TÉCNICO LISBOA

A Deep Temporal Consensus Clustering approach for Amyotrophic Lateral Sclerosis Patient Stratification

Miguel Pego Roque

Thesis to obtain the Master of Science Degree in

Computer Science and Engineering

Supervisors: Prof. Pedro Filipe Zeferino Tomás
Prof. Helena Isabel Aidos Lopes Tomás

Examination Committee

Chairperson: Prof. Diogo Manuel Ribeiro Ferreira
Supervisor: Prof. Pedro Filipe Zeferino Tomás
Member of the Committee: Prof. Nuno Ricardo da Cruz Garcia

November 2022

This work was created using LaTeX typesetting language
in the Overleaf environment (www.overleaf.com).

Acknowledgments

I would like to start by thanking my family for always supporting me in every way and for all the love and encouragement throughout these years. Without my brothers' and parents' love, these last years would've not been possible.

I would also like to thank my girlfriend for all the love and support, for making me feel at home, and for all the help keeping me sane and balanced throughout these last years.

To my paternal grandmother for all the support in many ways and for all the caring and loving, specifically these past years. To my maternal grandparents for always encouraging me, being an example in hard work, and sustaining me in prayer. To my cousins, aunts, and uncles for the continuous encouragement and friendship. To my girlfriend's family for their support and hospitality.

To my friends at Viseu and Lisbon for the much-needed moments of relaxation and fun. To my friends and colleagues here at Instituto Superior Técnico, for the companionship, group work, and mutual pushing forward. A special thanks to José Cruz, Pedro Monteiro, and João Guerreiro, with whom not only I worked but lived for the past five years, and to Tiago Melo, one of my oldest friends.

I would also like to acknowledge my dissertation supervisors Prof. Pedro Tomás and Prof. Helena Aidos, for their insight, support, knowledge sharing, and relentless motivation that has made this Thesis possible. A big, sincere thank you!

This work was partially supported by Fundação para a Ciência e a Tecnologia (FCT) through project AIpALS, ref. PTDC/CCI-CIF/4613/2020.

Lastly, I would like to thank God for His unconditional love in sending His Son. Without Him, none of this would be possible. "Unless the Lord builds the house, its builders work for nothing." - Psalm 127:1a

To each and every one of you – Thank you.

Abstract

Amyotrophic Lateral Sclerosis (ALS) is a neurodegenerative disease that affects upper and lower motor neurons. Currently without a cure, it leads to loss of voluntary muscle movement resulting, in most cases, in death from respiratory failure within 2 to 4 years after the initial diagnosis. Non-invasive ventilation (NIV) is the most effective treatment to increase life expectancy. However, the efficacy of NIV strongly depends on the timing of the beginning of treatment. Due to the many clinical presentations of the disease, this timing is not trivial to determine. Stratification can aid this task by dividing the patients into clinically relevant subgroups according to the disease progression. The prediction of the need for NIV can then be tuned for each sub-group to achieve better performance. Inspired by state-of-the-art approaches, this work introduces a novel patient stratification method, Deep Temporal Encoded Clustering (DTEC), that takes advantage of the temporal dimension of the data to find clinically relevant patients' subgroups. Considering only the disease beginning and using promising Deep Learning mechanisms for temporal encoding, a UMAP encoder, Hierarchical Clustering, and a Consensus Clustering methodology, DTEC found subgroups that present differences across disease progression and clinical presentation. A classification pipeline was also developed and applied to each subgroup. The results were compared with the pipeline without stratification. The classifiers showed improvements, reaching values of AUC up to 83% for some subgroups. Although limited by the dataset, the results showcased the potential of a personalized prognostic prediction solution using patient stratification in the ALS context.

Keywords

Temporal Stratification; Deep Learning; Consensus Clustering; Prognosis Prediction; ALS

Resumo

A Esclerose Lateral Amiotrófica (ELA) é uma doença neurodegenerativa, atualmente sem cura, que afeta os neurônios motores superiores e inferiores. Leva à perda do movimento muscular voluntário resultando, maioritariamente, em morte por insuficiência respiratória 2 a 4 anos após o diagnóstico. A ventilação não-invasiva (VNI) é o tratamento mais eficaz para aumentar a esperança de vida. No entanto, a sua eficácia depende do seu momento de início. As diversas apresentações clínicas da doença tornam difícil a escolha desse momento. A estratificação pode facilitar essa tarefa, dividindo os pacientes em subgrupos clinicamente relevantes de acordo com a progressão da doença. A previsão de necessidade de VNI pode ser então ajustada para cada subgrupo de modo a melhorar o desempenho. Inspirado em abordagens recentes, um novo método de estratificação de pacientes é apresentado, o ”Deep Temporal Encoded Clustering (DTEC)”, tirando partido da dimensão temporal dos dados para encontrar subgrupos de pacientes clinicamente relevantes. Considerando apenas o início da doença e usando mecanismos promissores de Aprendizagem Profunda para codificação temporal, um codificador UMAP, agrupamento de dados hierárquico e uma metodologia de ”Consensus Clustering”, o DTEC obteve subpopulações com diferenças na progressão e apresentação clínica da doença. Uma metodologia para classificação foi também desenvolvido e aplicado a cada subpopulação. Os resultados foram comparados com os obtidos sem estratificação. Os classificadores apresentaram melhorias, atingindo valores de AUC até 83% em alguns subgrupos. Embora limitados pelos dados, os resultados mostram o potencial de uma solução de previsão personalizada usando estratificação de pacientes no contexto da ELA.

Palavras Chave

Estratificação Temporal; Aprendizagem Profunda; Consensus Clustering; Previsão de Prognóstico; ELA

Contents

List of Figures

List of Tables

1

Introduction

Contents

Amyotrophic Lateral Sclerosis (ALS) is a neurodegenerative disease, currently without a known cure. It is characterized by a progressive degeneration of motor neurons, leading to the loss of muscle movement and, eventually, death. The life expectancy after the first symptoms varies from 2 to 4 years [6]. The most common treatments available are four approved drugs [7], which increase life expectancy in a few months [8], and non-invasive ventilation (NIV), which may increase life expectancy by over a year, depending on the starting time and clinical presentation [8]. NIV relevance is even higher when considering that the most common cause of death is due to respiratory failure.

This disease has many clinical presentations, varying not only in symptoms but also in disease progression speed. This fact leads to an increased challenge when understanding the stage of the disease and its evolution, which are crucial factors in deciding whether the patient is eligible for NIV treatment and when to start it. Therefore, a proper stratification of ALS patients is crucial to understand the disease progression and enhance the timing of the NIV start.

Although a few works on ALS patient stratification already exist, they primarily focus on clinical or traditional Machine Learning approaches [9–11] without taking full advantage of the temporal data present in ALS patient records. Most of these approaches use clinical criteria as stratification criteria and then employ traditional Machine Learning algorithms for the classification task.

Deep Learning methods for ALS patient stratification are still under-explored, despite its success in other domains and diseases [1–3, 12]. The nonlinearity of these models might offer new insights and help untangle different subgroups of patients with their own needs. By achieving a better stratification, it is possible to characterize the disease in subgroups of patients in order to uncover the underlying disease patterns. Furthermore, the prognosis can also be improved for a subgroup of patients, and personalized treatment can be offered. This specialized treatment can help improve the daily life of patients suffering from this disease and even extend their life expectancy.

In addition, Deep Learning techniques have been used to encode temporal information, which has shown to be of great value for patient stratification and prognosis in other diseases [1–3, 13]. The encoding of the temporal dimension allows a deeper exploration of the relations regarding the progression of a patient and its expected outcome. This can help determine the best starting time for NIV, ensuring a properly timed treatment and preventing a more exponential and uncontrolled decline of respiratory capacity.

The advances in these areas present a great extent of untapped potential for ALS patient stratification. Specifically, the breakthroughs in stratification using Deep Learning mechanisms offer the possibility to create methods that take advantage of the temporal dimension of the data to build clinically relevant subgroups and produce quality prognoses.

1.1 Objectives

This work aims to explore temporal stratification for ALS patients, specifically using Deep Learning techniques, to characterize underlying subgroups and improve patient prognosis. Using the Portuguese ALS Dataset, it strives to build clinically relevant subgroups of patients, characterize them, and improve the prediction of disease evolution and the consequent need of NIV for these patients.

This work seeks to integrate recent relevant Deep Learning techniques to build a novel method for ALS patient stratification that takes advantage of the temporal evolution of patients' records. We hypothesize that using this information will unveil underlying patterns of disease progression, providing a better characterization of the disease and contributing to a personalized prognosis prediction.

We aim to take a step forward in improving the quality of life of people suffering from this disease by providing clinicians with relevant insights on the different patients' subgroups that can help them decide the best treatment.

1.2 Contributions

The main contributions achieved with this work include the following:

- A literature review on state-of-the-art patient stratification and prognosis prediction methods with emphasis on temporal methods.
- A novel patient stratification method, Deep Temporal Encoded Clustering, specifically tailored for the Portuguese ALS Dataset context, which unveiled distinct patient subpopulations inside our dataset.
- A detailed characterization of the different subgroups of patients obtained, regarding both disease presentation and evolution, highlighting the main clinically relevant differences between them.
- A classification pipeline for ALS patients which showed evidence of the improvement of the prognosis prediction task when applied separately to subgroups of ALS patients with similar characteristics.

1.3 Thesis Outline

This dissertation thesis is organized as follows:

- Chapter 1, the current chapter, provides the motivation, objectives, and contributions of this work as well as the present outline.

- Chapter 2 introduces an overview of Amyotrophic Lateral Sclerosis and a literature review of current state-of-the-art patient stratification and prognosis prediction methods.
- Chapter 3 describes the dataset used in this work, the Portuguese ALS Dataset, and the preprocessing steps applied to it.
- Chapter 4 presents the patient stratification methodology implemented in this work, namely the creation of a novel method, Deep Temporal Encoded Clustering (DTEC).
- Chapter 5 summarizes the experiments and results obtained with the DTEC patient stratification method and the main conclusions and insights reached.
- Chapter 6 contains the description of the prognosis prediction pipeline implemented, as well as the experiences done, results obtained, and main takeaways regarding the classification task.
- Chapter 7 provides a final conclusion to this work and a view into the future work to be explored.

2

Background and Related Work

Contents

This chapter comprises an overview of Amyotrophic Lateral Sclerosis (ALS) disease and a literature review of state-of-the-art patient stratification and prognosis prediction approaches.

It will start by exposing some demographic and clinical information relevant to understand ALS, its heterogeneity, and how it affects its patients. After this section, an exposition of the most widely used clinical criteria for patient stratification is performed, followed by a literature review of state-of-the-art machine learning approaches. Several stratification and prognosis prediction methods will be introduced, not solely restricted to the context of ALS but also regarding it, mentioning the most relevant choices and the consequent results obtained.

In resume, this chapter will contextualize the domain and work already performed regarding the main topics of this dissertation.

2.1 Amyotrophic Lateral Sclerosis - ALS

Amyotrophic Lateral Sclerosis (ALS) is a neurodegenerative disease that mainly affects upper and lower motor neurons. It is characterized by progressive and fatal degeneration in these neurons located in the brain and spinal cord, leading to the loss of voluntary muscle movement and a state of paralysis. Most of the patients die from respiratory failure within 2 to 4 years after the first diagnosis, when the disease starts to affect the diaphragm muscle movement [6].

According to recent data [6, 14], ALS has an average estimated worldwide incidence of 2 per 100 000 inhabitants. Its estimated prevalence is 4 to 8 cases per 100 000 inhabitants. These values are primarily uniform in Western countries, with a higher reported frequency in the Western Pacific region. In Portugal, a prevalence of around 10 per 100 000 inhabitants is estimated, according to recent studies [15].

Regarding demographics, the average onset age is around 55-70 years old, with male predominance. Although the average onset age in clinical records is approximately 58, population-based records reach 70. ALS can occur at any age, and the risk increases with age. As the age increase, the male predominance fades as well, going from a 3/2 male/female ratio to a uniform 1/1 ratio for patients above 70 [16].

ALS has many clinical presentations based on different features such as motor neuron involvement, site of onset, disease focus, and cognitive involvement. These different presentations differ in prognostic, with some offering more optimistic prognostics than others. Due to its variety of forms [14], initial symptoms range from dysarthria and swallowing difficulty to general muscle weakness. As the disease progresses, the most common symptom becomes muscle weakness, which, although generally painless, tends to evolve and spread. When it reaches respiratory muscles, ventilatory support becomes a permanent need for the patients.

We can divide the main risk factors associated with ALS between genetic and external. Genetic

factors are associated with familial ALS, which encompasses 5-10% of total ALS cases [7]. More than 25 genes are identified and studied regarding their correlation with ALS, covering about 70% of all familial ALS cases [14]. On external risk factors, smoking is associated as the most relevant factor, especially in some subgroups such as women after menopause. In addition, other external factors have been studied, including physical activity, environmental toxin exposure, and military service. Nonetheless, the details of their correlations with ALS are still not clear [14].

Although ALS has a fatal prognosis, its survival rate varies significantly. The most common life expectancy at onset is of 2 to 4 years. However, up to 10% of the patients live for more than 8 years [14]. In fact, some patients live for decades, such as former physicist Stephen Hawking, who lived for over 50 years after being diagnosed. These differences are influenced by the clinical presentation and treatments' availability and timing.

Although ALS is currently incurable, some treatments slow its progression, improving patients' life expectancy and quality. Most common treatments include four US FDA-approved drugs (Riluzole, Nuedexta, Radicava, and Tiglutik) [7], that increase life expectancy in a few months, with the only one with documented results being Riluzole, increasing life expectancy in 3-6 months [8, 16]. Another usual treatment is NIV (non-invasive ventilation), which reportedly leads to an increase of 11 to 15,5 months [8] being associated with better survival outcomes specifically for patients older than 65 years [17].

As stated, NIV has obtained the most promising results for most patients. However, understanding the current stage and evolution speed of the disease is essential to infer whether a patient is eligible and what is the best time to start the treatment [18].

2.2 Patient Stratification and Prognosis

2.2.1 Clinical Stratification Criteria

When discussing ALS patient stratification, it is essential to mention the methods developed based on clinical criteria. These methods usually rely on a staging system, a scoring system that characterizes patients based on clinical characteristics. Two of the most relevant and recent staging systems applied to ALS are the King's ALS clinical staging system and the ALS Milano–Torino system (MiToS).

Regarding King's clinical staging, it proposes five disease stages. Stages 1 to 3 are assigned based on the number of regions of the El Escorial central nervous systems affected. Stage 4 is defined using the National Institute for Clinical Excellence Motor Neuron Disease Guidelines. It corresponds to nutritional or respiratory failure, requiring gastrostomy or noninvasive ventilation (NIV), respectively. Stage 5 represents death [19].

On the other hand, the MiToS system defines six stages. While King's clinical system focus on the anatomical spread, MiToS complements it, focusing more on functional impairment. MiToS uses

the Revised ALS Functional Rating Scale (ALS-FRS-R), a questionnaire-based scale, to assess the patient's current state.

Although limited, clinical methods are a valid and relevant approach to predicting the course of ALS. In specific for MiToS, a recent study reported levels of sensibility and specificity of 82% and 63% at 12 months and 71%/68% at 18 months when predicting death, tracheotomy, or more than 23-hour NIV [20].

2.2.2 Machine Learning Stratification and Prognosis Methods

The growing applications of Machine Learning led to several new patient stratification and prognosis methods. Some of these methods achieved excellent results, supporting health professionals in decision-making. Their relevance is justified by the discovery of new clinically relevant insights (and not only by being able to predict a clinician's most likely next step [21]).

Traditional Machine Learning algorithms are divided into supervised and unsupervised techniques. Supervised learning classifiers, such as Random Forests, SVM, and Logistic Regression, use labeled data to learn a model that can be generalized to new examples. These methods can be applied to classify patients' records according to specific criteria.

Regarding such methods, one relevant work is the one of Singh et al. [13]. Their work implemented three different Machine Learning approaches for risk stratification of renal function deterioration.

The first model is a widely used non-temporal method that summarizes the longitudinal information and uses a logistic regression model for prediction. The other two methods are Stacked and Multitask-Temporal. Stacked-Temporal concatenates the variables of all T time windows, mapping each example to an n $\times$ T dimensional vector (with n=number of variables). Multitask-Temporal uses each time window as a separate sub-task of the prediction and averages the results. Both methods also use a logistic regression model for prediction.

An interesting aspect of this work is the definition of time windows, where a trade-off is necessary. A finer granularity may not be relevant, while a too-coarse granularity may cause losing temporal relationships. This study concluded that incorporating temporal information in this context can improve the performance of the prediction models.

In the particular context of the Portuguese ALS Dataset, it is crucial to mention the work of Carreiro et al. [9], which first introduced the use of prognostic models based on time windows to predict a patient's need of NIV. A relevant aspect of its data preprocessing was the creation of snapshots through clustering of temporally-related medical exams. The prognostic models developed included three different time windows (90, 180, and 365 days) to be able to make short, medium, and long-term predictions.

Subsequently, they applied state-of-the-art Machine Learning models at each of the three time-windows in order to predict the requirement of NIV. The results were promising, outperforming classical approaches and pioneering in introducing a non-population-based approach for prognostic prediction in

ALS.

However, this approach does not take full advantage of the temporal information present in the data since it predicts the need for NIV for each snapshot independently, not considering past information.

Another work, built on top of the previous and relevant to mention, is the one of Martins et al. [10]. Their approach used the same time-windows, and consisted of the use of both itemset mining and sequential pattern mining to discover patterns in disease presentation and progression that can then be used as features to prognostic models.

Fernandes, F., Barbalho, I., Barros, D. et al. [22] did a systematic review of the applications of several state-of-the-art Machine Learning Algorithms to biomedical signal data in the context of ALS. This work concluded that SVM, LDA, and artificial neural networks (ANNs) were the most common techniques in diagnosis support, communication, and prediction survival tasks. They were also the ones achieving better performance.

Regarding stratification, traditional Machine Learning methods mostly use unsupervised techniques. Unsupervised learning includes a range of Clustering algorithms such as Affinity Propagation, DBSCAN, K-Means, Mean Shift, and many more. The primary purpose of these algorithms is to automatically group data according to its similarity in an unsupervised manner. The applications to patient stratification are straightforward since they allow the identification of different subtypes of patients, according to the similarity of their records, even when class labels are not available [23].

Deep Learning models are increasingly present in several stratification applications, as well. Unlike traditional Machine Learning techniques, Deep Learning methods can decide on their own whether their predictions are accurate or not and make self-adjustments. Characterized by the use of neural networks, they offer high potential for stratification tasks, mainly due to their non-linearity, obtained through the use of activation functions. Despite the generalized use of linear models for disease progression prediction, studies suggest that ALS progression can be non-linear, differing according to disease severity [24], increasing the relevance of Deep Learning applications in this context.

Moreover, Machine Learning stratification methods can be divided into temporal and non-temporal, considering the existence and exploitation of temporal features, with each group discussed separately next.

2.2.2.A Non-Temporal Stratification Methods

Traditional non-temporal methods are based on unsupervised techniques. Several works, in other contexts besides ALS, used such algorithms: Li and Wong's evolutionary multiobjective clustering method for patient stratification [25]; Khakabimamaghani and Ester's bayesian biclustering method for comparing different patient stratification datasets [26]; the entropy-based consensus clustering method of Liu et al. [27].

Specifically for the Portuguese ALS Dataset, it is pertinent to mention Pires' work [11] on the matter of non-temporal methods for patient stratification and prognostic prediction. Two stratification methods are used. The first uses a clinical measure (Progression Rate), building three Progression Groups (slow, neutral, and fast). The second is based on the creation of patient profiles using traditional clustering algorithms and clinical profiles. After stratification, state-of-the-art Machine Learning classifiers are applied. One of the key conclusions of this study was that using separate prognostic models for each subgroup of patients can improve the prediction results.

The use of deep-learning methods for patient stratification is present in several applications. For example, the ARCAS project [28] is an AI platform built specifically for cancer patient stratification, aiming to assess the right treatment for each type of patient, reducing monetary and temporal costs. Additionally, Kourou et al. [29] highlighted Artificial Neural Networks (ANNs) as a standard for cancer patient risk estimation and stratification. In specific, its application to breast cancer by Ayer et al. [12] produced promising results.

Another work worth mentioning is the one of Lin et al. [30] where the "Skip-Gram" architecture (a method typically used in Natural Language Processing - NLP) is used for pediatric patient risk stratification. The main goal of this study was to compare the results obtained using this technique with other popular risk prediction modeling strategies. Financial and clinical data were both fed to the algorithm to predict next-year hospitalization. The authors concluded that the Deep Learning model outperforms the traditional methods.

2.2.2.B Temporal Stratification Methods

Temporal Stratification methods, similarly to non-temporal ones, can be divided into traditional and deep-learning approaches.

In the ALS field, an interesting work using traditional Machine Learning models is the one of Berry et al. [31]. In this work, the authors compare traditional stratification with stratification using predicted survival in the context of small trials' randomization. Traditional stratification employs riluzole use and bulbar versus limb onset as stratifiers. The alternate method uses a single stratifier, the rank-ordered log-likelihood of predicted survival. A Gradient Boosting Machine (GBM) is responsible for prediction survival. A GBM is a Machine Learning algorithm that builds a new prediction model using an ensemble of multiple weak models, commonly decision trees. The study concluded that the second stratification method achieved better performance in maintaining the balance amongst different trial arms. It is also highlighted the applicability to other neurological diseases and the potential of reducing the size of medical trials, obtaining a more efficient representation.

Another relevant work in this context is the crowd-sourcing approach for the stratification of ALS patients of Kueffner et al. [32]. This study proposes a novel model that integrates several promising

clustering methods (developed by over 30 teams) and analyzes the clusters obtained across those methods. The model proved to be able to achieve clear clinically relevant patient sub-populations and correctly classify new data. By gathering the results obtained by the several approaches and forming a consensus clustering, this approach enables the identification of the best-defined subgroups of patients and the features that distinguish each.

Still concerning more traditional Machine Learning models explicitly applied to ALS, it is interesting to acknowledge the work of Ramamoorthy et al. [33]. In order to model longitudinal clinical data and find patterns in disease progression in ALS patients, the authors developed a method based on a Mixture of Gaussian Processes (MoGP). This approach allowed to produce a non-linear model able to obtain better performance than linear methods. These results can derive from the often non-linear progression of ALS, with stable periods followed by a fast decline. In another subsequent work, the authors experimented using Sequential Pattern Mining to extract features that encode temporal dependency [34]. However, one of the main challenges found in this dataset was the limited time entries for each patient, which negatively impacted the performance of the models implemented.

