3.4: Distribution of patients among four clusters in the chosen hierarchical algorithm

Figure 3.11 plots the mean and standard deviation of each vital among all patients in each cluster in their first 24 hours of hospitalization. The systolic and diastolic blood pressures of patients in cluster 1 were the highest. There is a lower level of oxygen saturation among patients in clusters 2 and 4. Patients in cluster 2 had a higher pulse rate, while clusters 1 and 3 had similar amounts, and cluster 4 had the smallest. In addition cluster 2 had the second most highest systolic and diastolic blood pressure and the highest pulse rate and respiratory rate. The temperature of patients in cluster 3 seemed stable throughout their stay, whereas cluster 4 was the lowest. Clusters 1 and 3 were fairly similar and cluster 4 had the lowest respiratory rate.

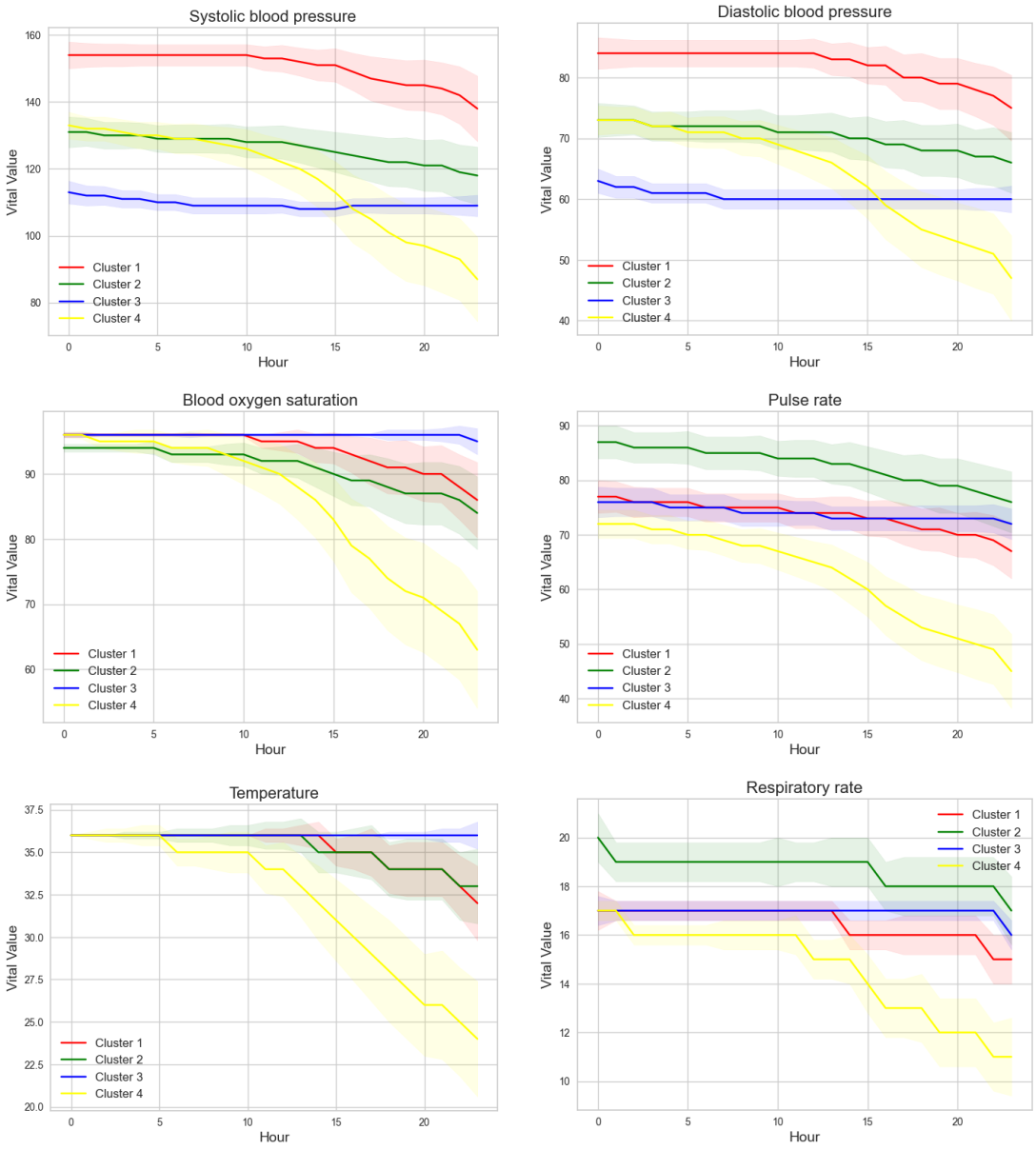


Figure 3.11: Mean and standard deviation distribution of each vital sign among clusters during the first 24 hours of admission

DBSCAN

Using three metric distances and two datatypes for all cases, DBSCAN would place all patients in one cluster, so we could not evaluate the clustering quality.

3.2 Discussion

In this section, we analyze and discuss the results shown in section 3.1.

3.2.1 Non-temporal analysis

According to the task description found in appendix B, the purpose of the non-temporal analysis was to answer the following research question:

RQ0: Based on the extraction of maximum, minimum, mean, and standard deviation and Principal Component Analysis for each feature, can there be found distinguishable clusters? If so, what conclusions could be drawn from this?

In response to the research question, we examined the non-temporal dimensions of the data by analyzing the minimum, maximum, mean, and standard deviation of each patient's vitals and studying their outliers according to the information provided about their consciousness. Moreover, we analyzed the trends and possible clusters using principle component analysis and compared patients with one another. We also contacted medical experts for supervision on these outliers and average patients.

The results of the last section indicate, however, that we could not find any pattern among the outliers and that almost all of the cases had the same level of consciousness. When conducting principle component analysis there were no distinguishable clusters among patients. In our presentation to the medical expert, we determined that most outlier cases result from registration errors.