Regarding temporal Deep Learning approaches, Huang et al. [35] developed a Deep Learning approach to predict patients' lung cancer risk and unveil subgroups with similar risk. They built a neural network, DeepLR, composed of two multilayer perceptrons (MLPs) with two hidden layers each.

The first MLP structure had layers with sizes five and two and the second with 51 and eight. The neural network used a cross-entropy loss function with L2 penalty as the activation function. For weight optimization, the authors used a quasi-Newton method and a stochastic gradient-based method. Moreover, the authors applied a rectified linear units (ReLu) function before the final layer. They also implemented downsampling due to the imbalance of the original data. These techniques led to a final normalized continuous output between 0 and 1.

In addition, the variable "nodule volume doubling time" (VDT) was calculated using the nodule volumes at both S1 and S2 and used as a stratifier. Additionally, the Lung CT Screening Reporting & Data System (Lung-RADS) was used as a third metric for stratification, assuming values 1, 2, 3, 4A, 4B, and 4X. The authors defined stratification criteria for dividing the patients into high-risk (requires an early recall for low-dose CT or diagnostic pathways) and low-risk (can wait until the upcoming scheduled low-dose CT) subgroups.

For the DeepLR, a patient is at high risk if the output is above 0.3 and is a low-risk patient if it is below or equal to that value. For the Lung-RADS criteria, values above or equal to 3 (i.e., 3/4A/4B/4X) corresponded to high-risk patients, while values below (i.e., 1/2) corresponded to low-risk patients. On the VDT approach, a patient is considered high-risk when the VDT value is below or equal to 600 days and low risk otherwise. The results confirmed that DeepLR outperformed the other approaches and accomplished its purpose of estimating cancer incidence probabilities within three years after the S2

scan date.

Madiraju [1] proposed a deep temporal clustering technique that uses a temporal autoencoder (TAE) to map the input temporal data into an effective latent representation. The architecture of this approach, as seen in Figure 2.1, is divided into two main components, the already mentioned TAE and a temporal clustering layer.

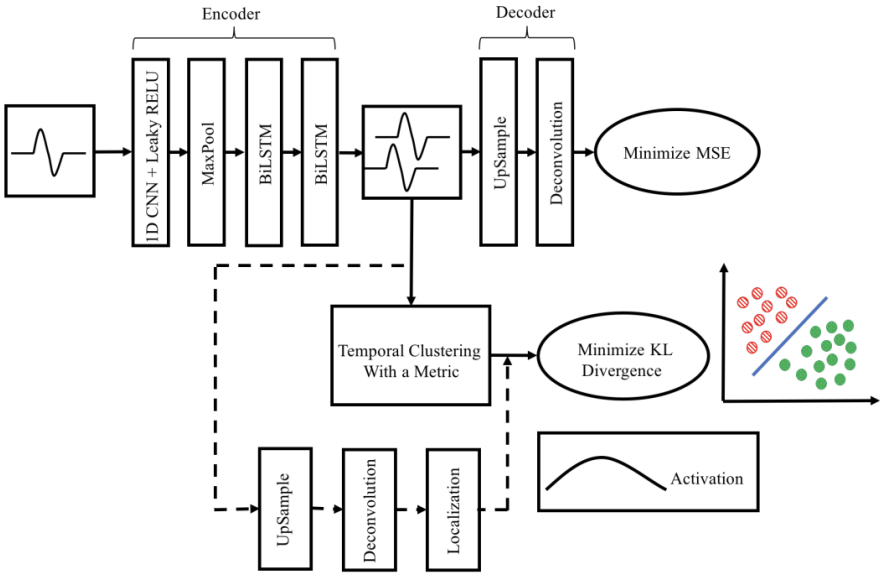


Figure 2.1: Architecture from the approach described in [1].

The TAE consists of an encoder with four layers and a decoder with two layers. The encoder segment's layers can be divided into two primary levels.

A 1D convolution layer (1D CNN), which uses leaky rectifying linear units (L-ReLUs), followed by a max-pooling layer of size P, composing the first level. This level receives the original temporal data, extracts key short-term features, and casts it into a more compact dimension to avoid too-long sequences that can harness the clustering performance.

The second level, constituted by the third and fourth layers, corresponds to Bidirectional LSTMs (BiLSTMs). The BiLSTMs learn temporal changes across both directions in order to collapse the input sequences into every dimension except time. This feature results in the input casting into a smaller latent space.

On the other side, the decoder is responsible for reconstructing the input and is composed of an upsampling layer of size P followed by a deconvolutional layer. Note that the decoder is solely used to learn the best weights for the encoder.

After being transformed by the encoder, the data follows to the temporal clustering layer. This layer initializes k centroids using the output of the TAE. These k centroids are then used in hierarchical clus-

tering with complete linkage over the latent representation feature space. The distance from a record to each centroid is computed using a similarity metric. Four similarity metrics were discussed and experimented, namely, Complexity Invariant Similarity(CID), first proposed by Batista et al. [36], Correlation-based Similarity(COR) introduced by Golay et al. [37], Auto Correlation-based Similarity(ACF) used by Galeano & Peña [38] and the Euclidean distance.

The merged optimization of two cost functions allows the learning in both the 1D CNN and the BiLSTMs. The two cost functions are the mean squared error (MSE) of the input sequence reconstruction from the BiLSTMs output, and the clustering metric of the temporal clustering layer (for example, the Kullback-Leibler (KL) divergence). Optimization using these cost functions respectively guarantees the correct representation of the input sequence after encoding and the separation into k clusters with distinct spatio-temporal behavior.

Finally, the authors use a heatmap-generating network to aid the visualization of the features that contribute most to cluster assignment.

The evaluation of this approach used several publicly available datasets from different domains, with the new model outperforming traditional methods. The authors attributed such success to the fully integrated temporal dimensionality reduction and the work regarding the clustering criterion chosen.

Furthermore, in the field of neurodegenerative diseases, Zhen et al. [2] developed a deep stratification network (DPS-Net) for Alzheimer's patients in the context of neuroimaging data.

The DPS-Net architecture is divided into three sub-components, a feature representation learning network, a subtyping network, and a prediction network, as seen in Figure 2.2.

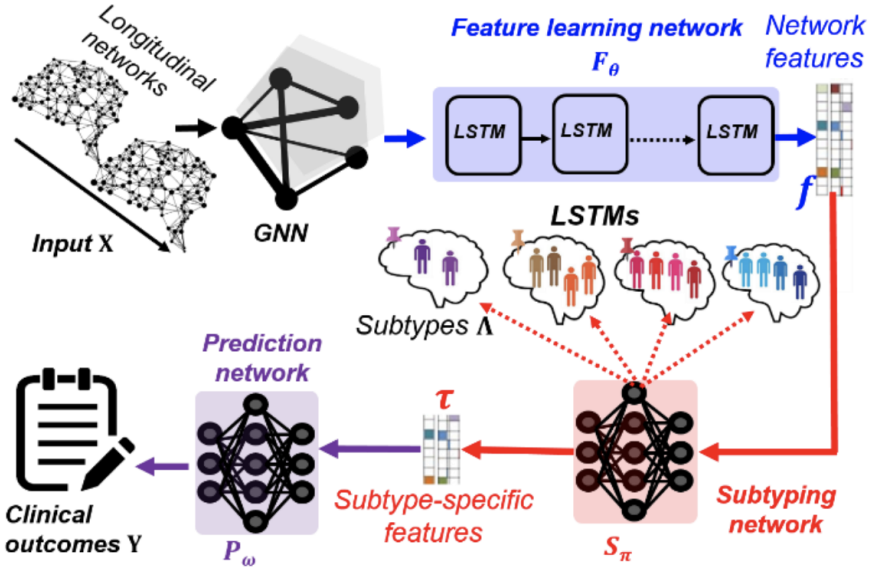


Figure 2.2: Architecture from the DPS-Net approach, described in [2].

The feature representation learning network is composed of an LSTM module responsible for mapping the longitudinal vectors received into a low-dimensional feature space by learning the temporal patterns present in the data.

The subtyping network consists of a fully connected neural network responsible for converting the input and the features obtained into a vector space where features represent the confidence that the record belongs to a latent subtype. Subsequently, subtypes are assigned based on the features (confidences) obtained previously. To incite fuzzy assignment and therefore minimize the possibility of a dominant cluster, this network learns by minimizing the entropy-based regularization.

Lastly, the prediction network, also composed of a fully connected neural network receives the subtype features, and predicts the clinical outcomes scores using the L2-norm loss function.

This approach produced satisfactory improvements in prediction precision. These improvements presumably come from the divergence from traditional unsupervised methods by giving a pivotal role to clinical outcome scores in the stratification process.

Another key work in temporal stratification is the one by Lee et al. [3], not only because of the domain similarity (i.e., chronic diseases patients' stratification) but also for its original temporal clustering approach. In this work, the authors propose a novel architecture that finds clusters and their centroids based not only on the similarity of the temporal observations but also on their future outcomes.

Instead of learning a latent representation of the temporal patterns by using LSTMs or BiLSTMs, this approach focuses on finding a representation based on the similarity of future outcomes. An example given by the authors is the case where we have three patients, A, B, and C. The traditional notion of clustering would group patients A and B. However, if we assume that patients A and C both will have respiratory failure, evidenced by the decrease of FEV1%, then this novel approach will cluster patients A and C instead (Figure 2.3).

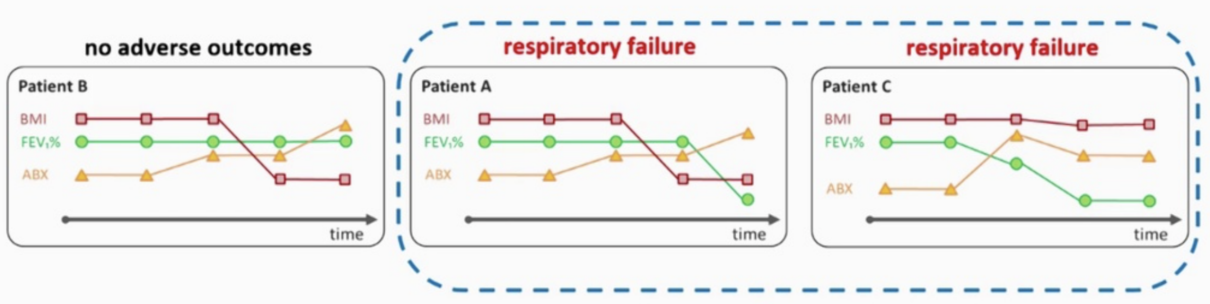


Figure 2.3: Patients grouping in [3], visualization from [4].

In other words, this approach finds clusters and centroids by learning discrete temporal representations that best represent the future outcome distribution. To do so, the author developed a method called AC-TPC, divided into three main components: the actor (composed by an encoder and a selector), the

critic (composed by a predictor), and an embedding dictionary (Figure 2.4).

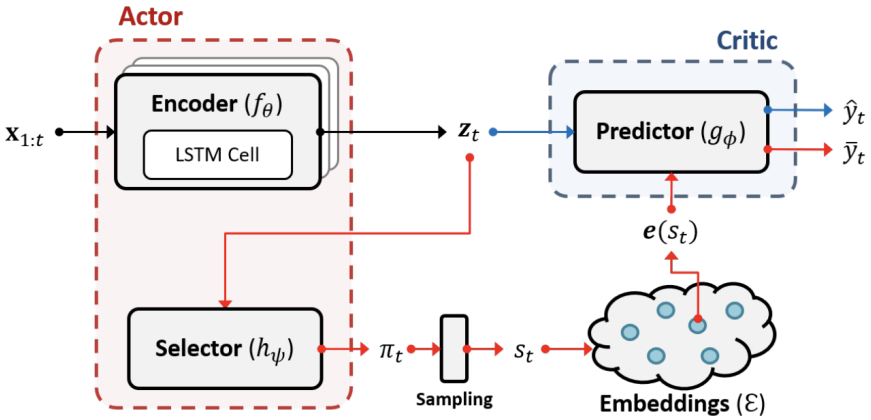


Figure 2.4: Architecture from the approach described in [3].

The encoder network is a Recursive Neural Network (RNN) responsible for mapping, at each instant t, the input sequence $x_{1:t}$ into a latent space, obtaining z_t. Next, the selector network corresponding to a fully-connected network is responsible for mapping z_t into a categorical distribution π_t. The obtained distribution corresponds to the probabilities of a record belonging to each cluster. After this, each record goes through sampling where cluster assignment s_t is done based on the values of π_t. Next, the embedding dictionary is responsible for the allocation of s_t to an embedding $e(s_t)$ that represents the cluster centroid of the initial input sequence $x_{1:t}$. Finally, the predictor network, which is also a fully-connected network, estimates the label distributions $p(y|S_t = s_t)$ if given the cluster centroid $e(s_t)$ or $p(y|X_{1:t} = x_{1:t})$ if given the latent encoding z_t.

The goal of the model is to obtain K predictive clusters where the output label distribution for the elements is homogeneous and thus well represented by the centroid. To evaluate this property, KL divergence is used. This metric can be represented by:

$$KL\left(Y_t\left|\mathbf{X}_{1:t}=\mathbf{x}_{1:t}\right\|Y_t\right|S_t=k\right) \text{ for } \mathbf{x}_{1:t}\in\mathcal{C}(k), \tag{2.1}$$

where $x_{1:t}$ corresponds to a subsequence of records and $Y|S = k$ a random variable for the output given cluster k. This expression is defined as:

$$\int_y p\left(y \mid \mathbf{x}_{1:t}\right)\left(\log p\left(y \mid \mathbf{x}_{1:t}\right)-\log p\left(y \mid s_t\right)\right) dy, \tag{2.2}$$

with $p(y|x_{1:t})$ and $p(y|s_t)$ being the label distributions conditioned on a subsequence $x_{1:t}$ and a cluster assignment s_t respectively.

Note that KL divergence achieves its minimum when the two distributions are equivalent. The model's

objective will then be to identify the set of predictive clusters C so that:

$$\underset{\mathcal{C}}{\text{minimize}} \sum_{k \in \mathcal{K}} \sum_{\mathbf{x}_{1:t} \in \mathcal{C}(k)} KL\left(Y_t \left|\mathbf{X}_{1:t} = \mathbf{x}_{1:t}\right\| Y_t\right| S_t = k) \tag{2.3}$$

In order to quantify the loss and consequently allow the model to learn, the authors introduce three novel loss functions. The first one is Predictive Clustering Loss representing the estimation of the objective (2.3), based on the expectation over cluster assignment. The second is the Sample-Wise Entropy of Cluster Assignment, which is responsible for motivating the selection of a dominant cluster. Finally, the Embedding Separation Loss prevents the collapse of the embeddings into a single point.

This model has reached promising results after being tested on two different datasets, the UK Cystic Fibrosis Registry and Alzheimer's Disease Neuroimaging Initiative. The model outperformed state-of-the-art benchmarks, producing clinically relevant clusters capable of helping the clinicians' decision process.

Another recent work worth mentioning is the one of Landi et al. [5], where the authors developed a Patient Stratification framework, for both multi-disease and disease-specific cohorts of patients. This approach focus on processing heterogeneous electronic health records (EHR) - demographic information, clinical descriptors, medical concepts extracted from clinical notes - to obtain a latent representation of patients and use it to find relevant sub-populations.

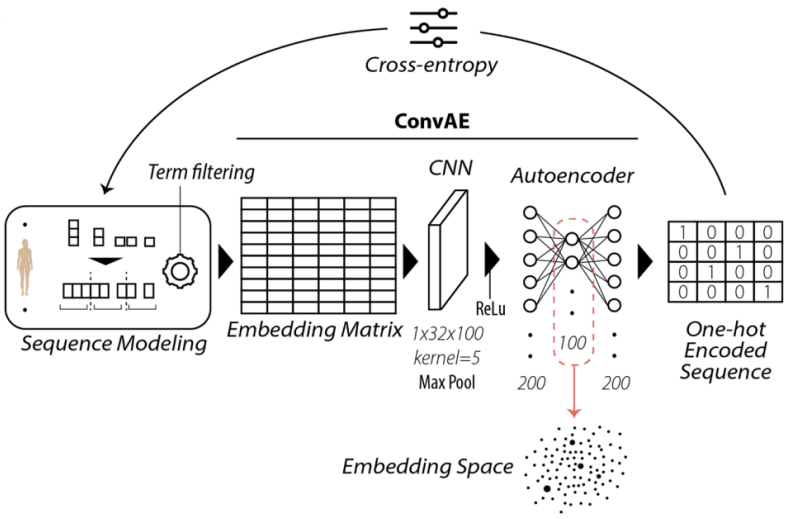


Figure 2.5: Patient stratification framework described in [5].

The framework developed by the authors (Figure 2.5) can be divided into three main stages: Data Preprocessing, Unsupervised Representation Learning, and Clustering Analysis of disease-specific cohorts.

The Data Preprocessing consists of filtering and dropping medical concepts deemed irrelevant (due to the lack or excess of frequency) or redundant (within fixed time periods) and truncating or adding padding as necessary so that all patient sequences of medical concepts have a fixed length L.

At the end of the Preprocessing, all patients are represented by a subsequence s:

$$s = (w_1, ..., w_L) \tag{2.4}$$

where w_n is the n^{th} medical concept.

The next stage is Unsupervised Representation Learning where the subsequences s are encoded in an Embedding space using an unsupervised deep learning architecture, the ConvAE. The ConvAE architecture(Figure 2.6) contains three modules: Embedding, CNNs and Autoencoders(AEs). Each medical concept is embedded in an N-dimensional vector to capture their semantic relationships. These embedded patient sequences are then fed to CNNs and an AE responsible for extracting temporal patterns and learn the embedded representations respectively.

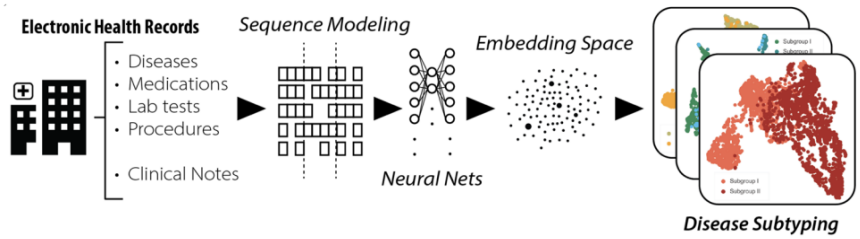


Figure 2.6: ConvAE architecture [5]

The optimization of this network is done by minimizing the AE's reconstruction error, using a Cross-Entropy loss function for that purpose.

This leads to the final stage of the proposed framework, the Clustering Analysis. The patients are first divided by disease, using the SNOMED—CT diagnosis method, in order to find subgroups inside each separate disease. For each disease, the representations learned with ConvAE are then used as input to perform Hierarchical Clustering using Ward's method and the Euclidean distance. Finally, the elbow method is used to find the smallest optimal number of clusters, minimizing the increase of explained variance.

After obtaining the clusters for each disease, the analysis of the medical concepts that distinguish each subcluster is performed. In order to identify the significant and/or unique medical concepts of each disease subgroup, the method considers both its frequency inside the subgroup and in the whole disease cohort. To rank the most relevant contributions, the percentage inside the subgroup is the first to be considered, followed by the percentage across the whole disease group. A pairwise chi-squared

test is used to verify the significance of the results obtained across disease subgroups.

This work also describes a multi-disease analysis by performing Hierarchical Clustering not in disease-specific subgroups but on a multi-disease cohort of patients to test the ability to stratify patients by conditions. A subset of N disorders was chosen, and the same Hierarchical Clustering previously described was performed with N number of clusters. These clusters were evaluated by evaluating if patients with the same condition were grouped together using entropy and purity scores.

The results achieved regarding the clustering tasks were promising. The method outperformed multiple baselines in identifying patients across several complex diseases and obtaining relevant subgroups of patients within a disease (distinguished by medically relevant factors such as disease progression, comorbidities, and symptom severity).

Specifically for ALS, few works on temporal patient stratification using Deep Learning methods were found which represents both a challenge and a motivation for the work to be developed.

Nonetheless, one relevant work to mention is the one of van der Burgh et al. [39] regarding the use of Deep Learning techniques for survival prediction of ALS patients. The records used corresponded mostly to deceased or terminal patients labeled as short, medium, or long survivors. This approach used four Deep Learning networks, two of them based on magnetic resonance imaging (MRI) data, another on clinical data, and a fourth combining the outputs of the previous three. The objective of these networks was to predict the short, medium, or long survival label. Although most of this work focuses on exploiting the MRI data, the clinical network, as the name suggests, uses only clinical data similar to our problem domain. However, the results of this network alone were not that promising, not being able to surpass 70% of accuracy. The improvement in the results was only reached when joining with the information retrieved from the MRI networks.

2.3 Summary

This chapter contextualizes our domain by presenting information about the ALS disease (both regarding demographics, symptoms, and current treatments) and an overview of the current state-of-the-art approaches to patient stratification and prognosis both in ALS and not.

Current clinical methods were presented, with the two main ones being King's ALS clinical staging system and MiToS (which uses the ALS-FRS-R). The limitations of these methods, allied with the potential and exponential growth of AI solutions, motivated the exposition of Machine Learning approaches to both stratification and prognosis. These methods proved to contribute in many different contexts for the optimization of both disease stratification and prognosis prediction.

Patient stratification mainly allows a more personalized prognosis by dividing patients into subpopulations with similar characteristics within and relevant differences between each other. Two different

categories of stratification methods were discussed: those that do not use temporal information and those that do. This literature review delivers a deeper analysis of the last ones since the aim of this work is to take advantage of the temporal dimension of patient records when performing the stratification task.

Not many works were found regarding patient stratification of ALS patients, which motivates the need for such work. In addition, as we will see in the next chapter, the Portuguese ALS data has some particularities that further motivate the need for a specifically tailored solution.

3

Data and Data Preprocessing

Contents

This chapter will describe the dataset used in this work and the main preprocessing choices.

It will start by presenting the main characteristics of the dataset, namely how it was initially obtained, the total number of patients in the cohort, the features and records used, etc.

After this, the main preprocessing steps, common to both the stratification and the prediction tasks, are motivated and described, being Feature Elimination, Missing Value Imputation, Encoding/Discretization, and Normalization.

3.1 The Portuguese ALS Dataset

The dataset used in this work is the Portuguese ALS Dataset. This dataset contains both demographic and clinical data for ALS patients followed at the ALS clinic of the Translational Clinic Physiology Unit, Hospital de Santa Maria, IMM, Lisbon.

Particularly in this work, we will use a version of the dataset after the preprocessing method described by Carreiro et al. [9] has been applied. In this method, patient snapshots are created and used together with time-windows to create learning instances. Due to the fact that a patient may take some days or weeks to perform all prescribed exams of a same appointment, it is relevant to be able to group all these into a single patient snapshot. The author implemented a method for grouping patient records by date, using Agglomerative Hierarchical Clustering. However, with the restrictions that a group cannot have two observations of the same test and that all observations in a group need to have the same value regarding the need of NIV (i.e. cannot have observations where the patient still does not require NIV and observations where it already does).

In order to create learning instances, the author considered the time windows of 90,180 and 365 days and added a feature “Evolution”, that takes the values Y or N, corresponding to whether the patient required NIV within the specified time-window or not. A patient is considered to require NIV, based on questions 10 and 12 of the ALS-FRS-R method - Dyspnea (difficulty breathing) and Breathing insufficiency, respectively. More specifically, if: $P10 \leq 1$ $P12 \leq 3$ or $Old\_P10 \leq 2$ (for the case of the Portuguese ALS Dataset where the patients still use the old scores and not the revised ones, i.e., when the new P10 and P12 are null).

The resulting processed dataset consists of three tables, each corresponding to a set of learning instances, according to a different time-window (90,180,365 days). The difference between the three learning instances corresponding to each time-window is solely the feature “Evolution”. For example, for the same patient record, a time-window of 90 days might not include the date at which the patient required NIV, but when considering 180 days, it might.

Because of the interest in exploring the data’s temporal dimension and the patients’ corresponding evolution, it is not relevant to consider patients with one or very few records. Therefore, the threshold

of at least 3 records was chosen, taking into consideration that a higher value would exclude more than half of the patients (see Figure 3.1).

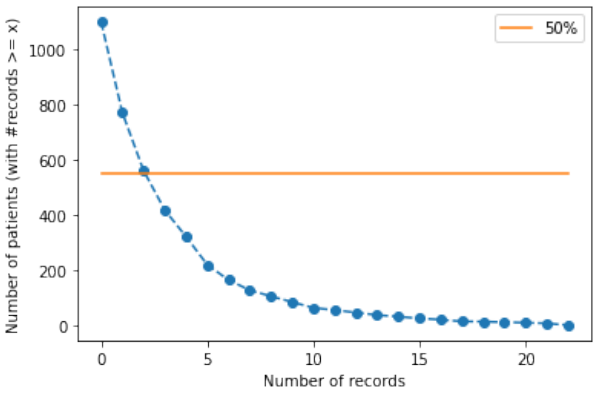


Figure 3.1: Number of Patients with at least X Records

This obtained dataset contains 48 features, both static and temporal, with data from a cohort of 561 ALS patients, totaling 3472 records, and with observations ranging from August 1996 to December 2021.

The data is heavily unbalanced, with 3371/3094/2560 negative records (target class "Evolution" equals "N") and 101/378/912 positive records (target equals "Y") for the time windows 90/180/365 days, respectively.

The Data Preprocessing described next corresponds to steps not specific to the prediction or stratification tasks. Regarding data processing operations that are particular to one of these tasks (i.e. balancing), their descriptions are present in the corresponding methodology section.

3.2 Data Preprocessing

3.2.1 Feature Elimination

The first step of Data Preprocessing was to exclude variables with more than 40% of missing values. This resulted in the elimination of 6 features. The features removed, as well as their percentage of missing values, are present on Table 3.1.

The choice of 40% as the threshold of non-missing values required was supported empirically by experimenting with different values and observing the impact on the subsequent stratification task. If choosing not to remove any feature (equivalent to a threshold of 0%), the data would be composed of a high percentage of imputed values resulting in a large amount of noise that negatively impacted the stratification task. If using a higher threshold (such as 80%) the resulting dataset lacked some features that empirically proved to be important for the stratification task (namely for 80%, the extra features

Feature	Missing values (%)
Airway Occlusion Pressure (%P0.1)	59.05%
Maximal Sniff Nasal Inspiratory Pressure (SNIP)	73.30%
Phrenic Nerve Response Latency	70.18%
Phrenic Nerve Response Amplitude	70.15%
Cervical Flexion	74.34%
Cervical Extension	74.34%

Table 3.1: Features removed and their percentage of missing values.

removed were “UMNvsLMN”, “Limbs_Impairment”, “C9orf72”, “%VC”, “%FVC”, “%MIP” and “%ME”).

Besides these features, the P1-P10 and 1R-3R features were removed since they simply correspond to the questions used to calculate the ALS-FRS and R sub-scores already present in the dataset. The “ALSFRS” feature is also removed due to the dataset already having the revised version of this score - “ALSFRS-R” - making the first one redundant. Lastly, the “criticalR” feature was also removed since it was equal to 0 for all records. The result was the removal of a total of another 14 features, totaling 21 features removed from the dataset.

From the remaining 27 features, it is essential to state that date variables, although used for some tasks(i.e., ordering the records of each patient chronologically, etc.), are not used for stratification or prediction purposes, being discarded before those tasks. In addition, the column identifying the patient each record corresponds to (the ”REF” column) is also not used as a feature for patient stratification or prognostic prediction.

The features remaining, i.e., the ones used in the stratification and prediction tasks, are described on Table 3.2.

3.2.2 Missing Value Imputation (MVI)

The next step of Data Preprocessing is the imputation of the missing values of the remaining features.

The MVI strategy depends on the features’ type. For the continuous numerical variables, missing values are replaced by the mean value.

For temporal numerical discrete variables, forward and backward fill are the strategy chosen to propagate the last valid observation of the patient forward and, for the cases where initial observations are invalid, backward. For the static categorical variables the choice was to use the most frequent value.

For DateTime variables, the rows that have missing values are removed. Once again, these date columns are mainly used for ordering purposes and not for the tasks of patient stratification or prognosis prediction.

Feature	Type	Subgroup	Temporal
Gender	Categorical (Binary)	Demographics	No
Body Mass Index (BMI)	Numerical (Continuous)	Demographics	No
UMN vs LMN	Categorical	Onset Evaluation	No
Age at Onset	Numerical (Discrete)	Onset Evaluation	No
Onset Form (Onset)	Categorical	Onset Evaluation	No
Diagnostic Delay	Numerical (Continuous)	Onset Evaluation	No
Limbs Onset (Limb_O)	Categorical	Onset Evaluation	No
Limbs Impairment	Categorical	Onset Evaluation	No
Limbs Side	Categorical	Onset Evaluation	No
C9orf72 Mutations (C9orf72)	Categorical (Binary)	Genetic	No
ALSFRS-R	Numerical (Discrete)	Functional Score	Yes
ALSFRSb	Numerical (Discrete)	Functional Score	Yes
ALSFRSsUL	Numerical (Discrete)	Functional Score	Yes
R	Numerical (Discrete)	Functional Score	Yes
MITOS-stage	Numerical (Discrete)	Functional Score	Yes
Vital Capacity (%VC)	Numerical (Continuous)	Respiratory Tests	Yes
Forced VC (%FVC)	Numerical (Continuous)	Respiratory Tests	Yes
Max Inspiratory Pressure (%MIP)	Numerical (Continuous)	Respiratory Tests	Yes
Max Expiratory Pressure (%MEP)	Numerical (Continuous)	Respiratory Tests	Yes
Evolution	Categorical (Binary)	Target variable	Yes

Table 3.2: Features of the Portuguese ALS Dataset after Feature Elimination

3.2.3 Encoding/Discretization of Categorical Variables

Since most machine learning algorithms require numerical variables to work, all categorical variables are one-hot encoded using pandas function $pd.get\_dummies()$. The one-hot encoding transforms each categorical column with N unique values into N binary columns. For each row, the column corresponding to the original column value has the value 1, and all others are represented by a 0.

This step results in the creation of 14 new columns, replacing the original 5 categorical columns ("UMNvsLMN", "Onset", "Limbs Onset", "Limbs Impairment" and "Limbs Side") and leading to a processed dataset with 38 features.

In addition, the "Gender" and "C9orf72" categorical boolean variables, which were previously represented by strings "male"/"female" and "yes"/"no", respectively, were converted to 1/0 values.

3.2.4 Normalization

The last step of Data Preprocessing is the Normalization of all variables except the identifier ("REF"), the target ("Evolution"), and the DateTime variables. The strategy chosen was to normalize the data between [-1,1].

Normalization mitigates possible implications on ML algorithms resulting from not all features having the same range of values (i.e., a feature with larger values being considered more important than others).

The choice of the [-1, 1] range is a result of the $tanh$ activation function, which returns values between [-1,1], used on the temporal encoding network, and that will be discussed in more detail on following sections.

3.3 Summary

In this chapter, a brief overview of the Portuguese ALS Dataset is presented, as well as the Data Preprocessing choices.

This chapter describes the features used in this work, regarding their type, subgroup, and whether they are temporal or not. The features of the original dataset selected totalize 27 columns, representing both temporal and static variables, of several types (categorical, DateTime, and numerical - both discrete and continuous) and include both Demographic data, Onset Evaluations, Respiratory Tests and ALS Functional Scores (among others).

The common Data Preprocessing pipeline applied to the data, before being fed to the patient stratification and prognosis prediction methods, is also presented, consisting on three main stages (MVI, Encoding/Discretization and Normalization). After the preprocessing steps, the dataset contains a total of 38 features.

The particularities of the dataset described in this chapter, such as the limited number of patients and records per patient, and the high imbalance might motivate the need for specifically tailored solutions.

Having described the data that will be used in this work and the preprocessing steps used to treat it, the next chapters will focus on the methods and results obtained for patient stratification and prognosis prediction.

4

Stratification Methodology

Contents

Current state-of-the-art stratification methods for ALS patient stratification are scarce. Although promising, other stratification methods present some key differences, especially regarding the data used and how it affects certain architectural choices.

Some of the current state-of-the-art stratification methods have a significant focus on dimensionality reduction mechanisms due to the high dimensionality and size of the data available. For example, the data used in the ConvAE method [5] was composed of a cohort of 1,608,741 patients, with an average of 88.9 records per patient. This reality highly contrasts with the Portuguese ALS dataset, which, as stated on Chapter 3, includes 561 patients, most of which have less than 5 records. Furthermore, the number of features of the Portuguese ALS dataset used in this work totals 38 features, contrasting with the 100-dimensional embeddings obtained in ConvAE for the 31,659 medical concepts used. The contrasting small size and dimensionality of the ALS dataset implies a cautious use of dimensionality reduction. Another example still related to the differences across datasets is the use of BiLSTMs on the DTC method [1] to learn temporal changes in both directions. Since the temporal dimension of each patient is short in the Portuguese ALS dataset, the use of BiLSTMs might increase the computing time without obtaining a relevant gain on the quality of the encoded representations. Simple LSTMs can be a more reasonable choice.

One of the methods considered more promising regarding patient stratification and prognosis prediction was AC-TPC [3]. However, some limitations were found when applying it to the ALS context.

Firstly, contrary to the original AC-TPC approach, the original goal of clustering, in this ALS context, is not to group patients according to a final labeled prognosis but to find unlabeled medically relevant groups of patients with similar evolution. Even if the goal was, in fact, to cluster patients according to the final label, in the ALS dataset, this label has some particularities. In the original context of AC-TPC, the label corresponds to possible prognoses amongst a set of conditions. In the Portuguese ALS dataset the label corresponds to if a patient requires or not NIV at a given time-window. This label brings an additional restriction: a patient cannot regress to not needing NIV if it was already required in previous appointments. The AC-TPC method assigns the clusters for each time-step and does not consider this restriction, resulting in the same patient being able to transition, across time-steps, between the clusters (which in this case are two, representing whether the need or no need for NIV). These transitions do not make sense from a medical point of view.

These concerns motivated the implementation of a new tailored architecture focused on encoding the temporal information present in our data and grouping the patients considering these representations. This chapter discusses and motivates the details of this solution.

4.1 General Methodology

The Deep Temporal Encoded Clustering method (DTEC) was developed to capture the temporal dimension of patients' records and use it to obtain medically relevant groups.

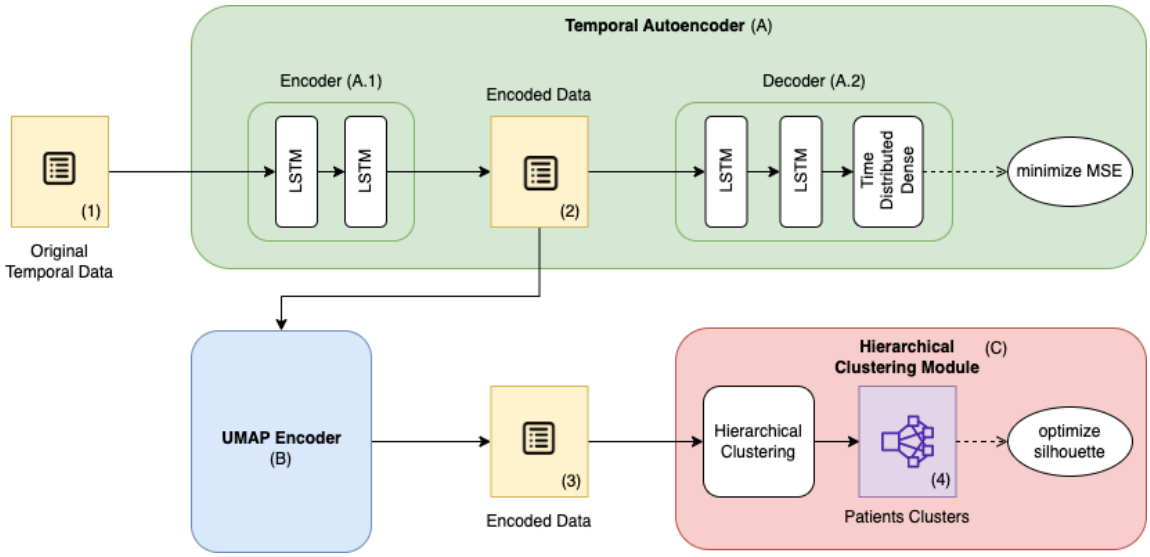


Figure 4.1: The proposed DTEC Architecture

The DTEC architecture is composed of three main modules, as shown on Figure 4.1. The Temporal Autoencoder - TAE (Figure 4.1-A), the UMAP Encoder (Figure 4.1-B) and the Hierarchical Clustering Module - HCLUST (Figure 4.1-C).

The TAE receives the original temporal data and returns a latent space corresponding to the encoding of the temporal information of each patient. The UMAP Encoder receives the encoded data and, using graph representations, obtains a new optimized encoding. Lastly, the HCLUST module is responsible for grouping the patients into subgroups based on previously obtained representations.

This architecture was inspired by promising literature architectures [1, 5] but adapted and optimized to the specific context of ALS and the Portuguese ALS Dataset. One key difference is the use of UMAP to optimize the encoding space used for the clustering without reducing the dimensionality of the representations. More regarding this module will be presented in its specific subsection further in this chapter (see Section 4.4).

The use of non-deterministic methods, both in the TAE and the UMAP Encoder, results in variations in the representations obtained. This leads to a challenge in terms of the stability of choosing the best number of clusters. Since the encoding space varies, the ideal number of clusters also changes from run to run. To improve this aspect, a consensus clustering methodology was implemented, inspired by state-of-the-art approaches like the works of Kueffner et al. [32], Liu et al. [27] and Fred et al. [40].

In the following subsections, the three main modules will be discussed in more detail as well as the

consensus clustering methodology.

4.2 Additional Data Preprocessing

In order for the data to be fed to the DTEC model, more specifically to the TAE, some preprocessing is required.

As seen on Figure 4.2, initially, the records' shape is:

$$\sum_{n=1}^{N} R_n \times F, \tag{4.1}$$

where $\sum_{n=1}^{N} R_n$ is the the total number of records, obtained by summing the number of records R_n of each patient n, and F is the total number of features (columns excluding the target).

However, to be used as input of the TAE, the data needs to be grouped by patient, being shaped as a matrix of dimension:

$$N \times M, \tag{4.2}$$

where N is the total number of patients, and M is a matrix with dimension $R \times F$ (where R is a fixed number of records per patient, and F is the total number of features - columns excluding the target).

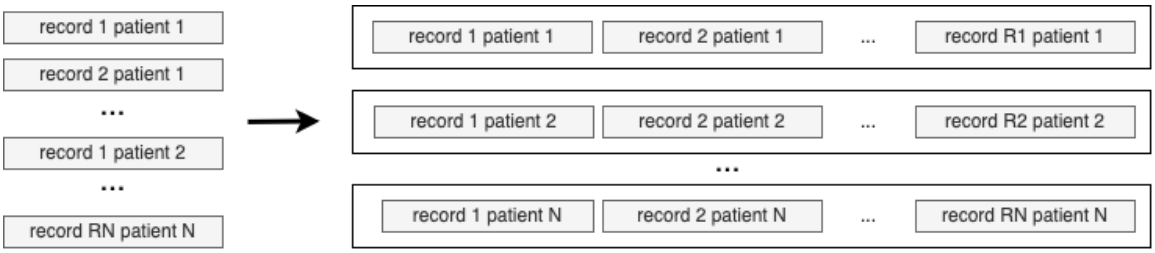


Figure 4.2: Data Transformation

The TAE requires a fixed length for each patient vector, i.e., a fixed number of records to be considered (R). It was decided to fix this length at 3, which means only using the first 3 records of each patient. Note that, as mentioned on Chapter 3, less than half of the patients have more than 3 records. If the number of records were not limited/truncated, or a higher value was chosen, a lot of empty/imputed entries would be present, which could negatively impact the representations obtained and, consequently, the stratification task.

Considering only the beginning of the disease (i.e., the first 3 records) is also interesting from a medical point of view since it can be relevant to identify sub-populations as soon as possible to proceed to a more personalized treatment.

For simplicity, the notation used for the shape of the input data in the rest of this chapter is:

$$n\_patients \times timesteps \times n\_features, \tag{4.3}$$

with, $n\_patients$ corresponding to the total number of patients, $timesteps$ corresponding to the number of records being considered (i.e., the first 3 records), and $n\_features$ corresponding to the to the total number of features (columns excluding the target).

4.3 Temporal Autoencoder - TAE

The Temporal Autoencoder is responsible for, given the original temporal data (after the preprocessing described in the previous section), learning a lower dimensionality space representation by capturing the most relevant information out of the data (temporal or non-temporal patterns, correlations, etc.).

A more detailed scheme showcasing this module and including the dimensions of each layer and the inputs and outputs obtained can be seen on Figure 4.3.

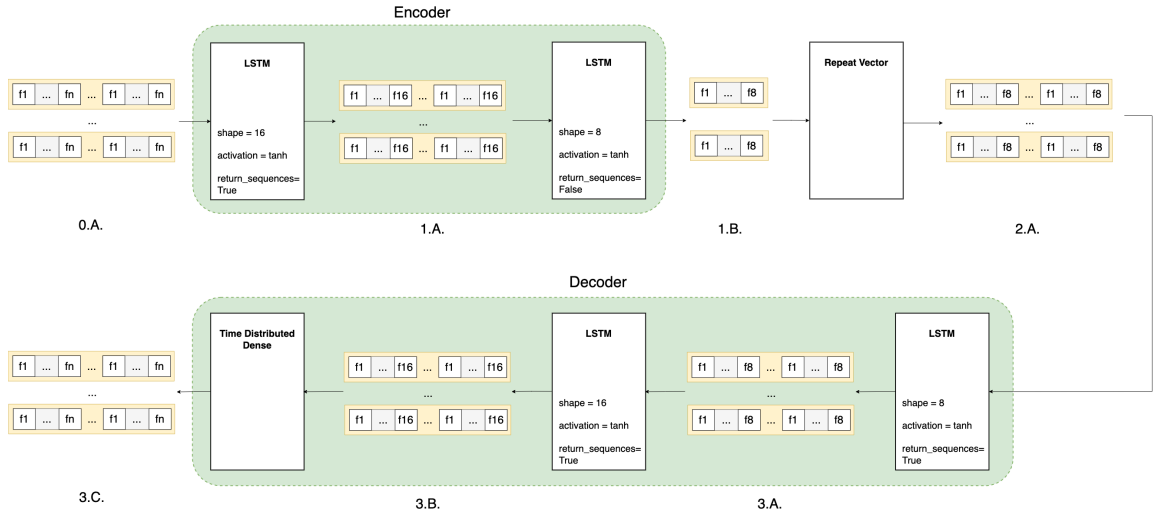


Figure 4.3: TAE detailed architecture

As an autoencoder, the network can be divided into two primary submodules, the encoder and the decoder, linked by a Repeat Vector layer.

The encoder consists of two LSTMs layers. It receives the original data and, using LSTMs, learns temporal changes across the records. The first LSTM layer encodes the original temporal data shaped $(n\_patients \times timesteps \times n\_features)$ (Figure 4.3 - 0.A.) into a 16 feature dimension space shaped like $(n\_patients \times timesteps \times 16)$ (Figure 4.3 - 1.A.). The second LSTM layer has the particularity of having the argument $return\_sequences = False$, which results in the projection to a lower dimension space, removing the temporal dimension. This parameter causes each cell to emit a single signal instead of

a signal per time-step. The LSTM also reduces the feature size from 16 to 8, resulting in an encoding shaped as $(n\_patients \times 8)$ (Figure 4.3 - 1.B.). The activation function used for both layers is $tanh$ defined as:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4.4}$$

A Repeat Vector layer connects the encoder to the decoder by repeating each embedding until the number of time-steps is reached. By doing so, it increases the dimension of the data again with output of this layer being shaped as $(n\_patients \times timesteps \times 8)$ (Figure 4.3 - 2.A.).