As a result of the comments of the medical experts and our analysis, we have concluded that without direct supervision from the hospital and an informative label, it is not possible to determine whether a person is ill or in good health, or whether an outlier is a patient who is actually ill or is just an error. Furthermore, neither the comments we received nor our studies suggested a pattern that indicates these outliers are connected to patient characteristics or ward IDs. Only 39 out of 5501 registrations with known wards had a different level of consciousness than awake, which indicates the data is not sufficiently varied. The lack of a correlation between vitals and wards and the lack of variety in the dataset led us to discard the ward ID information for the remainder of this study.

3.2.2 Clustering

According to the task description found in appendix B, the purpose of this section was to answer the following research questions:

RQ1: On the given time-series dataset on vitalia from NLSH, is it possible to identify a clustering algorithm able to identify certain clearly distinguishable criteria?

RQ2: Given that RQ1 is fulfilled by utilizing one or several algorithms, is it possible to connect these clusters to certain types of patient characteristics?

Besides the algorithms and libraries mentioned in methods and results, we examined a variety of clustering algorithms and libraries, including K-shape, kernel K-means and SKtime library. In light of the advantages outlined in the method section, we chose to use K-means from the Scikit-learn and TSlearn libraries and hierarchical clustering from the SciPy library to continue our research. The fact that some of the algorithms tested were either not compatible with our data type or did not produce results that could be discussed further was another reason why we did not move forward with these methods.

Based on our comparison of results, we found that training the K-means model on data type 1, principle components, gave the best results, according to the lower values of the measured Davies-Bouldin index in tables 3.1 and 3.3, when considering all distance metrics and data types applied to this section. Based on the clusters generated by this method in table 3.2 and figure 3.9, we found that both clusters are indeed distinct in the sense that patients in cluster 1 appear to experience higher vital values in all categories as well as lower oxygen saturation, indicative of being considered at higher risk compared to patients in cluster 2. According to the table 3.2, twice as many patients were experiencing difficulties in their consciousness level in cluster 1 as there were in cluster 2, which is in line with our findings based on figure 3.9.