Similarly to the encoder, the decoder is composed by two LSTM layers but appearing in the reverse order and with the addition of a Time Distributed Dense at the end of the last one. The first LSTM layer receives the output of the Repeat Vector layer and outputs a $(n\_patients \times timesteps \times 8)$ representation (Figure 4.3 - 3.A.). The second LSTM layer, on the other hand, outputs a $(n\_patients \times timesteps \times 16)$ representation (Figure 4.3 - 3.B.). Finally, the Time Distributed Dense layer makes the final output shape match the original input shape, $(n\_patients \times timesteps \times n\_features)$ (Figure 4.3 - 3.C.). Once again, both LSTM layers use the $tanh$ activation function.

The learning process of an autoencoder is done through the minimization of a loss function that penalizes how much the representation obtained by the decoder ($\hat{y}$) differs from the data received by the encoder (y). The loss function used was the mean-squared error (MSE) defined as :

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 \tag{4.5}$$

By minimizing MSE, we minimize the difference between the reconstructed and the original data, improving the quality of the embeddings obtained. The decoder is used solely as part of the training of the network. It is the output of the encoder that is used in the following steps.

The optimizer used to train our network is the Adam optimizer. Adam [41] is an adaptive learning rate optimization algorithm widely used for optimizing deep neural networks. It combines the best aspects of RMSprop and Stochastic Gradient Descent with momentum to obtain improved results in a faster computation time.

Different configurations for the network were tested, with the combinations of hyperparameters being empirically tuned to maximize the silhouette (which is the metric used to evaluate the quality of clusters, defined in Section 4.5) obtained by the HCLUST module. Furthermore, some configurations performed worst or increased the computation time without practical advantages on the silhouettes obtained and were therefore excluded. A resume of the configurations experimented is present on Table 4.1:

Table 4.1: Resume of the Configurations for the TAE Experimented

Configuration Change	Values experimented	Value chosen
Layers of the encoder	- 2 LSTM layers - 1 Dense layer + 2 LSTM layers - 3 LSTM layers - 2 BiLSTM layers	2 LSTM layers
Shapes of the LSTM layers	- Layer 1: 32, 24, 16, 12, 8, 6 - Layer 2: 16, 12, 8, 6, 4, 3, 2	- Layer 1: 16 - Layer 2: 8
Activation functions	- tanh - relu - sigmoid	tanh
Loss function	- mse - kl_divergence	mse
Optimizer	- adam - rmsprop	adam

4.4 UMAP Encoder

Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) [42] is a recent manifold learning method that uses fuzzy topological structure to find an accurate representation of the data without losing too much information. Its advantages, when compared to other embedding methods such as t-SNE, include high scalability and good preservation of both local and global structure within the data.

UMAP uses a weighted k-neighboured graph to arrange the data in a low-dimensional manifold. It starts by constructing a high-dimensional graph representation, building what is called a ”fuzzy simplicial complex”. This includes using a weighted k-neighboured graph, with each edge weight being the likelihood that the points are connected. This representation is then projected into lower dimensions, optimizing its similarity with the high-dimensional layout and obtaining a new embedding.

The UMAP Encoder module uses the UMAP method to encode the data into a new latent space using non-linear and non-stochastic transformations. The objective is to obtain an optimized, more ”clusterable” representation of the data.

The use of UMAP to optimize clustering is backed up by current state-of-the-art works such as the works of McConville et al. [43] and Allaoui et al. [44], which state the use of UMAP as a way of improving clustering accuracy. UMAP’s preservation of both the local and global structure potentiates the learning of an optimized, more clusterable embedding manifold, which leads to better clusters. Specifically, in the work of Allaoui et al. [44], UMAP proved to improve performance in different clustering algorithms,

from density-based approaches, such as HDBSCAN, to hierarchical clustering methods, such as the one used in the DTEC architecture, Agglomerative Clustering.

Regarding its parameters, the ones considered more relevant are n_neighbors, n_components, and min_dist.

The n_neighbors parameter is responsible for balancing local and global structure in the data by defining the number of neighbors used by UMAP when learning the low dimensional layout. A lower number will result in UMAP using only the closest neighbors obtaining representations focused on the local structure, while a larger number will result in a better reflection of the global structure. The value chosen was 50 neighbors, considering the number of patients of 561 and the experiments done with other values, further explored on Chapter 5.

Regarding n_components, although, traditionally, UMAP is used to reduce the dimension of the latent space, we maintain the latent space dimension obtained by the TAE (n_components=8). The rationale behind this choice is the fact that the TAE is already reducing the dimensionality of the data, and further reduction could lead to loss of information (considering all the particularities of the dataset). This was experimentally observed by the lack of improvement in the silhouette score of the clusters obtained with HCLUST when experimenting with lower values.

The min_dist parameter represents the minimum distance required for embedded points. Larger values will result in less packed points, while smaller values will result in more packed points. As stated in the UMAP documentation [45], a small value in this parameter can be beneficial for performing clustering. The value chosen corresponds to the default value of 0.1, which is already a small value. Experiments were done with a set of values for this parameter, but the improvements were not significant when compared with the default value.

The resume of the different configurations experimented for this module is present on Table 4.2. Similarly to the TAE, the choices were made by considering the improvements on the silhouette score of the clusters later obtained with HCLUST but also considering if the increase of runtime for some configurations was or was not accompanied by significant improvements. For example, using ParametricUMAP instead of UMAP increased the algorithm's runtime and showed no significant improvements.

4.5 Hierarchical Clustering (HCLUST)

The HCLUST module is the third and last step of the proposed DTEC method. To stratify the patients, represented by the latent space obtained previously, we use a Hierarchical Clustering method, more specifically Agglomerative Clustering using Ward linkage and Euclidean distance as the affinity metric.

Since the number of clusters is not identified *a priori*, clustering methods that require pre-specifying it, such as KNN, were excluded.

Table 4.2: Resume of the Configurations for the UMAP Encoder Experimented

Configuration Change	Values experimented	Value chosen
n_neighbors	15 (default), 30, 50, 100	50
n_components	8 (no dimensionality reduction), 4, 2 (default)	8
min_dist	0.1 (default), 0.05, 0.01	0.1
spread	10, 5, 3, 1 (default)	1
local_connectivity	1 (default), 2, 15, 20	1
repulsion_strength	1 (default), 2, 5, 10, 20	1
metric	euclidean (default), manhattan	euclidean
UMAP vs ParametricUMAP	UMAP, ParametricUMAP	UMAP

Density-based methods that do not require pre-specification of the number of clusters, like HDBSCAN, were considered. However, due to the high heterogeneity of the data and, consequently, of the embeddings, these methods obtained good silhouette scores only when considering a very high number of clusters. A large number of clusters in such a small dataset does not hold relevant medical value, so these methods were discarded.

Agglomerative Clustering is a hierarchical clustering algorithm that uses a bottom-up approach to group the data. It starts by making each point a cluster and then recursively grouping the two nearest clusters into one larger cluster. Some of the advantages of using a hierarchical approach are that it also does not require initializing the number of clusters, and instead, it provides a dendrogram to visualize the clusters allowing to cut it at any desired number of clusters. In addition, the dendrogram itself provides a way of finding the optimal number of clusters acting as one of the main advantages of this method when compared with others such as Gaussian Mixtures Models (GMM).

In order to evaluate the optimal number of clusters and its quality, the dendrogram was used allied with the silhouette score.

The silhouette score is a metric of both cohesion and separation, evaluating how much a cluster point is similar to its cluster when compared to its similarity to other clusters. It ranges from -1 to 1, with negative values representing a wrong cluster assignment, values near 0 representing overlapping clusters, and values closer to 1 representing clearly distinguished clusters. It is calculated using the following formula:

$$silhouette = \frac{b - a}{\max(a, b)} \tag{4.6}$$

where, a is the mean intra-cluster distance and b is the mean nearest-cluster distance.

The linkage method chosen was Ward, a method for hierarchical clustering analysis which is consid-

ered to perform well on noisy data and is widely used in applications with real-world data. It works by choosing the clustering step that minimizes the increase of the error sum of squares resultant of merging two clusters. Other methods, such as Single and Complete linkage, were tested but produced worse silhouette scores.

4.6 Robustness Improvement using Consensus Clustering

As already stated, hierarchical clustering methods provides a dendrogram to visualize the best number of clusters, allowing one to cut the dendrogram at any K number. However, due to the varying representations obtained, the approach still faces issues regarding stability, with the optimal value K changing across runs.

Consensus Clustering is an approach that tackles this challenge by providing a robust methodology to evaluate the stability of the clusters obtained and choose the clustering parameters, such as K. Instead of using only one iteration, Consensus Clustering considers the results of multiple iterations across different latent spaces and with different numbers of clusters.

Traditionally, consensus clustering is applied to different randomly obtained subsets of the original data. However, in our domain, this variability is introduced not by considering different subsamples but by considering the different latent spaces obtained across runs (due to the non-deterministic nature of the TAE and UMAP).

The methodology used was based on evidence accumulation clustering approaches such as the one proposed by Fred et al. [40]. For each run, clusters are registered in an incremental co-association matrix C, varying the K number of clusters from 2 to 4. We increment this one same co-association matrix across all runs and the number of clusters obtaining the following matrix:

$$C = \begin{bmatrix} n_{11} & n_{12} & n_{13} & .... & n_{1P} \\ n_{21} & n_{22} & n_{23} & ... & n_{2P} \\ n_{31} & n_{32} & n_{33} & ... & n_{3P} \\ .... & ... & ... & ... & ... \\ n_{P1} & n_{P2} & n_{P3} & ... & n_{PP} \end{bmatrix} \tag{4.7}$$

where, n_{xy} is the number of times patient x was in the same cluster as patient y, and P is the total number of patients;

Then, C is divided by $N \times n_{runs}$, to obtain a similarity matrix S:

$$S = \frac{C}{N \times n_{runs}} = \begin{bmatrix} 1 & p_{12} & p_{13} & .... & p_{1P} \\ p_{21} & 1 & p_{23} & ... & p_{2P} \\ p_{31} & p_{32} & 1 & ... & p_{3P} \\ .... & ... & ... & ... & ... \\ p_{P1} & p_{P2} & p_{P3} & ... & 1 \end{bmatrix} \tag{4.8}$$

where, C is the co-association matrix, N is the number of parameters tested (in this case three - K

number of clusters varying from 2 to 4), n_{runs} is the total number of runs, p_{xy} is the percentage of times patients x, and y were grouped at the same cluster, and P is the total number of patients;

We subtract this similarity matrix from an all-ones matrix obtaining a distance matrix M:

$$M = \mathbf{1} - S = \begin{bmatrix} 0 & d_{12} & d_{13} & .... & d_{1P} \\ d_{21} & 0 & d_{23} & ... & d_{2P} \\ d_{31} & d_{32} & 0 & ... & d_{3P} \\ .... & ... & ... & ... & ... \\ d_{P1} & d_{P2} & d_{P3} & ... & 0 \end{bmatrix} \tag{4.9}$$

where, S is the similarity matrix, d_{xy} is the "distance" between patient x and y, and P is the total number of patients.

Note that M is a symmetric square matrix with a diagonal of all zeros.

The distance matrix M is later used in Hierarchical Clustering to obtain a consensus dendrogram.

The best number of clusters is chosen by considering the silhouette score, dendrogram, and, additionally, a heatmap representation based on the distance matrix.

This last representation was obtained by, for a certain K number of clusters, ordering the distance matrix in terms of both the axis and then plotting a heatmap of the resulting ordered matrix. This process produced visualizations, such as the one on Figure 4.4

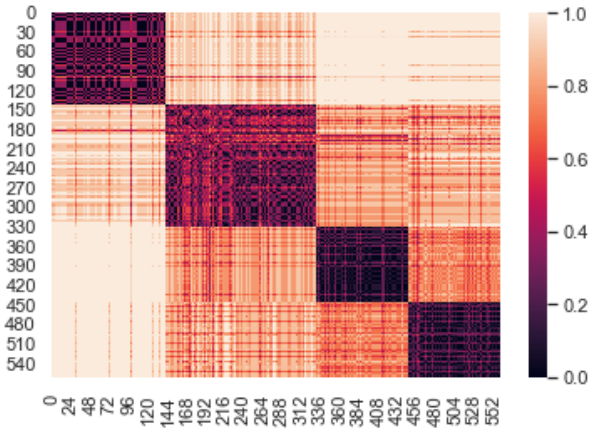


Figure 4.4: Ordered distance matrix heatmap

By comparing the heatmaps for different K clusters, we can visually observe which ones seem to obtain more well-defined clusters. This comparison will be made when choosing the best number of clusters for each configuration on Chapter 5.

Consensus clustering was applied to different configurations in this work, namely, considering the complete feature set, considering only temporal features, and considering only the first record of each patient.

4.7 Summary

This chapter presents an overview of the Stratification methodology proposed. It presents a novel Deep Temporal Encoded Clustering (DTEC) method inspired by current literature but focused on the context of this work and the specificities of its dataset.

The architecture of this approach is presented, both in general and for each module in particular. The three main modules that constitute DTEC are the Temporal Autoencoder (TAE), the UMAP Encoder, and the Hierarchical Clustering module (HCLUST). The architectures of each module are motivated and described in detail. Additional data preprocessing required is also described.

A concern regarding the clustering robustness is raised, namely, deciding the best number of clusters. This concern led to the introduction of a Consensus Clustering strategy, which was also implemented in this work.

The results obtained with DTEC with and without Consensus Clustering are described in the next chapter.

5

Stratification Results

Contents

This chapter presents the results obtained with the DTEC approach with and without Consensus Clustering. Different configurations were experimented with regarding the data features received by the Consensus Clustering methodology, namely ”Complete Feature Set” (where all features were used), ”Temporal Only” (where only temporal features were used), and ”First Record” (where all features were used but only considering the first record of each patient).

For each configuration experimented, the choice of the best K number of clusters is justified, and the clusters are characterized. This characterization presents the critical differences across their features' distribution and evolution.

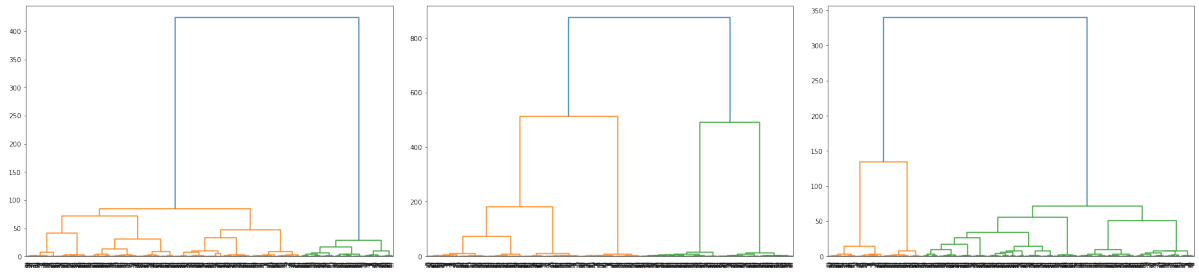
The results across configurations are compared both considering the silhouette and the corresponding clusters characterizations in order to justify the choice of one as the best/most clinically relevant.

5.1 Deep Temporal Encoded Clustering

5.1.1 Determining the Optimal Number of Clusters

In order to find the optimal number of clusters for each run, the corresponding dendrogram, obtained using Ward linkage, is used together with the silhouette score. To find the best number of clusters using the dendrogram it is necessary to identify the largest vertical distance between nodes and count the number of lines intersected if we draw an horizontal line intersecting that. However, not always is easy to identify this distance and, the optimal number of clusters achieved with this method not always presents the best silhouette score.

The differences between runs can be seen on Figure 5.1, where three different dendrograms are presented, corresponding to different independent runs.



(a) **(b)** **(c)**

Figure 5.1: Dendrograms for 3 independent Runs

Each dendrogram points to a different number of optimal clusters. On the left (Figure 5.1(a)) the best number of clusters is 2 with a silhouette score of 0.82. On the middle (Figure 5.1(b)) , the optimal number of clusters is 4, with a silhouette of 0.88. On the right (Figure 5.1(c)), the optimal number is

3, with a silhouette score of also 0.82. If we chose 4 as the optimal number of clusters, the silhouette results for the left and right dendrograms would decrease to 0.60 and 0.50, respectively.

Therefore, in order to aid the choice, the silhouette scores for different K numbers of clusters are also observed.

A total of 100 runs was performed, registering the silhouette score with K number of clusters varying from 2 to 8. The obtained distribution is presented on Figure 5.2.

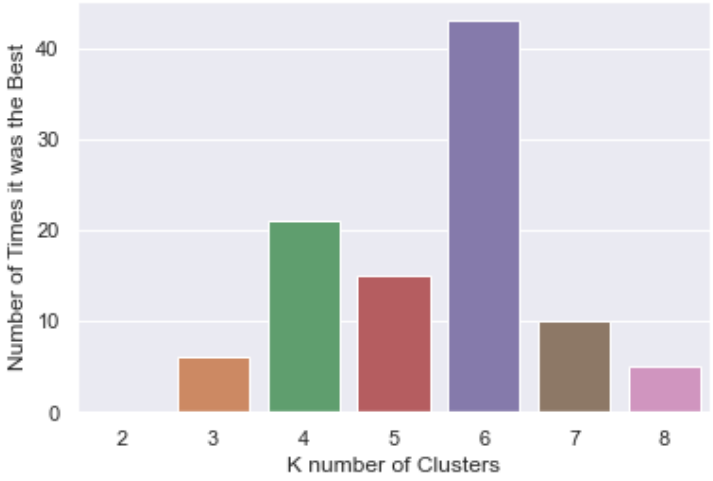


Figure 5.2: Distribution of the Number of times a K Number of Clusters achieved the largest Silhouette

As seen on Figure 5.2, the optimal number varies between several values depending on the run. The K value with the best silhouette in more runs is 6. However, there is still a considerable number of runs, with 4 and 5 as the best number of clusters. Moreover, six subgroups for the reduced number of patients might be too much and not be clinically justifiable.

The distributions of silhouettes, presented on Figure 5.3, showcase that for K=4, for example, the silhouette scores have a good median of 0.80. However, on these 100 runs, there were still some achieving lower values, closer to 0.60. Once again is also difficult to decide between 4, 5, or 6 since they present similar distributions.

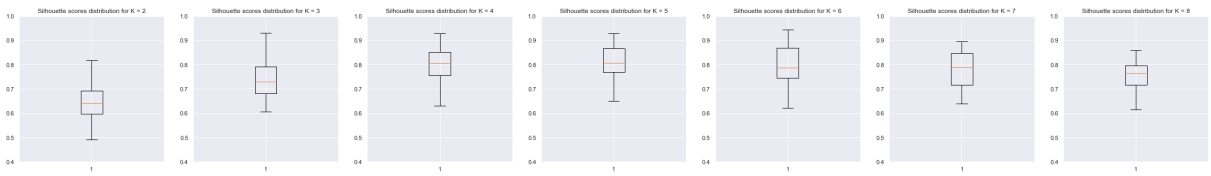


Figure 5.3: Distribution of Silhouette Scores for each K Number of Clusters

As already mentioned, the lack of stability in these results, causing difficulties in choosing a fixed K optimal number of clusters, motivated the pursuit of a Consensus Clustering approach. This approach is not dependent on a single run, increasing the robustness of the solution and providing a more straight-

forward way of choosing the best number of clusters.

5.2 Consensus Clustering Methods Results

In this section, the results of each configuration (Complete Feature Set, Using Only Temporal Features, and Using Only the First Record) will be presented separately.

For each one, it will be exposed the motivation behind the K number of clusters chosen, and the characterization of the clusters obtained.

5.2.1 Complete Feature Set

In this first consensus clustering configuration, the complete feature set obtained after the preprocessing (described in previous chapters) were used, both static and temporal.

5.2.1.A Determining the Optimal Number of Clusters

To access the optimal number of clusters for the Consensus Clustering the experiments of DTEC were conducted considering a range of values for optimal K number of clusters (i.e. [2,3,4]).

Using the distance matrix calculated as already explained on Chapter 4, the dendrogram on Figure 5.4 is obtained.

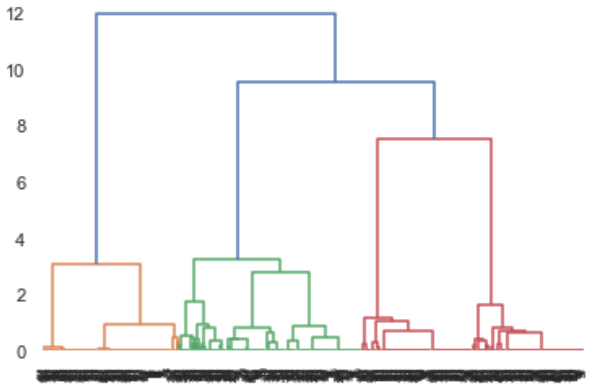


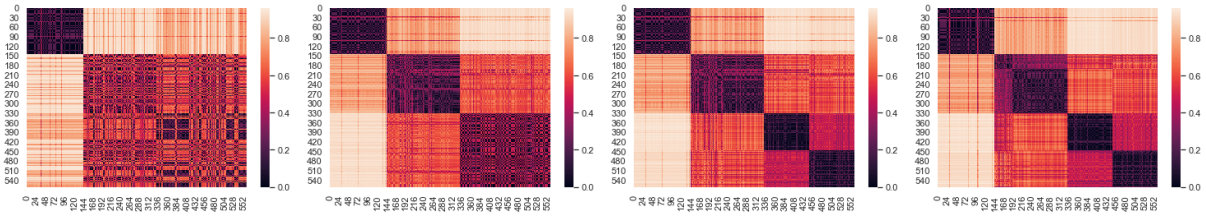
Figure 5.4: Dendrogram for the Consensus Clustering method using the complete feature set

Analyzing the dendrogram, the best number of clusters appears to be 4. However, since the visual analysis of the dendrogram can be deceiving, the decision is made considering two additional visualizations: the silhouette score plot (Figure 5.5) and heatmap visualizations of the distance matrix for each K number of clusters ordered (Figure 5.6).

The silhouette evolution appears to reach a maximum when the K number of clusters is 4. The value obtained is 0.75.

Figure 5.5: Silhouette Evolution for the Consensus Clustering method using the complete feature set

The heatmap visualization of the distance matrix, for each K number of clusters, is obtained by ordering the matrix across both axes according to the clustering labels. The heatmap is then plotted where the lowest values in the matrix cell (points with a smaller distance) are darker, and the highest values (points with a higher distance) are lighter. The heatmaps were plotted for K number of clusters varying from 2 to 9. On Figure 5.6 some of these visualizations are represented (the rest of them can be found on Appendix A). It is observed that for K=5, the clusters obtained by splitting one bigger cluster seem to be poorly defined between them. The best number of clusters appears to be 4.