Based on our comparison of hierarchical clustering results, the model trained on interpolated data with the global alignment kernel produced the best results. After further analysis of how clusters were formed, it was found that cluster 4 patients are the ones with the fewest registrations and the most interpolation. Our interpolation method decays to zero over time and by the amount of interpolation, and the fact that cluster four decays to zero on every vital faster than all the other clusters and the distribution of patients confirms that the number of registrations each patient has plays a bigger role here. We do not recommend this method, since the clustering algorithm is dependent on the number of times each patient has been seen, not the level of their illness. Thus, while the clusters in this method appear distinct from the figure 3.11, being

dependent on such details is not an optimal result.

According to the discussion and the scores in tables 3.1 and 3.3, overall K-means clustering in connection with utilizing the Type 1 dataset is the most effective and identifies clearly distinct clusters. Studying the available characteristics of table 3.2, including gender and consciousness level, we can see that the clusters do not relate to the gender of the patients, but rather to their consciousness level, which is itself dependent on vital values. Unfortunately, we had no other characteristics to examine further.

The description also listed two other research questions:

RQ3: Given that a clinician could assist in labeling a set of characteristic time-series with diagnosis, will it be possible to perform semi-supervised learning based on this?

RQ3.1: could a semi-supervised learning could be aided by employing GAN(Generative Adversarial Network) algorithms to generate a larger set of labeled time-series?

Unfortunatly we were unable to answer and address these two questions due to the lack of further assistance that was needed from a clinician.

4

Conclusion and Future Work

Our thesis began with an analysis of the National Early Warning Score and identified the need for an updated system. Nordlandssykehuset presented us with two datasets to explore the possibility of creating an anomaly detection system using unsupervised learning on patient physiological registrations. A non-temporal dimension and the viability of training clustering models on physiological data are examined throughout the thesis. With 29000 records trained using various algorithms and four distance metrics, we evaluated each model's viability and practicality, then selected the most effective one and discussed its results. The models were also tested using three data types, including raw data, interpolated data, and a latent space generated from non-temporal dimension analysis. As per the task description, we have successfully addressed the first three research questions of the highest priority.

In conclusion, we would like to suggest a few recommendations and ideas for moving forward with this project.

In order to further advance this research, we need access to more data, comprehensive labels, and patient characteristics, along with more clinical assistance at every step. We would find it more helpful if we had access to the ward ID of all patients, not just a few. Nonetheless, a data-sharing concept for different hospitals might also be a good idea in order to expand data resources.

As stated in the introduction, the current issue of the NEWS is that it considers the same range of normals for all patients regardless of their circumstances. As for future versions of this system, I suggest further evaluating the normal range based on each ward and providing patients with more specifically tailored notifications.

The imputation method used in this thesis was developed to fill in the missing hours between vitals record registrations, but it did not predict the future, so it had zeros after the registration period ended. The supervisor Helge Fredriksen has developed a newer imputation method which fills in the remaining hours of patients' missing hours with a mean of the vitals from the recorded registrations. This method could be tested further, but another suggestion would be to only consider patients with at least 24 hours' worth of registration, so each can be imputed equally.

There are other clustering algorithms that could be explored, such as spectral clustering, GRIDSCAN, affinity propagation, and OPTICS, which we unfortunately did not have the time to consider during our work with the thesis. In addition, as suggested in research question 3 in case of having access to diagnosis we can explore if it's possible to perform semi-supervised learning and additionally generate a larger labeled dataset using a Generative Adversarial Network.

Bibliography

[1] B. Davazdahemami, H. M. Zolbanin, and D. Delen, "An explanatory analytics framework for early detection of chronic risk factors in pandemics," *Healthcare Analytics*, vol. 2, p. 100020, 2022.

[2] RCP, "Royal college of physicians: National early warning score (news): standardising the assessment of acute-illness severity in the nhs," *RCP (London, England)*, 2012.

[3] RCP, "Royal college of physicians: National early warning score (news) 2: Standardising the assessment of acute-illness severity in the nhs," *RCP (London, England)*, 2017.

[4] F. Vella, "Estimating models with sample selection bias: A survey," *The Journal of human resources*, vol. 33, pp. 127–169, 1998.

[5] B. M. Marlin, D. C. Kale, R. G. Khemani, and R. C. Wetzel, "Unsupervised pattern discovery in electronic health care data using probabilistic clustering models," p. 389–398, 2012.

[6] S. Seto, W. Zhang, and Y. Zhou, "Multivariate time series classification using dynamic time warping template selection for human activity recognition," *CoRR*, vol. abs/1512.06747, 2015.

[7] R. Donders, G. van der Heijden, T. Stijnen, and K. Moons, "Review: A gentle introduction to imputation of missing values," *Journal of clinical epidemiology*, vol. 59, pp. 1087–91, 11 2006.

[8] Y. Li, Y. Ren, T. J. Loftus, S. Datta, M. Ruppert, Z. Guan, D. Wu, P. Rashidi, T. Ozrazgat-Baslanti, and A. Bihorac, "Application of deep interpolation network for clustering of physiologic time series," 2020.

[9] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, "Indexing multi-dimensional time-series with support for multiple distance measures," p. 216–225, 2003.

[10] F. M. Bianchi, S. Scardapane, S. Løkse, and R. Jenssen, "Reservoir computing approaches for representation and classification of multivariate time series," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, 03 2018.

[11] K. Øyvind Mikalsen, F. M. Bianchi, C. Soguero-Ruiz, and R. Jenssen, "Time series cluster kernel for learning similarities between multivariate time series with missing data," *Pattern Recognition*, vol. 76, pp. 569–581, 2018.

[12] A. M. Turing, "Computing machinery and intelligence mind," vol. 49, pp. 433–460, 1950.

[13] J. Bell, *What Is Machine Learning?* John Wiley and Sons, Ltd, 2014.

[14] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[15] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

[16] T. J. Cleophas, A. H. Zwinderman, and H. I. Cleophas-Allers, *Machine learning in medicine*, vol. 9. Springer, 2013.

[17] I. El Naqa and M. J. Murphy, *What Is Machine Learning?* Cham: Springer International Publishing, 2015.

[18] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, 2015.

[19] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, second ed., 2018.

[20] P. Rai and S. Singh, "A survey of clustering techniques," *International Journal of Computer Applications*, vol. 7, no. 12, pp. 1–5, 2010.

[21] A. Oliveira and C. Antunes, *Temporal data mining: An overview*. 2001.

[22] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, "Time-series clustering – a decade review," *Information Systems*, vol. 53, pp. 16–38, 2015.

[23] T.-c. Fu, F.-l. Chung, V. Ng, and R. Luk, *Pattern discovery from stock time series using self-organizing maps*, vol. 1. 2001.

[24] E. Keogh, S. Lonardi, and B.-c. Chiu, *Finding surprising patterns in a time series database in linear time and space*. 2002.

[25] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," vol. 1, pp. 281–297, 1967.

[26] A. Geron, *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O'Reilly Media, 2017.

[27] V. Cohen-addad, V. Kanade, F. Mallmann-trenn, and C. Mathieu, "Hierarchical clustering: Objective functions and algorithms," *J. ACM*, vol. 66, no. 4, 2019.

[28] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Addison Wesley, us ed ed., 2005.

[29] G. W. Milligan, "An examination of the effect of six types of error perturbation on fifteen clustering algorithms," *Psychometrika*, vol. 45, pp. 325–342, 1980.

[30] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[31] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," 1996.

[32] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," 2013.

[33] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *The Journal of Open Source Software*, vol. 2, 03 2017.

[34] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel,

M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[36] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, and E. Woods, "Tslearn, a machine learning toolkit for time series data," *Journal of Machine Learning Research*, vol. 21, no. 118, 2020.

[37] Z. Khan, A. Anjum, K. Soomro, and M. A. Tahir, "Towards cloud based big data analytics for smart future cities," *Journal of Cloud Computing*, vol. 4, 2015.

[38] T.-c. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, pp. 164–181, 2011.

[39] H. Sakoe and S. Chiba, *A Dynamic Programming Approach to Continuous Speech Recognition*, vol. 3. Budapest: Akadémiai Kiadó, 1971.

[40] E. Keogh, S. Lonardi, and B.-c. Chiu, *Dynamic Time Warping*. Springer Berlin Heidelberg, 2007.

[41] D. J. Berndt and J. Clifford, *Using Dynamic Time Warping to Find Patterns in Time Series*. 1994.