(a) K=2 **(b)** K=3 **(c)** K=4 **(d)** K=5

Figure 5.6: Distance Matrix ordered Heatmap visualizations for each K number of clusters for the Consensus Clustering method using the complete feature set

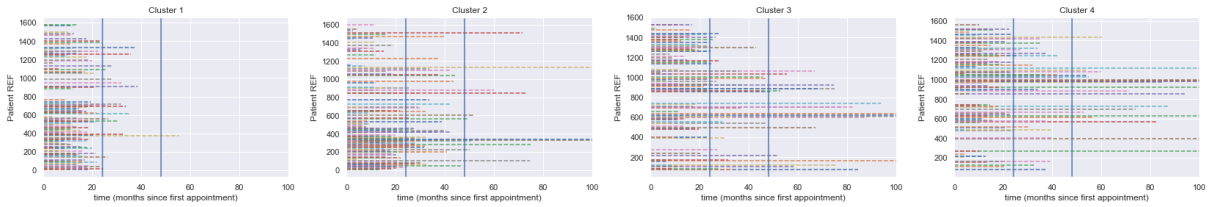
With this in mind, since all the analysis point in that direction, the optimal number of clusters is considered to be 4 for this configuration.

5.2.1.B Clusters Characterization

By considering the first 100 months of duration of patients' records at each cluster (Figure 5.7), some differences across clusters are observable. The two vertical lines correspond to 24 and 48 months, respectively, which translates to 2 and 4 years, the average clinical survival span of an ALS patient.

Patients in cluster 1 (Figure 5.7(a)) appear to have a shorter span in its records, with most patients'

duration not exceeding more than 24 months (2 years). This fact might point to a faster progression. In contrast, patients on cluster 3 (Figure 5.7(c)) and 4 (Figure 5.7(d)) appear to last longer (with some even surpassing the average clinical survival span) possibly hinting towards a slower progression. Finally, cluster 2 seems to have many patients inside the average clinical survival span limits (Figure 5.7(b)), possibly hinting at an average progression.



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure 5.7: Duration of each Cluster Record

By analyzing the distribution of the duration for each cluster (Figure 5.8), similar differences are noticed.

(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure 5.8: Duration Distribution of each Cluster

Considering the original temporal features' evolution, some particularities between clusters are also distinguished. In particular, the features in which these divergences are more substantial and, therefore, more relevant and easily observed are the ALS-FRS-R and some of its corresponding subscores.

For the following visualizations, the plots of each cluster were stopped when less than 30% of the patients remained active (i.e., have records). The timesteps chosen were spaced by 3 months each (i.e., 0, 3, 6, 9, 12, etc.). The values for each patient at each timestep (inside their duration span) were obtained with a simple interpolation. The medians for each timestep were then calculated considering only active patients at the given timestep.

By plotting the median evolution of the ALS-FRS-R metric for each cluster (see Figure 5.9), it is observed that patients of clusters 1 and 2 have lower values and a faster fall across time. On the other hand, patients in clusters 3 and 4 have higher values and more stable evolution.

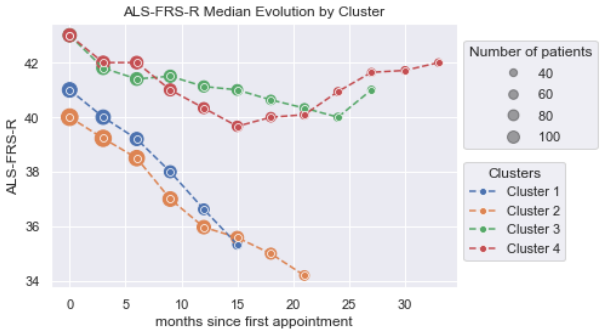
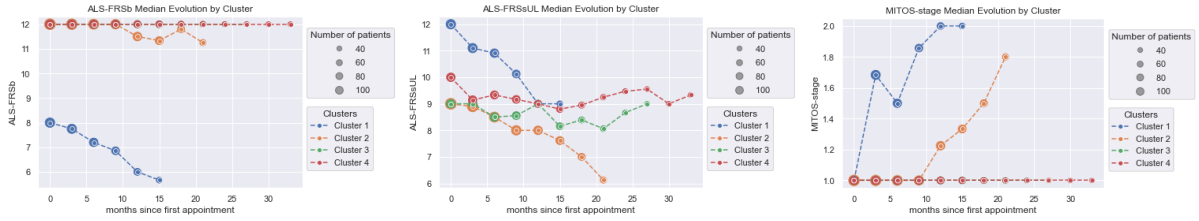


Figure 5.9: ALS-FRS-R Median Evolution across Clusters

Differences also occur on some of the subscores of ALS-FRS-R, more specifically on ALS-FRSb and ALS-FRSsUL, and on the MiToS score (Figure 5.10).



(a) ALS-FRSb **(b)** ALS-FRSsUL **(c)** MiToS

Figure 5.10: ALS-FRSb, ALS-FRSsUL and MiTos Median Evolution across Clusters

Analyzing the evolution of the ALS-FRSb (Figure 5.10(a)), cluster 1 stands out from the rest of them. The initial median is smaller and decreases fast, contrasting to the other clusters that stay mostly static at 12. In addition, cluster 2 presents a relatively slow reduction after 9 months.

Analyzing the ALS-FRSsUL median evolution (Figure 5.10(b)), a few differences across clusters are noticed as well. For instance, cluster 1 has initial higher values but drops quickly until around 9 points in the score. Cluster 2 starts at about 9 but drops as well, reaching a final value close to 6. Clusters 3 and 4 have a similar evolution, varying up and down between values 8 and 10. Cluster 3 has relatively lower values than cluster 4, ranging between 8 and 9.

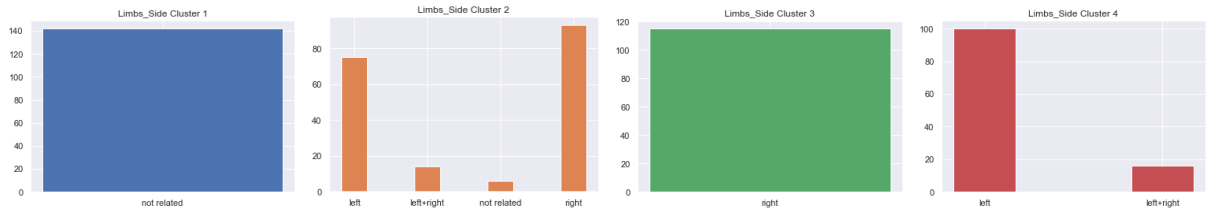
The MiToS stage (Figure 5.10(c)) also showcases a few differences. For instance, clusters 1 and 2 increase across time, with cluster 1 increasing faster and reaching value 2 (corresponding to loss of independence in two functional domains). Cluster 2 takes more months to start increasing but then does so fast until around 1.8. Clusters 3 and 4 medians stay constant at value 1, corresponding to a loss of independence in one functional domain.

It is clear by the analysis of clusters' duration and temporal features evolution that cluster 1 patients seem to correspond to a faster progression, and clusters 3 and 4 patients appear to hint at a slower

progression. On the other hand, cluster 2 patients' progression stays between these, possibly corresponding to an intermediate progression. After the analysis, however, it is still unclear how patients from cluster 3 and 4 differ.

An analysis of the static features was also performed to find what distinguishes the two groups of patients. Only the most relevant features distributions are presented, however additional results can be found on Appendix A.

By plotting the distribution of the static features for each cluster, it is evident that the main difference between clusters 3 and 4 lies in the Limbs Side feature (Figure 5.11). Cluster 3 contains patients only with right limbs side while cluster 4 contains mostly patients with left limbs side but also some with both (left+right).



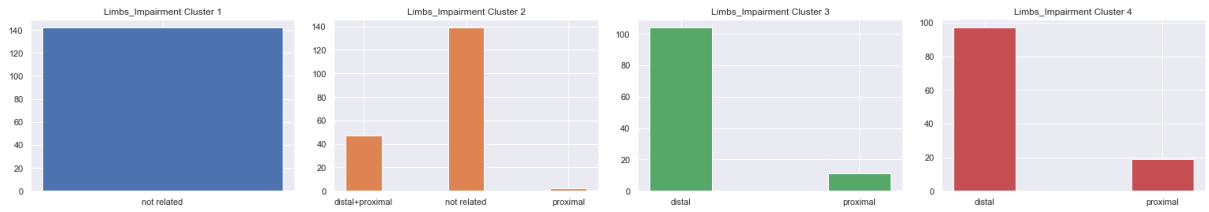
(a) Cluster1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure 5.11: Limbs Side Distribution by Cluster

Another insight extracted by analyzing the static features is that cluster 1, on its hand, appears to have no relation with limbs impairment, visible not only on the Limbs Side feature but also in other limb-related features such as Limbs Impairment, and on the Onset feature (Figures 5.12, 5.13 and 5.15).

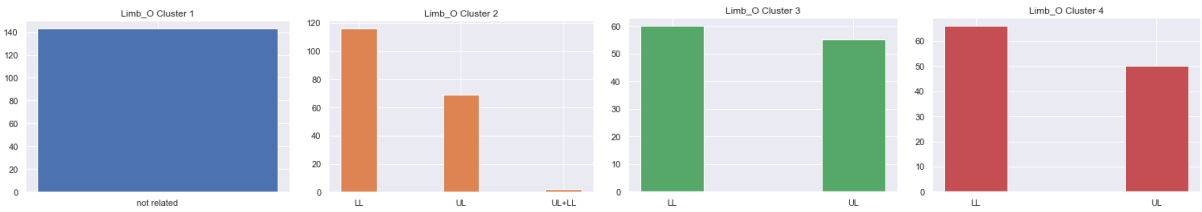
Furthermore, clusters 3 and 4 appear to be related to distal limbs impairment while cluster 2 appears to have a more diverse distribution for this feature (Figure 5.12).

Still regarding static features, more specifically the Gender (Figure 5.14), cluster 1 contains a slight majority of female patients, contrasting to the rest of the clusters where it is observed a slight male predominance.



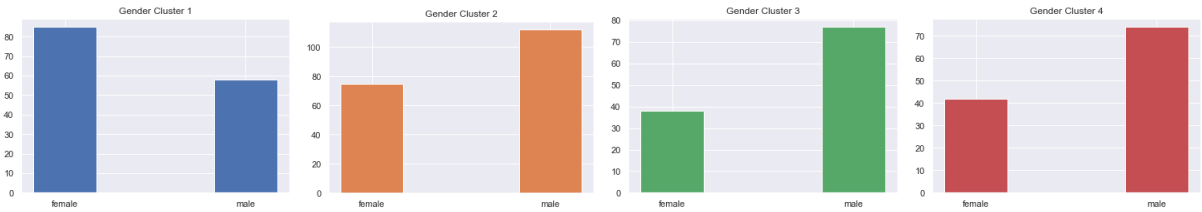
(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure 5.12: Limbs Impairment Distribution by Cluster



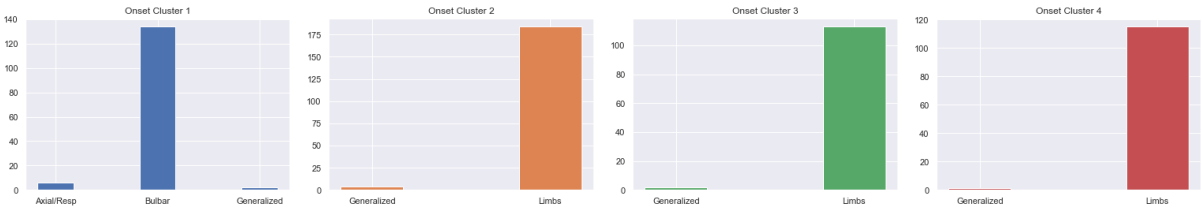
(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure 5.13: Limbs Onset Distribution by Cluster



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure 5.14: Gender Distribution by Cluster



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure 5.15: Onset Distribution by Cluster

Some other differences in terms of non-categorical static features are also found:

- The median age of patients on cluster 1 is higher (66), followed by clusters 2 and 3 (60) and with cluster 4 presenting the lowest value (59).
- The median diagnostic delay is the same for clusters 3 and 4 (11.99) and very close to the one of cluster 2 (12.02), cluster 1 is the only one with a more diverging value (9.03).

By looking to the whole analysis we can hypothesize that:

- Cluster 1 seems to correspond to patients with, mostly, bulbar onset but also includes patients with an Axial/Respiratory onset. The bulbar onset is medically associated with a slightly faster progression, and a shorter duration [46], compared with limbs onset. This insight is also observed

in the results found. Furthermore, the patients of this Cluster present an older age at onset and, in contrast with the other clusters, have a slight female predominance.

- Cluster 2 appears to correspond to patients with an average/fast progression and duration associated with limbs onset. A considerable number of patients shows both distal and proximal limbs impairment. There are also many ”not related” values at limbs impairment that are most certainly a result of the MVI, holding no clinical meaning due to the fact that the onset is limbs-related. In fact, although not possible to confirm, it is possible that these patients also correspond to distal+proximal limbs impairment due to the generally faster progression.

- Patients in clusters 3 and 4 present limbs onset and are associated with a slower progression and higher duration. Contrasting with cluster 2, patients in these clusters only present one type of impairment (only distal or proximal, but predominantly distal). They also show only one type of limbs affected (either upper or lower).

- The main difference between clusters 3 and 4 is the side of limbs affected. While cluster 3 contains only patients with right limbs affected, Cluster 4 includes patients with either left or, some times, both limbs affected.

5.2.2 Using Only Temporal Features

For this configuration, only features with temporal evolution were used. The distinction between temporal and static features is present on Chapter 3.

5.2.2.A Determining the Optimal Number of Clusters

A similar analysis is performed using the same methodology described previously in Section 5.2.1.A, considering the dendrogram, silhouette evolution, and distance matrices heatmaps for the Consensus Clustering method using only temporal features.

The dendrogram obtained (Figure 5.16) hints at an optimal value of K number of clusters of 2. However, only two clusters are not considered to be enough in our context from a medical point of view. Therefore, excluding 2 as a possible number of clusters, the best value that the dendrogram appears to point to is 3.

By analyzing the silhouette evolution plot(Figure 5.17), the same condition is observed with the choice of 3 or 5 as the optimal number of clusters appearing to be the best choice if excluding 2. The values of silhouette achieved for 3 and 5 are, respectively, 0.67 and 0.64. Therefore 3 appears to be the best choice.

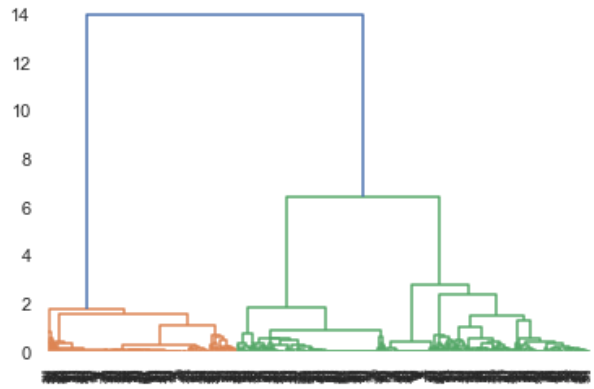
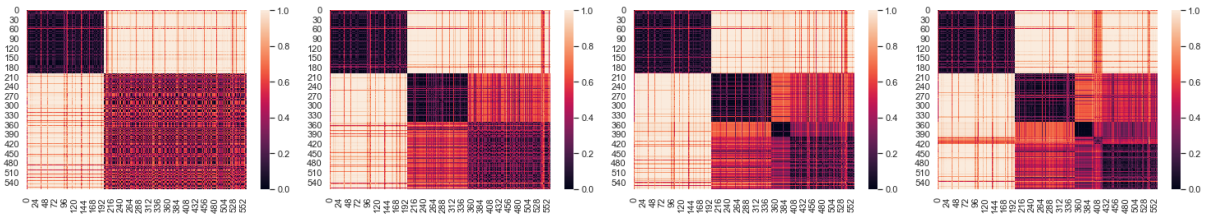


Figure 5.16: Dendrogram for the Consensus Clustering method using only temporal features

Figure 5.17: Silhouette Evolution for the Consensus Clustering method using only temporal features.

Furthermore, when observing the heatmap visualizations of the distance matrix for each K number of clusters ordered (Figure 5.18), 3 also appears to obtain reasonably defined groups. With 5 appearing to get smaller clusters that lack definition, appearing to be more blended between them.



(a) K=2 **(b)** K=3 **(c)** K=4 **(d)** K=5

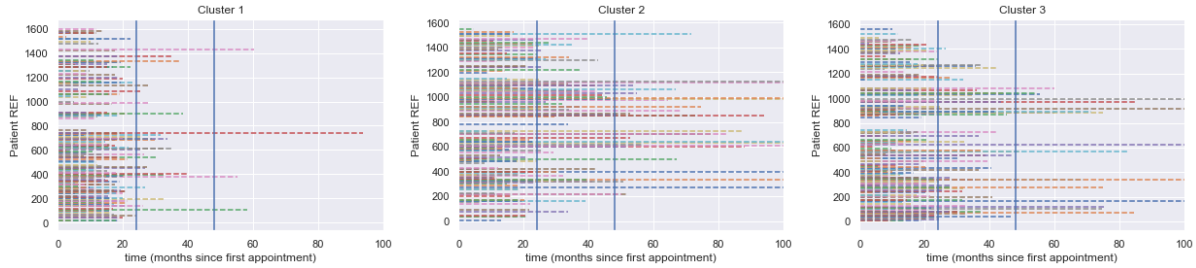
Figure 5.18: Distance Matrix ordered Heatmap visualizations for each K number of clusters for the Consensus Clustering method using only temporal features

The optimal number of clusters is chosen to be 3, achieving a silhouette of 0.67, with, once again, all visualizations supporting this decision (after excluding 2 as a viable option).

5.2.2.B Clusters Characterization

The following analysis is done using the same techniques as the previous configuration. Once again, by considering the first 100 months of duration of patients' records at each cluster (Figure 5.7), some insights are found.

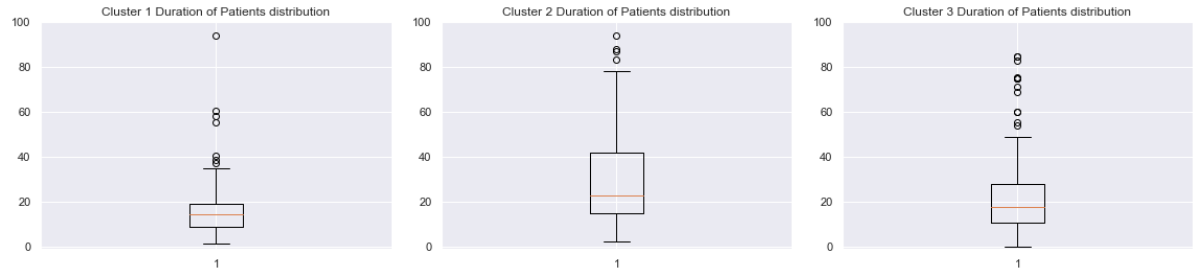
Patients in cluster 1 (Figure 5.19(a)) seem to have a shorter duration, most of which do not exceed the 2 years mark, pointing to a faster progression. Patients on cluster 2, on the other hand, (Figure 5.19(b)) appear to be the ones lasting longer, with several surpassing the 4 years mark, possibly hinting towards a slower progression. Finally, cluster 3 (Figure 5.19(c)) patients present an intermediate duration compared to the other two, possibly hinting towards an intermediate progression too (Figure 5.19(c)).



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3

Figure 5.19: Duration of Each Cluster

These differences are also supported by observing the distribution of the duration of each cluster (Figure 5.20).



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3

Figure 5.20: Duration Distribution of each Cluster Record

Regarding the evolution of the temporal features, once again, the divergences more easily identified are the ones on ALS-FRS-R and corresponding subscores. By plotting the median evolution of the ALS-FRS-R metric for each cluster (see Figure 5.21), it is observed that patients of clusters 1 and 3 have lower values and a more drastic fall across time. On the other hand, cluster 2 patients have higher

values and more stable evolution.

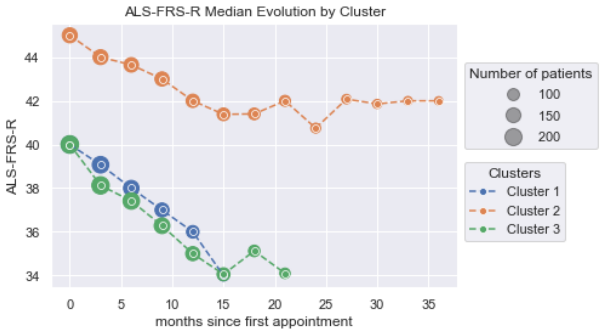
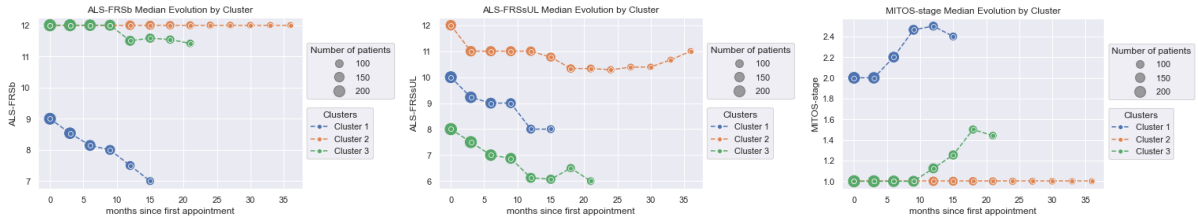


Figure 5.21: ALS-FRS-R Median Evolution across Clusters

Differences occur particularly in ALS-FRSb, ALS-FRSsUL, and in the MiToS scores (Figure 5.22). Analyzing the ALS-FRSb evolution (Figure 5.22(a)), cluster 1 has a smaller initial median, and it decreases fast; cluster 2 remains static at the top (12); cluster 3 starts at the top and slowly decreases after some time.



(a) ALS-FRSb **(b)** ALS-FRSsUL **(c)** MiToS

Figure 5.22: ALS-FRSb, ALS-FRSsUL and MiTos Median Evolution across Clusters

Regarding ALS-FRSsUL median evolution (Figure 5.22(b)), cluster 1 and 3 decrease in a similar degree, however, cluster 3 presents lower values (starts at 8 and decreases until 6) than cluster 1 (starts at 10 and decreases to 8). Cluster 2 presents the highest values, starting at 12, decreasing to 11 and then maintaining relatively stable values across time.