[42] M. Uwe and D. Schramm, "Multivariate dynamic time warping in automotive applications: A review," *Intelligent Data Analysis*, vol. 23, p. 535–553, 2019.

[43] M. Cuturi and M. Blondel, "Soft-dtw: a differentiable loss function for time-series," *International Conference on Machine Learning*, 2017.

[44] J. Tabak, *Geometry: The Language of Space and Form*. 2004.

[45] B. Ghojogh, M. Crowley, F. Karray, and A. Ghodsi, *Elements of Dimensionality Reduction and Manifold Learning*. 2023.

[46] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 11, pp. 559–572, 1901.

[47] I. Jolliffe, *Principal component analysis*. New York: Springer Verlag, 2002.

[48] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[49] M. Garzon, C.-C. Yang, D. Venugopal, N. Kumar, K. Jana, and L.-Y. Deng, *Dimensionality reduction in data science*. Cham, Switzerland: Springer International Publishing AG, 2022.

[50] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[51] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020.

[52] T. pandas development team, "pandas-dev/pandas: Pandas," Feb. 2020.

[53] P. T. Inc., "Collaborative data science," 2015.

[54] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[55] B. Bengfort and R. Bilbro, "Yellowbrick," 2017.

[56] M. K. Pakhira, "A linear time-complexity k-means algorithm using cluster shifting," pp. 1047–1051, 2014.

A

Installation and User Guide

The code used for the project is contained in a zip-archive attached to the thesis, it can also be found in the following repository: https://github.com/asal97/DTE-3900-Master-Thesis. The program consists of a series of Python files and datasets. This appendix describes the installation of packages required to run the program.

A.1 Installation

Once the project archive has been unpacked, one should navigate to the resulting directory in a console window or open the folder in the IDE of choice. In order to access a Jupyter notebook directly from Windows, just type "`jupyter notebook`" in your Windows Command line (CMD). This directory's root contains a file called "requirements.txt" which lists the packages required for the project. In this installation guide, Python 3.7 with pip is assumed to be installed, but other recent versions of Python 3 will also work.

"requirements.txt" contains a list of all required packages. These packages are all available from the Python Package Index, and can be installed with pip, a tool included with Python installations after version 3.4, using the following command:

```
pip install -r requirements.txt
```

A.2 User Guide

This project contains two Jupyter notebooks called Non-temporal analysis and Clustering, which correspond to the code for the sections of this thesis with the same name. Comments are added to the code and sectioned according to their functionality in the notebook. Data files are also included that are read from by the notebooks in order to process each algorithm.

B

Task Description

Included in this appendix is the original task description for this thesis. The following pages contain the document in its entirety, as received at the start of the project period.

UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

Faculty of Engineering Science and Technology
Department of Computer Science and Computational Engineering
UiT - The Arctic University of Norway

Clustering of clinical multivariate time-series utilizing recent advances in machine-learning

Asal Asgari

Thesis for Master of Science in Technology / Sivilingeniør

Background, problem description, scope and limitations

Nordlandsykehuset (NLSH) represented by Ståle Haugset Nymo has initiated a project investigating the possibility to develop an early warning system for patients at the intensive care unit. This system aims to detect anomalies in measurements of vitalia such as blood pressure, temperature, respiration rate, consciousness, blood samples etc, combined with data on patient characteristics like age, sex, clinic, previous hospital admissions and similar. The idea so far is to use unsupervised machine learning on previously recorded time-series on vitalia to produce clusters of patients with similar trajectories. There have been recent advances on this field based on various machine learning techniques including deep learning. For more details, see for instance the review article on the topic from Ruiz, Flynn, Large, Middlehurst, and Bagnall (2021) and the work from Bianchi, Scardapane, Løkse, and Jenssen (2021) accompanied by code freely available for testing and extension. Other approaches also exists with published code, but from a supervised learning perspective (Shukla & Marlin, 2019). However, unsupervised clustering should be possible to extract from these approaches. Python libraries such as sktime (https://www.sktime.org/) and tslearn (Tavenard et al., 2020) also offer clustering of multivariate timeseries and should be considered.
The suggested work on this thesis would be to conduct a systematic review of these various approaches based on adapting and testing them towards data available from NLSH.

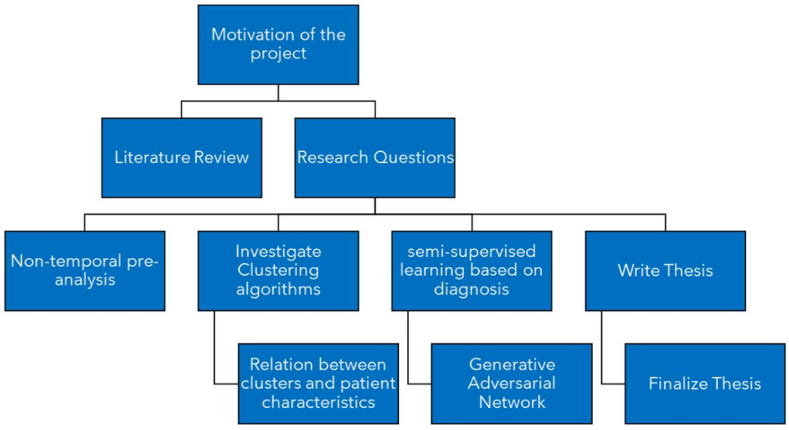
Problem description

Scope and limitations
If the work results in a successful clustering algorithm, capable of capturing characteristic patient profiles according to various clinical conditions that can be verified offline, it will form a basis for an anomaly detection system in future work.

Research questions
Given this background, we propose an investigation towards unsupervised learning of the data consisting of a set of multivariate, irregularly sampled timeseries of variable lengths. Initially, a certain degree of pre-processing might take place in the form of studying only max/min/mean/std.dev parameters and Principal Component Analysis of each time series to attain a certain degree of knowledge about the data.
RQ0: Based on the extraction of maximum, minimum, mean and standard deviation and Principal Component Analysis for each feature, can there be found distinguishable clusters? If so, what conclusions could be drawn from this?
Enriched by the knowledge about the data from this pre-analysis, one might devise certain pre-processing on the data like intelligent imputation values and interpolation of sparse series when the temporal dimension is taken into consideration.
RQ1: On the given timeseries dataset on vitalia from NLSH, is it possible to identify a clustering algorithm able to identify certain clearly distinguishable criteria?
This RQ might be considered from a more theoretical perspective, where one can investigate topological characteristics of the time series being clustered by the algorithm(s) investigated. If clusters are identifiable, a further follow up towards NLSH would be to request more information about outlier's patient IDs. Do these outliers characterize anomalies?
RQ2: Given that RQ1 is fulfilled by utilizing one or several of the algorithms, is it possible to connect these clusters to certain type of patient characteristics?
Finally, there would be of great interest to tie possible clusters to certain characteristics. As per se, an interesting label of each series is the ward. Certain diagnosis given to patients will be

deterministic towards which wards the patient will be admitted to, so one could hope for correlation towards time-series characteristics.
RQ3: Given that a clinician could assist in labelling a set of characteristic timeseries with diagnosis, will it be possible to perform semi-supervised learning based on this?
RQ3.1: Could semi-supervised learning be aided by employing GAN (Generative Adversarial Network) algorithms to generate a larger set of labelled timeseries?
RQ3 is a follow-up of RQ2, but requires time allocated from a clinician to look into the journal data for certain patients which is not yet available to the project. Thus, we have to give this a low priority status.
In general, the priority of the various RQs follow the numbering. RQ1 should of course be possible to answer based on a criterion for clustering quality. We also expect that a fair amount of the hyperparameter space is explored to find optimal clustering for each algorithm investigated. However, if the answer to RQ1 is a clear NO utilizing all the mentioned algorithms/toolkits and search through hyperparameter spaces, RQ2 and RQ3 is not possible to investigate further. However, if the answer is YES for one or some of the toolkits, RQ2 could be explored further, but should be considered as a medium priority task for this thesis. And as mentioned RQ3 must be considered low priority due to the dependency of time allocated from the clinic.