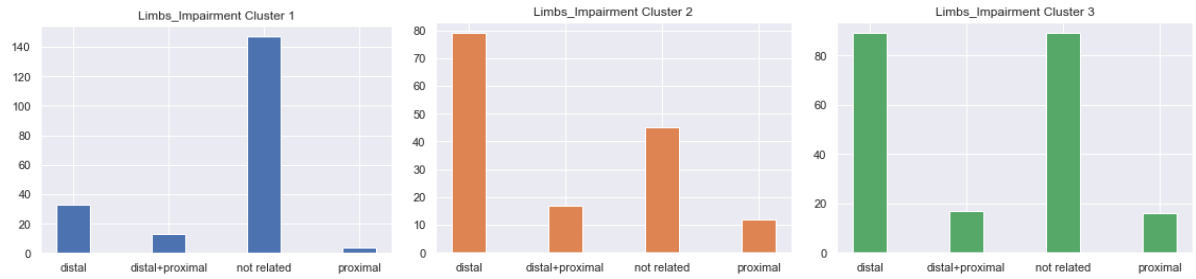
For MiToS stage evolution (Figure 5.22(c)), clusters 1 and 3 increase across time, and cluster 2 remaining static at value 1. Cluster 1 starts at 2, increasing first and reaching a value of 2.5. Cluster 3 starts lower at 1, and it takes more months to start to rise, but then does so fast until 1.5.

Although this configuration only uses temporal features, some insights can also be retrieved when analyzing the distribution of the static features. In particular, by observing the differences on some of the presentation of the disease at onset.

Cluster 1 appears to contain most patients with bulbar onset. This type of patient is a tiny minority in the other two clusters (Figure 5.26). This fact also leads to the majority of patients in that cluster not

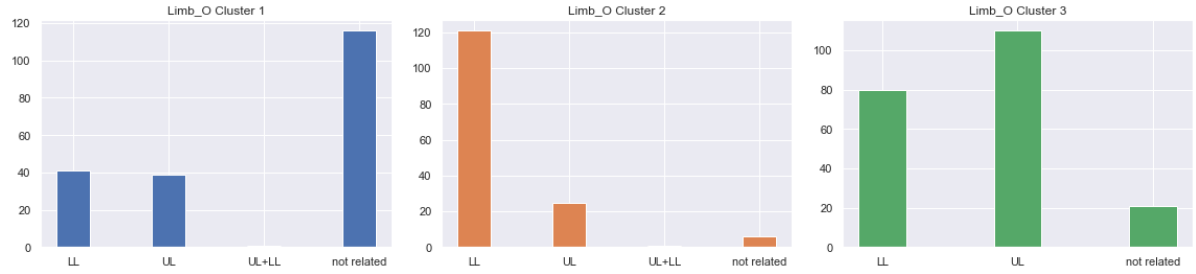
being related to limbs impairment (Figure 5.23).

Furthermore, cluster 3 appears to include the majority of patients with upper limbs onset, while the predominant limbs onset in cluster 2 is lower limbs (Figure 5.24).



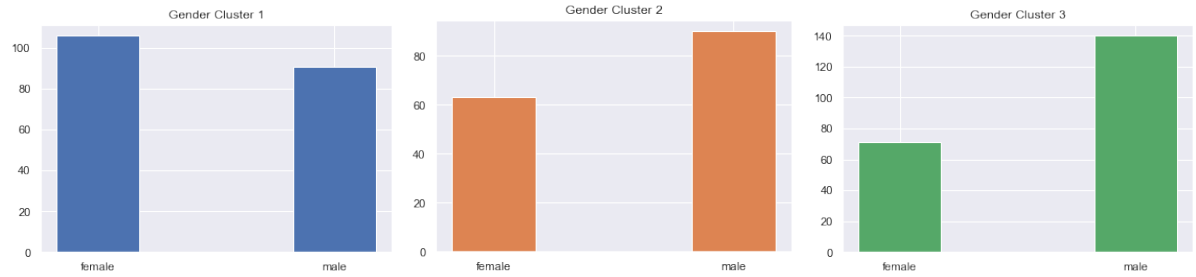
(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3

Figure 5.23: Limbs Impairment Distribution by Cluster



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3

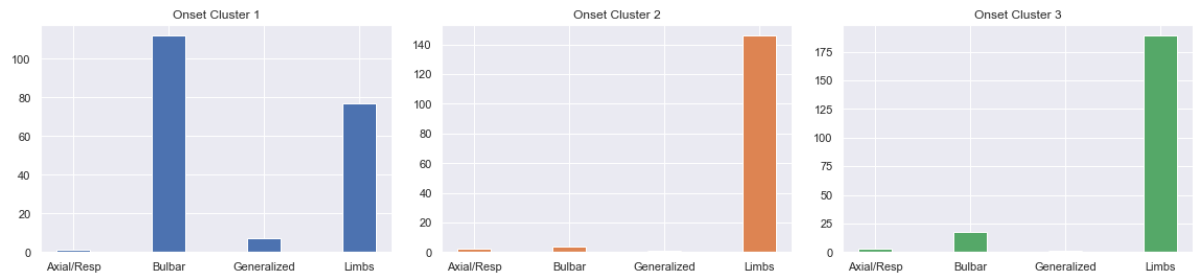
Figure 5.24: Limbs Onset Distribution by Cluster



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3

Figure 5.25: Gender Distribution by Cluster

Once again a slight female predominance is observed in cluster 1 contrasting to the other clusters where there is a slight male predominance (Figure 5.25).



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3

Figure 5.26: Onset Distribution by Cluster

Additional differences, specifically on the numerical static features include:

- The median age of patients, which is higher in cluster 1 (64), followed by clusters 3 (61), and finally, cluster 2 (59).
- The median diagnostic delay is also slightly different, 9.56 for cluster 1, 11.99 for cluster 3, and 12.19 for cluster 2.

Resuming the results obtained:

- Patients on cluster 1 appear to be relatively older at onset, presenting a slight majority of female patients (in contrast with the other clusters). Patients appear to be associated with a faster progression, bulbar onset, and primarily unrelated to limbs impairment.
- Cluster 2 patients can be associated with a slower progression and limbs onset, especially regarding lower limbs and a distal impairment.
- Cluster 3 seems to contain mostly younger patients with limbs onset, especially associated with upper limbs. The limbs impairment distribution is similar to the one of cluster 2. The progression in this last cluster also appears to be fast, but not as fast as cluster 1.

5.2.3 Using Only the First Record

An adjustment is made in the DTEC method to use only the first record of each patient. Specifically, the TAE module is replaced by a Dense Autoencoder (DAE). The key differences in this autoencoder are the use of a Flatten layer at the beginning and the replacement of LSTM layers by Dense layers. All encoding dimensions and activation functions are the same, as well as the optimizer and loss functions used. All the features are used for this configuration, both static and temporal(note, however, that only the first record of temporal features is used as well).

5.2.3.A Determining the Optimal Number of Clusters

Once again, using the same methodology, an analysis is performed considering the dendrogram, silhouette evolution, and distance matrices heatmaps for the Consensus Clustering method using only the first record of each patient.

The dendrogram obtained (Figure 5.27) hints at an optimal value of K number of clusters of 2. However, only two clusters are not considered to be enough in our context from a medical point of view. Excluding 2 as a possible number of clusters, the best value that the dendrogram appears to point to is 3.

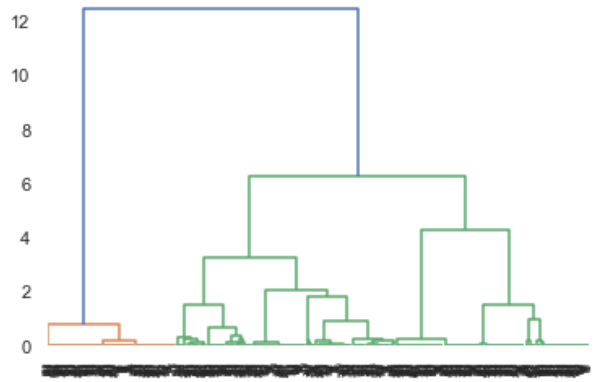
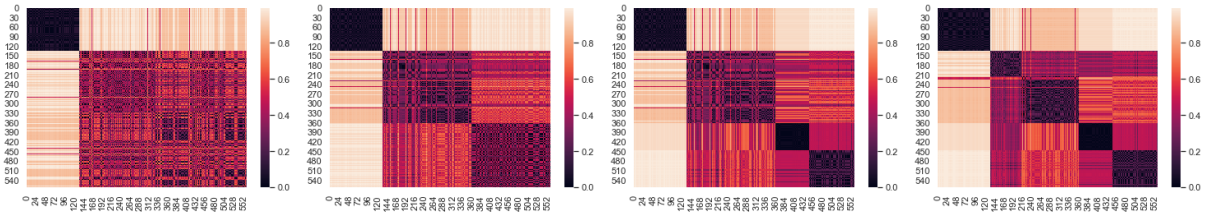


Figure 5.27: Dendrogram for the Consensus Clustering method using only the First Record

However, if we analyze the silhouette evolution plot(Figure 5.28), the choice of 3 as the optimal number of clusters appears to be a wrong choice since its the lowest value achieved (0.58). Higher K's appear to obtain better results. However, a too-high number of clusters for the small number of patients we are working on is also not considered medically relevant. The best-compromised value would then be 4, which achieves a silhouette score of 0.68.

Figure 5.28: Silhouette Evolution for the Consensus Clustering method using only the first record

Furthermore, the same situation can be observed when analyzing the heatmap visualizations of the distance matrix for each K number of clusters ordered (Figure 5.29). In this case, 4 seems to be the most reasonable choice.



(a) K=2 **(b)** K=3 **(c)** K=4 **(d)** K=5

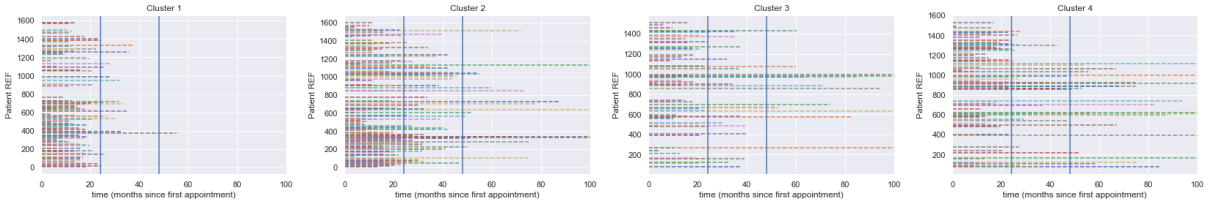
Figure 5.29: Distance Matrix ordered Heatmap visualizations for each K number of clusters for the Consensus Clustering method using only the first record

The optimal number of clusters is then decided to be 4 for this configuration, with a silhouette of 0.68, since it is the best-compromised value found when analyzing all the visualizations.

5.2.3.B Clusters Characterization

One more time, considering the first 100 months of duration of patients' records at each cluster (Figure 5.30) differences are observed.

Similarly to the first configuration ("Complete Feature Set") there is a cluster with shorter duration (cluster 1 - Figure 5.30(a)), one with an average duration (cluster 2 - Figure 5.30(b)) and two with an apparent higher duration (cluster 3 and 4 - Figures 5.30(c) and 5.30(d)). Analyzing the duration distributions for each cluster, values appear to increase from cluster 1 to cluster 4 (Figure 5.31).



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure 5.30: Duration of each Cluster Record

The evolution of the ALS-FRS-R score for each cluster (see Figure 5.32) is also similar to the first configuration. Patients on clusters 1 and 2 present smaller scores of ALS-FRS-R and showcase a faster drop. Patients on clusters 3 and 4 have higher values with a more stable evolution across time.

(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure 5.31: Duration of each cluster record

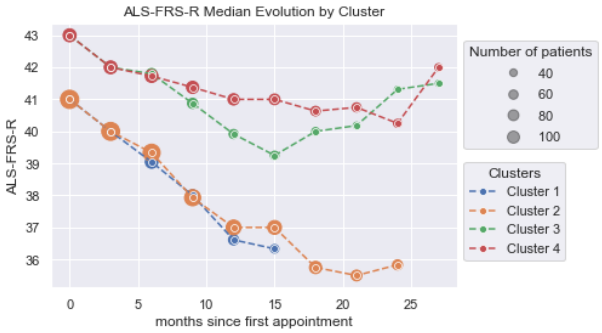
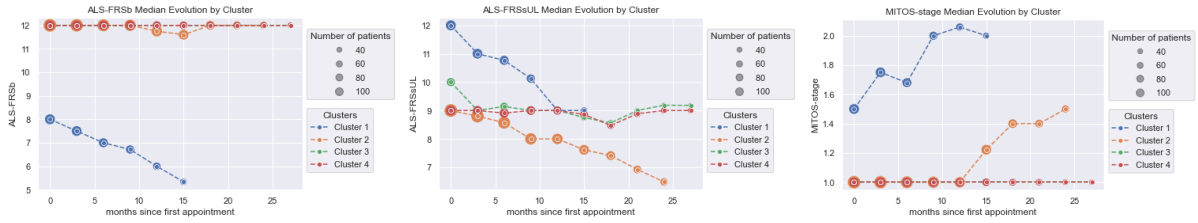


Figure 5.32: ALS-FRS-R Median Evolution across Clusters

Regarding other relevant temporal features such as ALS-FRSb, ALS-FRSsUL, and the MiToS score, differences are again present (Figure 5.33).



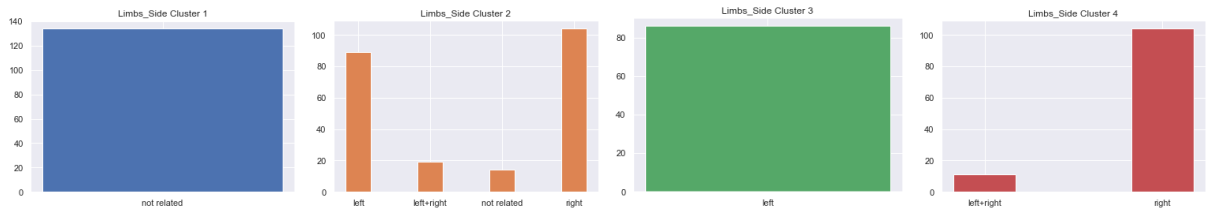
(a) ALS-FRSb **(b)** ALS-FRSsUL **(c)** MiToS

Figure 5.33: ALS-FRSb, ALS-FRSsUL and MiTos Median Evolution across Clusters

For ALS-FRSb (Figure 5.33(a)), cluster 1 has lower values and decreases across time contrasts with the static evolution of the other 3 clusters. On ALS-FRSsUL(Figure 5.33(b)) both cluster 3 and 4 maintain stable values around 9 points of the score while cluster 1 and 2 decrease across time. Cluster 1 diverges from cluster 2 by presenting higher values: cluster 1 starts at 12 and decreases until 9; cluster 2 starts at 9 and decreases until close to 6. The MiToS score remains stable for clusters 3 and 4 and increases across time for clusters 1 and 2, with cluster 1 starting first and reaching higher values.

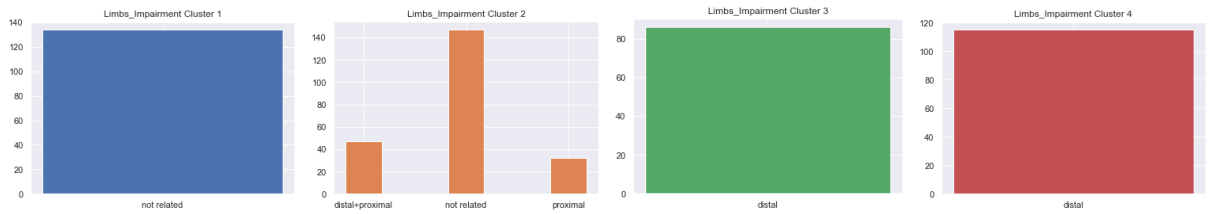
Similarly to the first configuration of features, the analysis of temporal features is not enough to

differentiate cluster 3 from cluster 4. Analyzing the remaining static features, however, it is clear that the key diverging aspect is once again the side of the affected limbs (Figure 5.34).



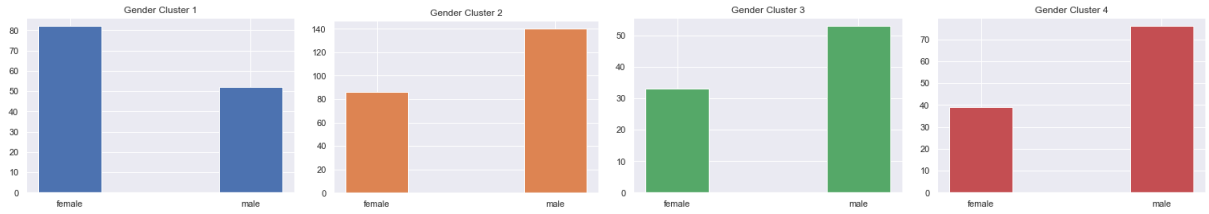
(a) Cluster1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure 5.34: Limbs Side Distribution by Cluster



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure 5.35: Limbs Impairment Distribution by Cluster



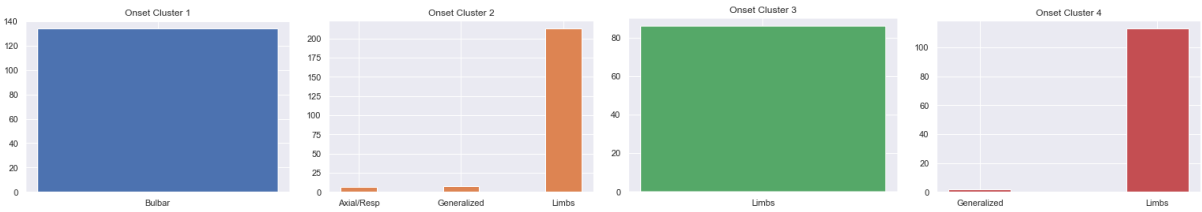
(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure 5.36: Gender Distribution by Cluster

For the results of this configuration, cluster 3 contains patients with only left side onset, and cluster 4 patients with right or both sides onset. Moreover both cluster 3 and 4 present exclusively distal impairment(Figure 5.35).

Cluster 1 contains all the bulbar onset patients solely. While cluster 2 contains mostly limbs onset patients with mostly no limbs impairment at the onset (Figures 5.35 and 5.37).

Similarly to the other configurations, a slight female predominance was observed in cluster 1, contrasting to the male predominance on the rest of the clusters (Figure 5.25).



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure 5.37: Onset Distribution by Cluster

Regarding non-categorical static features:

- The median age of patients is 60 for all clusters except for cluster 1, where it is 66.5
- The diagnostic delay is lower in cluster 1 (9.12), increasing slightly across cluster 2 (11.86), cluster 3 (11.99) and cluster 4 (12.81).

Looking at the whole analysis we can extract a few insights:

- Cluster 1 corresponds to bulbar onset patients, generally older, with a slight female predominance, and presenting a faster progression which leads to a shorter duration.
- Cluster 3 and 4 present a similar slower progression and longer duration, diverging in the side of the limbs affected. Cluster 3 contains patients with left limbs onset, while cluster 4 presents patients with right or both limbs affected.
- Cluster 2 lies between the other clusters, presenting a relatively fast progression but with a longer duration when compared with cluster 1. It contains limbs onset patients with distinct sides affected. Regarding limbs, impairment, it contains all the distal+proximal patients but also many ”not related” patients. As already explained in other configurations, the ”not related” impairment on limbs onset patients is a result of MVI and holds no meaning. Once again, since the evolution analysis showed a faster progression and the cluster also contains distal+proximal patients, it is more likely that the patients marked as ”not related” are also distal+proximal patients.

5.3 Discussion and Conclusions

Initially, the DTEC method raised a few concerns regarding the difficulty of choosing the optimal number of clusters due to the variations in the data representations obtained. These variations were likely a result of the use of non-stochastic steps at DTEC combined with a high heterogeneity and small size of the dataset.

This issue was overcome with the Consensus Clustering approach, which increased the robustness of our solution by providing a clear way of choosing the best number of clusters. This method is not dependent on a single run, using the results obtained across several runs and with different numbers of clusters to calculate the distance between points and, with that, choose the optimal number of clusters that maximizes the silhouette.

Between the three Consensus Clustering configurations experimented, the clusters that obtained a better silhouette score were the ones of the ”Complete Feature Set” configuration (Section 5.3), reaching a silhouette of 0.75. This configuration was also the one where it was more evident which was the best number of clusters (4 in this case) by observing the visualizations implemented.

Configuration	Number of Clusters	Silhouette score
Complete Feature Set	4	0.75
Only Temporal Features	3	0.67
Only the First Record	4	0.68

Table 5.1: Silhouette comparison across configurations

Regarding cluster characterization, a few similarities across configurations can be found. Generally, bulbar onset patients are grouped in one cluster, always associated with an apparent faster progression/shorter duration, which is also supported by clinical studies [46]. These patients also appear to be older at onset and present a slight female predominance in contrast with patients on the rest of the clusters. One or two clusters appear to group patients with a slower progression in all configurations, associated with limbs onset, sometimes divided by the side of the limbs affected. Finally, the remaining patients appear to be grouped in a cluster with a more intermediate progression.

Furthermore, the clusters obtained using only the first record appears to present similarities with the ones of the first configuration (”Complete Feature Set”), as observed on the Sankey diagram of Figure 5.38. This similarity could point to the fact that more than three records might be required to find more temporal patterns on the data. However, as seen previously, the size of the data when considering patients with more than 3 records is very small (only around 400 patients, a drop of around 25-30% of the current 561 patients, which is already a small number). Furthermore, the clusters obtained using only three records do present differences in the evolution of the disease as observed on the results obtained. An alternative hypothesis is simply that the onset observations and the disease evolution have a strong correlation. Important to note that although the clusters obtained presented similarities, the ”Complete Feature Set” configuration obtained a silhouette score 5% higher than the ”First Record Only” configuration.