Objectives and methods



To begin, we will explore the theoretical elements of the project by looking at the motivation behind it and why this system is needed by interviewing people involved and reviewing possible literature. In addition, a literature review of relevant multivariate clustering problems can provide insight into how other researchers have tackled these problems while keeping the research questions clear in mind. The purpose of the literature review is to provide insight into the tools and methods that will enable us to achieve our project's goal. Following that, we perform a non-temporal analysis of the data, such as calculating min, max, standard deviation, mean and principal component analysis for each patient's time series and see if these analyses provide us with any insights about how to proceed with the project. After analysing the pre-analysis results, we examine clustering algorithms to determine whether the frameworks and the research based on the referenced papers could be applied to our problem. It is also important to analyse how these algorithms behave in comparison with one another, as well as whether the characteristics of the patients are related to how these algorithms create clusters. The clustering needs to be constructed and evaluated based on established metrics. Optimal number of clusters cannot easily be determined on a qualitative basis, but statistical algorithms like “Gap-statistic” may be tested. To compare the clustering results, we may utilize different clustering evaluation metrics like Davies-

Bouldin Index (DBI) and Silhouette score. In case we had access to characteristic time series with diagnoses, we should investigate whether semi-supervised learning is possible and if we can use generative adversarial networks along with semi-supervised learning for data generation. Further, we will use the results obtained from the methods, strategy, and theoretical parts of the project to write and finalize our thesis.

References

Bianchi, F. M., Scardapane, S., Løkse, S., & Jenssen, R. (2021). Reservoir Computing Approaches for Representation and Classification of Multivariate Time Series. *IEEE Transactions on Neural Networks and Learning Systems, 32*(5), 2169-2179. doi:10.1109/TNNLS.2020.3001377

Li, Y., Ren, Y., Loftus, T. J., Datta, S., Ruppert, M., Guan, Z., . . . Bihorac, A. (2020). Application of Deep Interpolation Network for Clustering of Physiologic Time Series. *arXiv preprint arXiv:2004.13066*.

Ruiz, A. P., Flynn, M., Large, J., Middlehurst, M., & Bagnall, A. (2021). The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery, 35*(2), 401-449. doi:10.1007/s10618-020-00727-3

Shukla, S. N., & Marlin, B. M. (2019). *Interpolation-Prediction Networks for Irregularly Sampled Time Series*. Paper presented at the International Conference on Learning Representations.

Tavenard, R., Faouzi, J., Vandewiele, G., Divo, F., Androz, G., Holtz, C., . . . Woods, E. (2020). Tslearn, A Machine Learning Toolkit for Time Series Data. *Journal of Machine Learning Research, 21*(118), 1-6. Retrieved from http://jmlr.org/papers/v21/20-091.html

Parent project: "A New Approach to Early Warning of Adverse Events" at Nordlandssykehuset (NLSH)

Project plan and risk analysis

The preliminary plan of the project is to investigate and contrast several of the APIs that has been devised for unsupervised clustering of irregularly sampled multivariate time-series. For instance, Li et al. (2020) seem to have performed clustering on vitalia timeseries utilizing the ideas from Shukla and Marlin (2019). However, this research considered vitalia collected over a longer time span and involved serum biomarkers and diagnostic/procedural codes.

The project has a certain degree of risk associated with it, due to the explorative nature of the research. We do not know in advance that clearly distinguishable clusters are going to appear among the time series. If indeed clusters do appear, we cannot

guarantee that they will be connectable to a certain patient characteristic, especially since this will require more data from NLSH and preferably labelled data to some extent.

Thus, if this occurs, a "plan B" can be followed along alternative paths such as:

a) Consider other approaches. Suggested framework so far is the one of Bianchi et al. (2021), sktime and tstime, but one may also consider other frameworks, like the one of Shukla and Marlin (2019). The pyts toolkit could also be a candidate to explore.
b) Develop a new approach more targeted towards the use case at hand. This might involve logic beyond machine learning based on statistical methods as the ones mentioned above. For instance, one may consider frequency of measurement, abrupt changes in certain vitalia etc. to symbolize meaning beyond mere topological attributes. One may also consider handcrafted feature engineering.
c) Consider supervised learning. Other EHR timeseries datasets including labelling might then be investigated.

Task	1.2	15.2	1.3	15.3	1.4	15.4	1.5	15.5
RQ0 (Non-temporal pre-analysis)	X							
RQ1 (Investigate clustering algorithms)	X	X	X					
RQ2 (Patient characteristics)			X	X	X			
RQ3 (Semi-supervised learning)				X	X	X		
Write thesis			X	X	X	X	X	
Finalize thesis								X

Participants

The research is supervised by Associate Professor Helge Fredriksen and Professor Bernt Bremdal. The originator of the project was clinician Ståle Haugseth Nymo from Nordlandssykehuset, which also is the provider of the data in the project. He will participate regularly in meetings with the project team and serve as our main informant in the sense that he can inform us on validity on results and be consulted for feedback on various issues with the data. If RQ3 and partially RQ2 can be reached, he will be the main point of contact.

Deliveries and dissemination

The results of this research will be presented in a delivered report (the master thesis), along with the source code for the developed software.

Dates

Date of distributing the task: <09.01.2023>

Date for submission (deadline): <15.05.2023>

Contact information

Candidate	Asal Asgari aas047@post.uit.no
Supervisor at UiT-IVT	Bernt Bremdal
Co-supervisor at UiT-NT	Helge Fredriksen
Co-supervisor at Nordlandssykehuset	Ståle Haugseth Nymo

General information

This master thesis should include:

- Preliminary work/literature study related to actual topic
  - A state-of-the-art investigation
  - An analysis of requirement specifications, definitions, design requirements, given standards or norms, guidelines and practical experience etc.
  - Description concerning limitations and size of the task/project
  - Estimated time schedule for the project/ thesis
- Selection & investigation of actual materials
- Development (creating a model or model concept)
- Experimental work (planned in the preliminary work/literature study part)
- Suggestion for future work/development

Preliminary work/literature study

After the task description has been distributed to the candidate a preliminary study should be completed within 3 weeks. It should include bullet points 1 and 2 in "The work shall include", and a plan of the progress. The preliminary study may be submitted as a separate report or "natural" incorporated in the main thesis report. A plan of progress and a deviation report (gap report) can be added as an appendix to the thesis.

In any case the preliminary study report/part must be accepted by the supervisor before the student can continue with the rest of the master thesis. In the evaluation of this thesis, emphasis will be placed on the thorough documentation of the work performed.

Reporting requirements

The thesis should be submitted as a research report and could include the following parts; Abstract, Introduction, Material & Methods, Results & Discussion, Conclusions,

Acknowledgements, Bibliography, References and Appendices. Choices should be well documented with evidence, references, or logical arguments.

The candidate should in this thesis strive to make the report survey-able, testable, accessible, well written, and documented.

Materials which are developed during the project (thesis) such as software / source code or physical equipment are considered to be a part of this paper (thesis). Documentation for correct use of such information should be added, as far as possible, to this paper (thesis).

The text for this task should be added as an appendix to the report (thesis).

General project requirements

If the tasks or the problems are performed in close cooperation with an external company, the candidate should follow the guidelines or other directives given by the management of the company.

The candidate does not have the authority to enter or access external companies' information system, production equipment or likewise. If such should be necessary for solving the task in a satisfactory way a detailed permission should be given by the management in the company before any action are made.

Any travel cost, printing and phone cost must be covered by the candidate themselves, if and only if, this is not covered by an agreement between the candidate and the management in the enterprises.

If the candidate enters some unexpected problems or challenges during the work with the tasks and these will cause changes to the work plan, it should be addressed to the supervisor at the UiT or the person which is responsible, without any delay in time.

Submission requirements

This thesis should result in a final report with an electronic copy of the report including appendices and necessary software, source code, simulations and calculations. The final report with its appendices will be the basis for the evaluation and grading of the thesis. The report with all materials should be delivered according to the current faculty regulation. If there is an external company that needs a copy of the thesis, the candidate must arrange this. A standard front page, which can be found on the UiT internet site, should be used. Otherwise, refer to the "General guidelines for thesis" and the subject description for master thesis.

The supervisor(s) should receive a copy of the the thesis prior to submission of the final report. The final report with its appendices should be submitted no later than the decided final date.