Interestingly, only three clusters were defined when removing the static features (”Temporal Only” configuration). This fact might reflect the importance of static features in separating some of the clusters

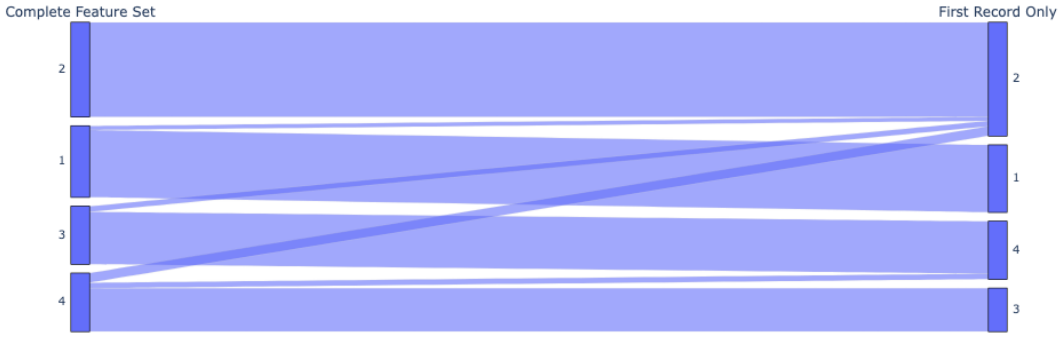


Figure 5.38: Sankey diagram comparing the clusters obtained by the ”Complete Feature Set” and the ”First Record Only” configurations

in the ”Complete Feature Set” configuration. In fact, for this configuration, the most significant difference in clusters 3 and 4 is indeed a static feature: the side of the limbs affected.

A medical study by Zhang et al. [47] suggests that the side of the limbs affected appears to be related to the area of the brain involved and to the severity of the disease. This medical study [47] also notes that patients with right limb onset have an older onset age and lower disability severity (measured by ALS-FRS-R) compared with left limb onset patients. In our data, these differences are not so easily noted. The age gap between cluster 3 (right-side patients) and 4 (primarily left-side patients) of the ”Complete Feature Set” configuration is relatively trim (60 vs. 59), and the ALS-FRS-R evolution is similar. Nonetheless, the correlation between the limbs side and the disease presentation/brain regions affected [47] justifies the clinical relevance of the clusters obtained.

With this in mind, the configuration considered to obtain the most relevant results was the ”Complete Feature Set” configuration. The clusters obtained with this configuration will be used for testing a personalized prognostic prediction approach (see Chapter 6), which can also be viewed as a validation mechanism.

5.4 Summary

This section presents the main results of the implemented patient stratification methodology.

Different configurations are detailed, and a characterization of the clusters obtained is presented for each. The configurations chosen correspond to experiments within a DTEC configuration without Consensus Clustering and three Consensus Clustering variants. The Consensus Clustering variants are ”Complete Feature Set”, the first one using both static and temporal features; ”Temporal Only”, using

only the temporal features; and ”First-Record Only”, using the complete feature set but just considering the first record of each patient.

Clusters are evaluated considering their silhouette but also the differences they present in terms of records duration, temporal features evolution and static features distribution.

Furthermore, the results across configurations are compared with each other to decide which one produces the best and most clinically relevant subgroups. Some similarities across configurations were found, with the best performing configuration considered to be ”Complete Feature Set”, which obtained 4 clusters with a silhouette of 0.75.

The chosen clusters will be used in the next chapter Chapter 6 to implement a personalized prognostic prediction approach.

6

Prognostic Prediction

Contents

This chapter describes the prognostic prediction approach developed. It exposes the general methodology of the classification pipeline implemented, using the stratified groups of patients obtained on Chapter 5, and how this pipeline was trained and evaluated.

It further presents the results obtained regarding the need of NIV prediction by cluster across the different classifiers experimented. These results are compared with a baseline without stratification and the main conclusions reached are exposed.

6.1 Methodology

Inspired by the promising results obtained from previous stratification works when applying prognosis prediction to each separate subpopulation (such as the one of Pires et al. [11]), a classification pipeline was developed. Using the medically relevant clusters obtained by the DTEC, the classification pipeline is applied to each to deliver a more personalized and targeted prediction of the patients' need of NIV.

The general methodology pipeline of the prediction task is present on Figure 6.1.

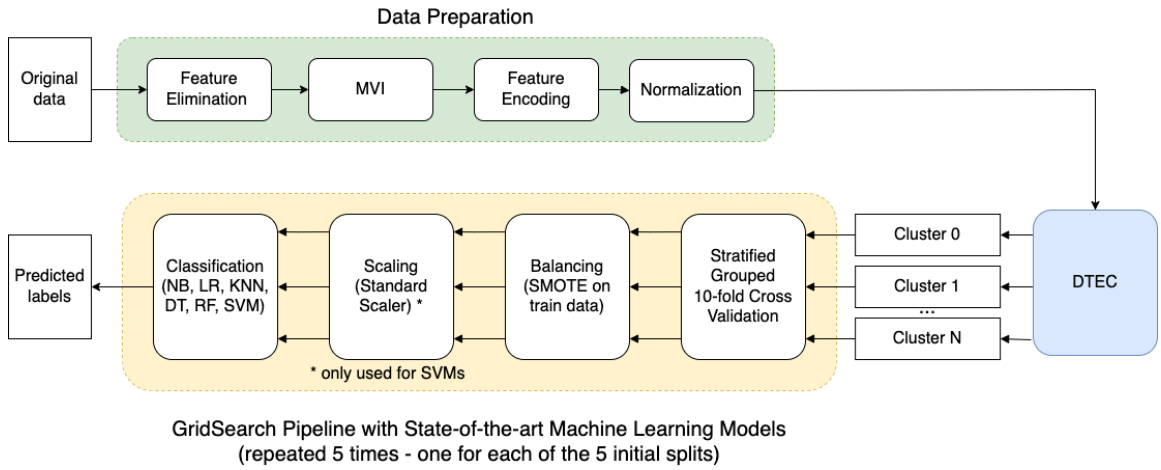


Figure 6.1: Classification task workflow

After the initial data preparation (described on Chapter 3), the data is grouped in clusters by the DTEC method(described on Chapter 4). The clusters obtained by the best configuration of DTEC (Chapter 5) are fed separately to the Prediction Pipeline. For each cluster of patients C, the data fed to the Classification Pipeline corresponds to the records of all patients in cluster C after the initial Data Preparation is applied.

The Classification Pipeline uses a Gridsearch and a cross-validation methodology to train state-of-the-art classifiers, obtaining a predicted label for each record. It includes a Balancing step (and, in the case of SVMs, Scaling also) to preprocess the data and improve the results. Several classifiers were used, namely, Naive Bayes (NB), Logistic Regression (LR), K-nearest neighbors (KNN), Decision

Trees (DT), Random Forests (RF), and Support Vector Machines (SVM). The training and evaluation methodology will be discussed in more detail next.

6.1.1 Determining the Time-Window to use

In order to decide the best time-window to use from the three available (90, 180 and 365), an evaluation regarding target variable distribution on the whole dataset, as well as on each separate cluster, was performed. The results are summarized on Table 6.1.

Table 6.1: Balancing of each Cluster by Time-window

Time-window	Cluster	Positive Records	Negative Records	# Records	# Patients
90	1	24 (3.64%)	635 (96.36%)	659	142
	2	46 (4.00%)	1103 (96.00%)	1149	188
	3	14 (1.70%)	810 (98.30%)	824	115
	4	17 (2.02%)	823 (97.98%)	840	116
	TOTAL	101 (2.91%)	3 371(97.09%)	3472	561
180	1	99 (15.02%)	560 (84.98%)	659	142
	2	149 (12.97%)	1000 (87.03%)	1149	188
	3	64 (7.77%)	760 (92.23%)	824	115
	4	66 (7.86%)	774 (92.14%)	840	116
	TOTAL	378 (10.89%)	3094 (89.11%)	3472	561
365	1	241 (36.57%)	418 (63.43%)	659	142
	2	354 (30.81%)	795 (69.19%)	1149	188
	3	149 (18.08%)	675 (81.92%)	824	115
	4	168 (20.00%)	672 (80.00%)	840	116
	TOTAL	912 (26.26%)	2560 (73.74%)	3472	561

We can observe that the 90 days time-window presents a very high imbalance of the data, both generally and by cluster, therefore being the first to be excluded. The 180 days time-window is more balanced but still highly unbalanced. The remaining 365 days time-window, although containing some clusters with a high imbalance of the target variable, is still the best among the three.

6.1.2 Training and Evaluation Methodology

The evaluation and training strategy of the classifiers is based on a 5×10 cross-validation methodology. First, we perform a stratified grouped shuffle split, using 5 different random seeds for each data table, with 70% of the data assigned to train and 30% to test. This technique ensures that the records of

a single patient are not divided across train and test (i.e., are all in train or all in test). In addition, it both shuffles the data and guarantees that the split is as stratified as possible (i.e., the class distribution remains as close as possible to the original data).

For each of these 5 splits, and in order to train the model, we apply a 10-fold stratified group cross-validation (again to ensure records of the same patient are not separated) and use a grid search for tuning the hyperparameters. The grid search follows a pipeline composed of two stages, balancing using SMOTE (which is highly relevant due to how unbalanced the data is, as seen previously) and the classifier.

SMOTE (synthetic minority oversampling technique) was chosen as the balancing strategy. The number of items in the minority class of some clusters is residual and so undersampling strategies would most likely not suffice or reduce the already small dataset too much. SMOTE has the advantage of generating new, slightly different synthetic records instead of simply replicating the existing ones. Doing so increases the number of records in the minority class, improving the balance of the dataset.

Additionally, an extra step is placed after SMOTE and before the classifier for SVMs, namely, Scaling using *scipy*'s $StandardScaler$, which helps speed up the algorithm.

We used six state-of-the-art classifiers, Naïve Bayes (NB), Logistic Regression (LR), K nearest neighbors (KNN), Decision Trees (DT), Random Forests (RF), and Support Vector Machines (SVM). The hyperparameters values experimented for each classifier are present in Table 6.2.

Table 6.2: Hyperparameters experimented for each Algorithm

Algorithm	Hyperparameters	Values experimented
NB	Priors	None
	Variance Smoothing	1×10^{-7}; 1×10^{-8}; 1×10^{-9}
LR	Penalty	L2
	C	0.01; 0.1; 1; 10; 100;
	Solver	newton-cg; lbfgs; liblinear
KNN	Metric	Manhattan; Euclidean; Chebyshev
	No. Neighbours	1; 2; 3; 5; 7; 9; 11; 13; 17; 19; 23; 29
DT	Criterion	Gini; Entropy
	Max. Depth	2; 5; 7; 13; 23; 29
	Min. Impurity Decrease	0.05; 0.025; 0.01; 0.005; 0.0025; 0.001
RF	No. Estimators	5; 15; 25; 50; 75; 100; 150; 200; 250; 300
	Max. Depth	2; 5; 7; 13; 23; 29
	Max. Features	0.1; 0.25; 0.5; 0.75; 0.9; 1
SVM	C	0.1; 1; 10; 100
	Gamma	1; 0.1; 0.01; 0.001
	Kernel	RBF; Polynomial

The evaluation metric used for the grid search optimization is the area under the ROC curve (AUC). We use Specificity (Spe.), Sensitivity (Sen.), and AUC to evaluate the classifiers. It is important to note that all the records are considered independent for evaluation purposes.

6.2 Results obtained

In this section, the results obtained with our classification pipeline are presented. The results of applying the pipeline to each subgroup obtained from DTEC are compared with the ones obtained with the complete dataset (Table 6.3). In this table, the algorithm considered to perform best for each section is shaded green. Additionally, the best values for each metric, if not the ones of the best algorithm, are shaded blue.

Table 6.3: Prediction results using Patient Stratification (DTEC) by Cluster and using No Stratification (365 days time-window). For each cluster it is indicated the number of patients and records (and the percentage of the total data they represent in terms of patients and records, respectively).

	NB	**LR**	**KNN**	**DT**	**RF**	**SVM**
cluster 1 // 142 patients (25.31%) // 659 records (18.98%)						
Spec.	80.0 +- 0.0	70.0 +- 2.49	63.33 +- 9.13	59.77 +- 7.24	80.35 +- 3.95	50.56 +- 14.49
Sen.	30.59 +- 2.63	69.23 +- 0.0	47.06 +- 0.0	50.68 +- 5.25	44.0 +- 7.6	65.0 +- 11.18
AUC	55.29 +- 1.32	69.62 +- 1.24	55.2 +- 4.56	55.23 +- 5.27	61.56 +- 2.27	57.78 +- 1.96
cluster 2 // 188 patients (33.51%) // 1149 records (33.09%)						
Spec.	52.32 +- 1.75	66.78 +- 0.38	57.52 +- 7.2	63.06 +- 9.66	67.99 +- 0.69	75.11 +- 7.48
Sen.	77.86 +- 1.47	62.28 +- 1.98	44.11 +- 2.06	58.4 +- 9.98	58.24 +- 2.82	51.24 +- 6.82
AUC	65.09 +- 0.14	64.53 +- 0.8	50.81 +- 4.48	60.73 +- 2.15	63.11 +- 1.36	63.18 +- 1.22
cluster 3 // 115 patients (20.50%) // 824 records (23.73%)						
Spec.	61.74 +- 1.94	73.33 +- 3.95	26.96 +- 11.67	74.07 +- 6.93	98.18 +- 1.66	50.00 +- 0.0
Sen.	80.0 +- 0.0	71.43 +- 0.0	92.0 +- 17.89	63.33 +- 7.45	28.57 +- 10.1	100.0 +- 0.0
AUC	70.87 +- 0.97	72.38 +- 1.98	59.48 +- 3.11	68.7 +- 3.23	63.38 +- 4.56	75.0 +- 0.0
cluster 4 // 116 patients (20.68%) // 840 records (24.19%)						
Spec.	72.71 +- 5.52	70.99 +- 2.69	33.18 +- 2.76	53.27 +- 6.13	84.66 +- 4.44	81.43 +- 1.6
Sen.	58.18 +- 13.28	45.09 +- 6.95	49.7 +- 7.61	58.54 +- 7.74	45.65 +- 10.95	85.71 +- 0.0
AUC	65.45 +- 3.88	58.04 +- 3.08	41.44 +- 2.43	55.9 +- 3.64	65.16 +- 3.96	83.57 +- 0.8
No Stratification results // 561 patients // 3472 records						
Spec.	62.64 +- 0.0	73.25 +- 0.22	63.33 +- 2.13	72.44 +- 2.76	64.77 +- 1.97	68.37 +- 0.77
Sen.	64.66 +- 0.0	63.41 +- 0.0	49.16 +- 1.26	50.92 +- 3.08	57.26 +- 3.03	59.36 +- 1.25
AUC	63.65 +- 0.0	68.33 +- 0.11	56.24 +- 1.51	61.68 +- 1.56	61.01 +- 0.71	63.87 +- 0.24

As observed on Table 6.3, the classification task obtained better AUCs for 3 out of 4 clusters compared with the best value without stratification (68.33% with Logistic Regression). Particularly, we can see that:

- Cluster 1 obtained a slightly better AUC of 69.62% with Logistic Regression. The Specificity obtained was slightly worse, while the Sensitivity was better.
- Cluster 2 had slightly worse results than the no stratification approach (AUC = 64.53% with Logistic Regression).
- Cluster 3 achieved a considerable improvement in the Sensitivity score, which also reflects on the increase of the AUC score to 72.38% with Logistic Regression.
- Cluster 4 was the one with more significant improvements, with the best algorithm being SVMs. It achieved an improvement of 8.18% on Specificity, 22.3% on Sensitivity, and 15.24% on AUC when compared with the no-stratification results (AUC = 83.57%).

6.3 Discussion and Conclusions

Three out of 4 clusters obtained better AUC scores compared with the no-stratification approach, revealing the potential of a personalized prognostic prediction solution.

The models that obtained the best results were Logistic Regression and SVMs, which usually perform well in small datasets (such as the one used) because of their lower complexity.

Clusters 3 and 4, which contained patients with a slower progression, obtained the most promising results. The slower progression is reflected in records of the same patient varying less and on a longer duration. This longer duration, in turn, can result in more records per patient. When put together, this might result in a larger quantity of similar records (i.e., a lower heterogeneity of the data), which might aid the classification task.

Cluster 1 presents the opposite characteristics, having patients with a faster progression and shorter duration. However, records are relatively similar between them, having a lower heterogeneity, which might once again aid the classifier.

The cluster that achieved the worst results was cluster 2, the largest cluster. One hypothesis is that, due to its larger size, it contains more heterogeneous patients with different progression, leading to increased difficulty in predicting the need for NIV for a specific record. This heterogeneity of cluster 2 patients was, in fact, already observed on Chapter 5 on some of the features, such as limbs side and impairment.

Heterogeneity can be an impairing factor for the classification, especially when working with such small partitions. Predicting a record outcome if the training sample only contained heterogeneous val-

ues, possibly different from the one at hand, presents a challenge to a classifier. If, on the contrary, it is using a set of relatively similar records, it is easier to predict the desired outcome of a particular record after analyzing only a few of them.

Additionally, the current approach evaluates the performance of the algorithms considering each record independently. However, the clusters were obtained by taking advantage of the temporal dimension of each patient. A more complex prediction mechanism could be required in order to take advantage of the temporal information and explore the full potential of the subpopulations obtained by DTEC.

The general imbalance, even on the time-window chosen, provides a challenge for this approach and points to the need for new time-windows in future works. This inadequacy results from a change in the definition of the target variable. In previous works that used an older version of the dataset, the need of NIV was measured by whether the patient started to use NIV in that time-window. However, it was noticed that some patients were using NIV without requiring it (by their request, they started NIV preventively, for example). The definition was then changed to be related to ALS-FRS-R, more specifically, to topics 10 and 12 - Dyspnea (difficulty breathing) and Breathing insufficiency, respectively. Now, a patient is considered to require NIV in a time-window if it contains an appointment where: ($P10 \leq 1$ or $P12 \leq 3$) or $Old\_P10 \leq 2$ (for the case of the Portuguese ALS Dataset where the patients still use the old scores and not the revised ones, i.e., when the new P10 and P12 are null). Although clinically relevant, this change provokes a higher imbalance of the current time-windows, which will harm the results obtained. Larger time-windows (i.e., 1.5 years or 2 years) can mitigate this effect and result in higher-quality data for the classification task.

In conclusion, although restricted by some factors, including the limitations of the dataset, the results obtained contributed to showing the relevance of a personalized prognostic prediction solution in the ALS context.

6.4 Summary

In this chapter, the pipeline of the classification task implemented is described, and the corresponding results are presented.

The overall solution architecture is exposed together with the training and evaluation methodology. The classification pipeline includes Balancing using SMOTE, Scaling (when working with SVM), and the Classifier steps.

A series of state-of-the-art machine learning classifiers were experimented with: Naive Bayes (NB), Logistic Regression (LR), K nearest neighbors (KNN), Decision Trees (DT), Random Forests (RF). Furthermore, each cluster was used as input for the classification pipeline in order to obtain personalized predictions.

For each cluster the best classifier was chosen and compared with the results without stratification. Some clusters obtained considerable improvements. The main conclusions are exposed and discussed.

The potential of a personalized solution is shown, although restricted by some current limitations (i.e., time-windows available, size of the dataset, etc.).

7

Conclusion

Contents

7.1 Main Conclusions

This work proposed a stratification method using deep learning techniques to unveil the temporal disease patterns of ALS and produce clinically relevant clusters of patients. Using a processed subset of the Portuguese ALS dataset, this work introduced one of the first specifically tailored solutions for ALS temporal patient stratification, the Deep Temporal Encoded Clustering (DTEC) method.

To increase the robustness of this solution, a Consensus Clustering approach was implemented, using the accumulated results obtained by running DTEC several times and saving the outcomes with different numbers of clusters.

Using only the first three records of each patient, the method was able to obtain four clinically relevant sub-populations within ALS patients, distinguished by their different evolution and clinical presentations of the disease and achieving a silhouette score of 0.75. A detailed characterization of these subgroups was presented and compared with the subpopulations obtained with different configurations of this same approach.

Some key insights obtained include that bulbar onset patients showed to be related to faster progression. In contrast, limbs onset patients varied their progression from slow to fast, with the main onset differences between sub-populations being the limbs side and type of impairment. Notably, patients that showed distal limb impairment are associated with a slower progression.

The main focus of this work is on the temporal patient stratification methodology in order to obtain medically relevant subgroups; nonetheless, a prognosis prediction pipeline was also implemented. The classification pipeline was mainly developed as a validation mechanism to test the possible improvements regarding the prediction of the need of NIV when considering each cluster separately. These improvements were evident, particularly in some of the clusters found, reaching AUC values up to 83.57%, contrasting to the 68.33% AUC without stratification. Although limited by some factors, which resulted in the improvements not being so evident in some of the clusters, the classification pipeline contributed to showcasing the potential of personalized prognostic prediction.

7.2 Future Work

Further exploration of prognostic prediction methods using the obtained clusters might produce even more significant improvements. For example, an approach that considers each patient instead of each record would be highly relevant. This can be done, for example, by using the DTEC's TAE + UMAP encoding or by using a new encoding module specific to the classification task. It is also pertinent to explore other classification mechanisms besides the ones already used, for example, Deep Learning classification methods.

Another relevant follow-up task would be to integrate the optimization process of DTEC with the prediction pipeline, producing an end-to-end mechanism. This way, DTEC would be optimized by considering the improvements of the classification metrics (Specificity, Sensitivity, and AUC) and not only clustering metrics such as the silhouette.

Alternative solutions to improve the stability of DTEC results, besides the implemented Consensus Clustering mechanism, can also be studied. An additional proposed future work is to explore data augmentation techniques to increase the quantity of training data available since the small size of the dataset is one of the factors considered to impact the results obtained negatively.

Finally, it would be interesting to see how this method performs in other contexts, using a different cohort of ALS patients or even adapting it to another disease.

Bibliography

[1] N. S. Madiraju, “Deep temporal clustering: Fully unsupervised learning of time-domain features,” Ph.D. dissertation, Arizona State University, 2018.

[2] A. Zhen, M. Kim, and G. Wu, “Disentangling the spatio-temporal heterogeneity of alzheimer’s disease using a deep predictive stratification network,” in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 2021, pp. 46–49.

[3] C. Lee and M. van der Schaar, “Temporal Phenotyping using Deep Predictive Clustering of Disease Progression,” 2020. [Online]. Available: https://arxiv.org/abs/2006.08600

[4] ——, “Temporal Phenotyping using Deep Predictive Clustering of Disease Progression Presentation,” https://papertalk.org/papertalks/5851, Last accessed 13 Dec 2021.

[5] I. Landi, B. S. Glicksberg, H.-C. Lee, S. Cherng, G. Landi, M. Danieletto, J. T. Dudley, C. Furlanello, and R. Miotto, “Deep representation learning of electronic health records to unlock patient stratification at scale,” *npj Digital Medicine*, vol. 3, no. 1, Jul. 2020. [Online]. Available: https://doi.org/10.1038/s41746-020-0301-z

[6] C. A. Brown, C. Lally, V. Kupelian, and W. D. Flanders, “Estimated prevalence and incidence of amyotrophic lateral sclerosis and SOD1 and c9orf72 genetic variants,” *Neuroepidemiology*, vol. 55, no. 5, pp. 342–353, Jul. 2021.

[7] “ALS Association ”Understanding ALS: What is ALS?” page,” https://www.als.org/understanding-als/what-is-als, Last accessed 17 Dec 2021.

[8] B. Vrijsen, D. Testelmans, C. Belge, W. Robberecht, P. Van Damme, and B. Buyse, “Non-invasive ventilation in amyotrophic lateral sclerosis,” *Amyotroph. Lateral Scler. Frontotemporal Degener.*, vol. 14, no. 2, pp. 85–95, Mar. 2013.

[9] A. V. Carreiro, P. M. T. Amaral, S. Pinto, P. Tomás, M. de Carvalho, and S. C. Madeira, “Prognostic models based on patient snapshots and time windows: Predicting disease progression to assisted ventilation in amyotrophic lateral sclerosis,” *J. Biomed. Inform.*, vol. 58, pp. 133–144, Dec. 2015.

[10] A. S. Martins, M. Gromicho, S. Pinto, M. d. Carvalho, and S. C. Madeira, “Learning prognostic models using DiseaseProgression patterns: Predicting the need fornon-invasive ventilation in amyotrophic LateralSclerosis,” *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. PP, pp. 1–1, May 2021.

[11] S. Pires, M. Gromicho, S. Pinto, M. de Carvalho, and S. C. Madeira, “Patient stratification using clinical and patient profiles: Targeting personalized prognostic prediction in ALS,” in *Bioinformatics and Biomedical Engineering*, ser. Lecture notes in computer science. Cham: Springer International Publishing, 2020, pp. 529–541.

[12] T. Ayer, O. Alagoz, J. Chhatwal, J. W. Shavlik, C. E. Kahn, Jr, and E. S. Burnside, “Breast cancer risk estimation with artificial neural networks revisited,” *Cancer*, vol. 116, no. 14, pp. 3310–3321, Apr. 2010.

[13] A. Singh, G. Nadkarni, O. Gottesman, S. B. Ellis, E. P. Bottinger, and J. V. Guttag, “Incorporating temporal EHR data in predictive models for risk stratification of renal function deterioration,” *J. Biomed. Inform.*, vol. 53, pp. 220–228, Feb. 2015.

[14] S. Martin, A. Al Khleifat, and A. Al-Chalabi, “What causes amyotrophic lateral sclerosis?” *F1000Res.*, vol. 6, p. 371, Mar. 2017.

[15] B. Conde, J. C. Winck, and L. F. Azevedo, “Estimating amyotrophic lateral sclerosis and motor neuron disease prevalence in portugal using a pharmaco-epidemiological approach and a bayesian multiparameter evidence synthesis model,” *Neuroepidemiology*, vol. 53, no. 1-2, pp. 73–83, May 2019.

[16] “Mayo Clinic ALS information page,” https://www.mayoclinic.org/diseases-conditions/amyotrophic-lateral-sclerosis/symptoms-causes/syc-20354022, Last accessed 30 Nov 2021.

[17] W. Siirala, R. Aantaa, K. T. Olkkola, T. Saaresranta, and A. Vuori, “Is the effect of non-invasive ventilation on survival in amyotrophic lateral sclerosis age-dependent?” *BMC Palliat. Care*, vol. 12, no. 1, p. 23, May 2013.

[18] J. Dorst and A. C. Ludolph, “Non-invasive ventilation in amyotrophic lateral sclerosis,” *Ther. Adv. Neurol. Disord.*, vol. 12, p. 1756286419857040, Jun. 2019.

[19] R. Balendra, A. Al Khleifat, T. Fang, and A. Al-Chalabi, “A standard operating procedure for king’s ALS clinical staging,” *Amyotroph. Lateral Scler. Frontotemporal Degener.*, vol. 20, no. 3-4, pp. 159–164, May 2019.

[20] I. Tramacere, E. Dalla Bella, A. Chiò, G. Mora, G. Filippini, G. Lauria, and EPOS Trial Study Group, “The MITOS system predicts long-term survival in amyotrophic lateral sclerosis,” *J. Neurol. Neurosurg. Psychiatry*, vol. 86, no. 11, pp. 1180–1185, Nov. 2015.

[21] B. K. Beaulieu-Jones, W. Yuan, G. A. Brat, A. L. Beam, G. Weber, M. Ruffin, and I. S. Kohane, "Machine learning for patient risk stratification: standing on, or looking over, the shoulders of clinicians?" *NPJ Digit. Med.*, vol. 4, no. 1, p. 62, Mar. 2021.

[22] F. Fernandes, I. Barbalho, D. Barros, R. Valentim, C. Teixeira, J. Henriques, P. Gil, and M. Dourado Júnior, "Biomedical signals and machine learning in amyotrophic lateral sclerosis: a systematic review," *Biomed. Eng. Online*, vol. 20, no. 1, p. 61, Jun. 2021.

[23] C. M. Salgado and S. M. Vieira, "Machine learning for patient stratification and classification part 2: Unsupervised learning with clustering," in *Leveraging Data Science for Global Health*. Cham: Springer International Publishing, 2020, pp. 151–168.

[24] D. Ramamoorthy, K. Severson, S. Ghosh, K. Sachs, J. D. Glass, C. N. Fournier, J. Berry, K. Ng, E. Fraenkel, Answer ALS, and Pooled Resource Open-Access ALS Clinical Trials Consortium, "Identifying patterns of ALS progression from sparse longitudinal data," May 2021.

[25] X. Li and K.-C. Wong, "Evolutionary multiobjective clustering and its applications to patient stratification," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1680–1693, May 2019.

[26] S. Khakabimamaghani and M. Ester, "Bayesian biclustering for patient stratification," *Pac. Symp. Biocomput.*, vol. 21, pp. 345–356, 2016.

[27] H. Liu, R. Zhao, H. Fang, F. Cheng, Y. Fu, and Y. Y. Liu, "Entropy-based consensus clustering for patient stratification," *Bioinformatics*, vol. 33, no. 17, pp. 2691–2698, 2017.

[28] "ARCAS Homepage," https://arcas.ai/, Last accessed 29 Nov 2021.

[29] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Comput. Struct. Biotechnol. J.*, vol. 13, pp. 8–17, 2015.

[30] E. Lin, J. L. Hefner, X. Zeng, S. Moosavinasab, T. Huber, J. Klima, C. Liu, and S. M. Lin, "A deep learning model for pediatric patient risk stratification." *The American journal of managed care*, vol. 25 10, pp. e310–e315, 2019.

[31] J. D. Berry, A. A. Taylor, D. Beaulieu, L. Meng, A. Bian, J. Andrews, M. Keymer, D. L. Ennist, and B. Ravina, "Improved stratification of ALS clinical trials using predicted survival," *Ann. Clin. Transl. Neurol.*, vol. 5, no. 4, pp. 474–485, Apr. 2018.

[32] R. Kueffner, N. Zach, M. Bronfeld, R. Norel, N. Atassi, V. Balagurusamy, B. Di Camillo, A. Chio, M. Cudkowicz, D. Dillenberger, J. Garcia-Garcia, O. Hardiman, B. Hoff, J. Knight, M. L. Leitner, G. Li, L. Mangravite, T. Norman, L. Wang, ALS Stratification Consortium, J. Xiao, W.-C. Fang,

J. Peng, C. Yang, H.-J. Chang, and G. Stolovitzky, “Stratification of amyotrophic lateral sclerosis patients: a crowdsourcing approach,” *Sci. Rep.*, vol. 9, no. 1, p. 690, Jan. 2019.

[33] D. Ramamoorthy, K. Severson, S. Ghosh, K. Sachs, J. D. Glass, C. N. Fournier, J. Berry, K. Ng, E. Fraenkel, Answer ALS, and Pooled Resource Open-Access ALS Clinical Trials Consortium, “Identifying patterns of ALS progression from sparse longitudinal data,” May 2021.

[34] A. Carreiro, “An integrative mining approach for prognostic prediction in neurodegenerative diseases,” Ph.D. dissertation, Instituto Superior Técnico, Lisbon,Portugal, 2016.

[35] P. Huang, C. T. Lin, Y. Li, M. C. Tammemagi, M. V. Brock, S. Atkar-Khattra, Y. Xu, P. Hu, J. R. Mayo, H. Schmidt, M. Gingras, S. Pasian, L. Stewart, S. Tsai, J. M. Seely, D. Manos, P. Burrowes, R. Bhatia, M.-S. Tsao, and S. Lam, “Prediction of lung cancer risk at follow-up screening with low-dose CT: a training and validation study of a deep learning method,” *Lancet Digit Health*, vol. 1, no. 7, pp. e353–e362, Nov. 2019.

[36] G. Batista, X. Wang, and E. Keogh, “A complexity-invariant distance measure for time series,” 04 2011, pp. 699–710.

[37] X. Golay, S. S. Kollias, G. Stoll, D. Meier, A. Valavanis, and P. Boesiger, “A new correlation-based fuzzy logic clustering algorithm for fmri,” *Magnetic Resonance in Medicine*, vol. 40, 1998.

[38] P. Galeano and D. Peña, “Multivariate analysis in vector time series,” *Resenhas*, pp. 383–404, 2000.

[39] H. K. van der Burgh, R. Schmidt, H.-J. Westeneng, M. A. de Reus, L. H. van den Berg, and M. P. van den Heuvel, “Deep learning predictions of survival based on MRI in amyotrophic lateral sclerosis,” *NeuroImage Clin.*, vol. 13, pp. 361–369, 2017.

[40] A. Fred and A. K. Jain, “Evidence accumulation clustering based on the k-means algorithm,” in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2002, pp. 442–451. [Online]. Available: https://doi.org/10.1007/3-540-70659-3_46

[41] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[42] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” 2018. [Online]. Available: https://arxiv.org/abs/1802.03426

[43] R. McConville, R. Santos-Rodriguez, R. J. Piechocki, and I. Craddock, “N2d: (not too) deep clustering via clustering the local manifold of an autoencoded embedding,” 2020.

[44] M. Allaoui, M. L. Kherfi, and A. Cheriet, “Considerably improving clustering algorithms using UMAP dimensionality reduction technique: A comparative study,” in *Lecture Notes in Computer Science*. Springer International Publishing, 2020, pp. 317–325. [Online]. Available: https://doi.org/10.1007/978-3-030-51935-3_34

[45] “UMAP documentation page : min_dist parameter,” https://umap-learn.readthedocs.io/en/latest/parameters.html#min-dist, Last accessed 19 Oct 2022.

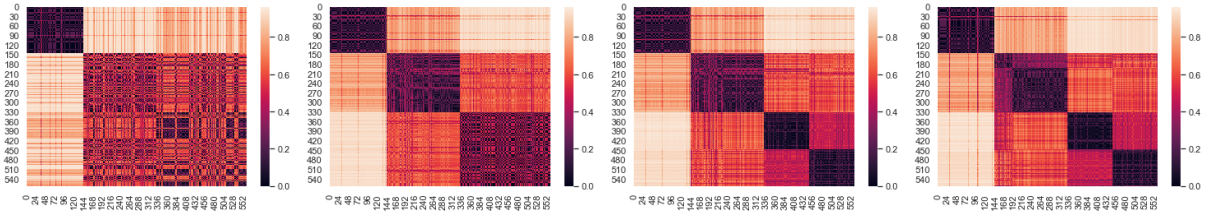
[46] J. R. Green, Y. Yunusova, M. S. Kuruvilla, J. Wang, G. L. Pattee, L. Synhorst, L. Zinman, and J. D. Berry, “Bulbar and speech motor assessment in ALS: Challenges and future directions,” *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration*, vol. 14, no. 7-8, pp. 494–500, Jul. 2013. [Online]. Available: https://doi.org/10.3109/21678421.2013.817585

[47] Q. Zhang, C. Mao, J. Jin, C. Niu, L. Bai, J. Dang, and M. Zhang, “Side of limb-onset predicts laterality of gray matter loss in amyotrophic lateral sclerosis,” *BioMed Research International*, vol. 2014, pp. 1–11, 2014. [Online]. Available: https://doi.org/10.1155/2014/473250

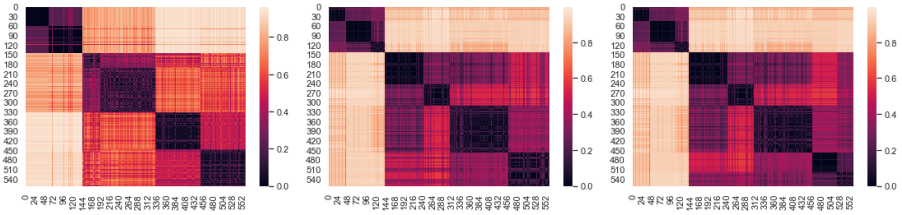
A

Choosing the Best Number of Clusters - Additional Results

This appendix chapter presents additional distance matrix heatmap visualizations for K number of clusters from 2 to 8, used to assess the best number of clusters for each Consensus Clustering configuration. Information regarding how these visualizations were obtained and used is present on Chapter 5.

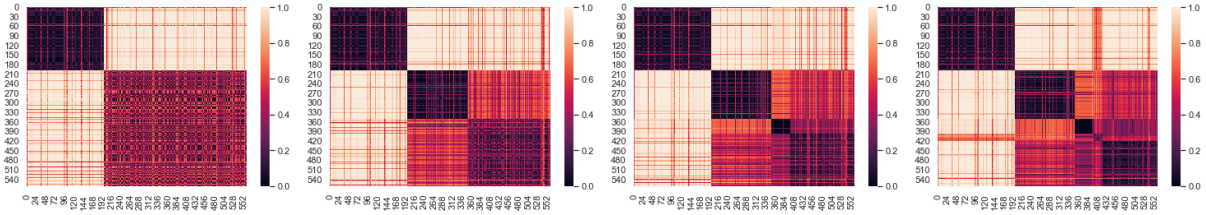


(a) K=2 **(b)** K=3 **(c)** K=4 **(d)** K=5

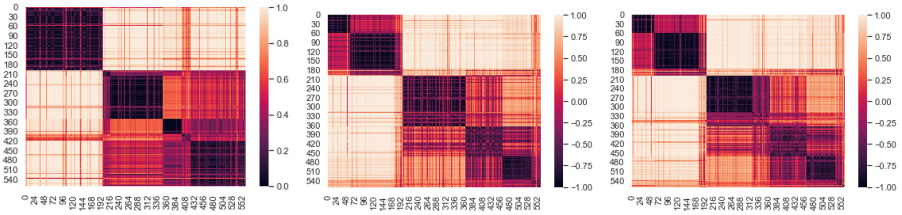


(e) K=6 **(f)** K=7 **(g)** K=8

Figure A.1: Distance Matrix ordered Heatmap visualizations for each K number of clusters for the Consensus Clustering method using the complete feature set

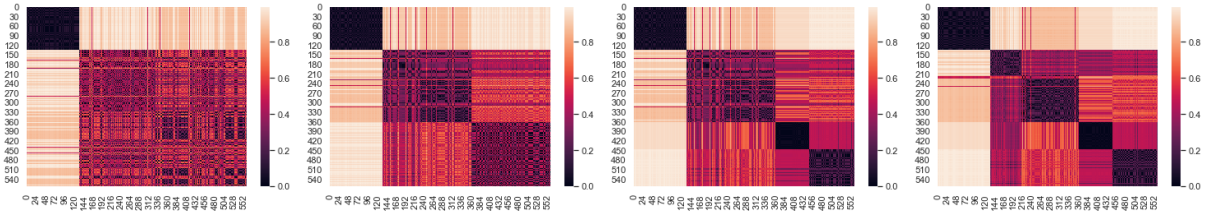


(a) K=2 **(b)** K=3 **(c)** K=4 **(d)** K=5

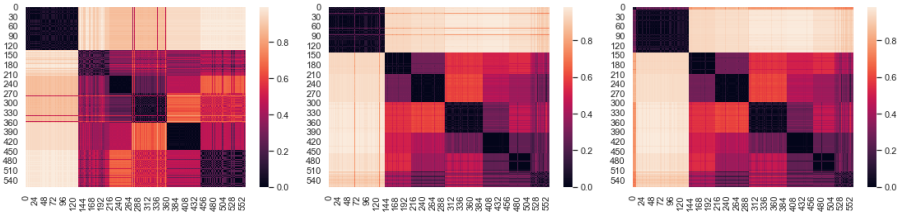


(e) K=6 **(f)** K=7 **(g)** K=8

Figure A.2: Distance Matrix ordered Heatmap visualizations for each K number of clusters for the Consensus Clustering method using only temporal features



(a) K=2 **(b)** K=3 **(c)** K=4 **(d)** K=5



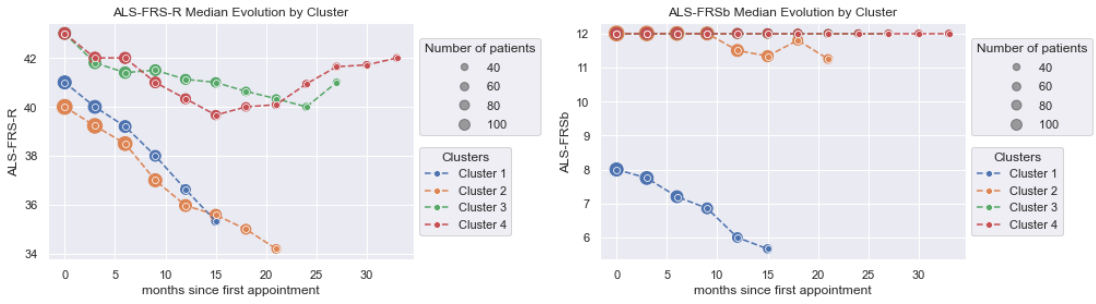
(e) K=6 **(f)** K=7 **(g)** K=8

Figure A.3: Distance Matrix ordered Heatmap visualizations for each K number of clusters for the Consensus Clustering method using only the First Record

B

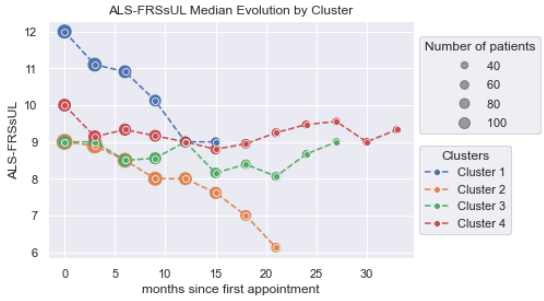
Clusters Characterization - Additional Results

This appendix chapter presents additional visualizations regarding temporal features evolution and static features distribution. For each Consensus Clustering configuration, it showcases some already presented features, where the differences among clusters are evident (see Chapter 5) and others where these differences are considered less relevant.

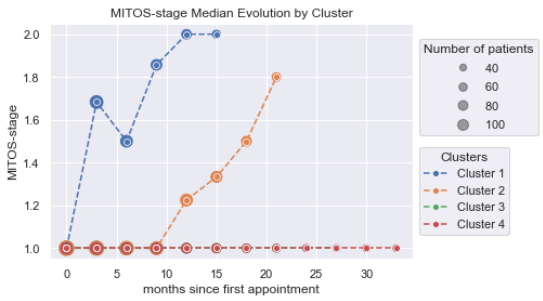


(a) ALS-FRS-R

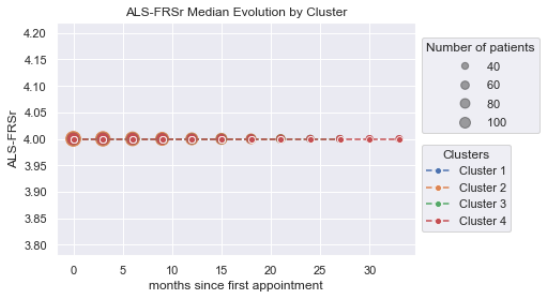
(b) ALS-FRSb



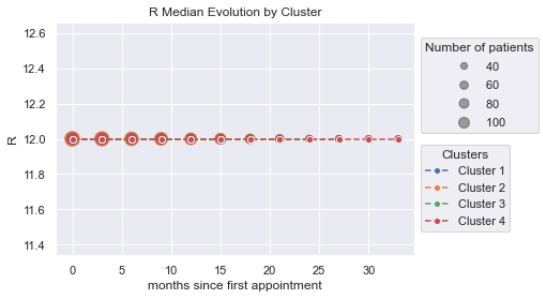
(c) ALS-FRSsUL



(d) MiToS



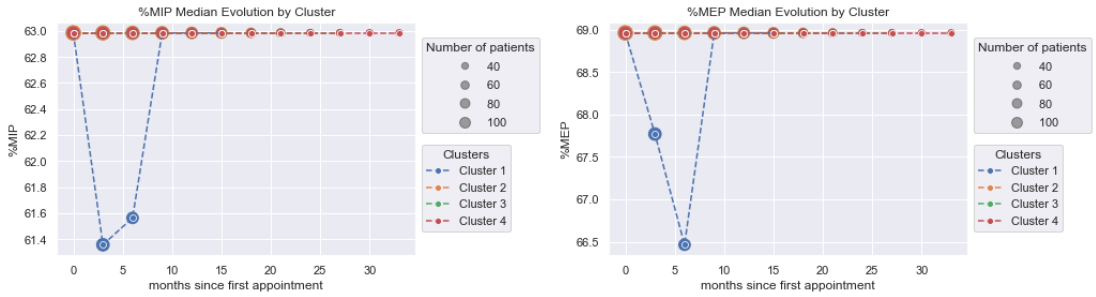
(e) ALS-FRSr



(f) R

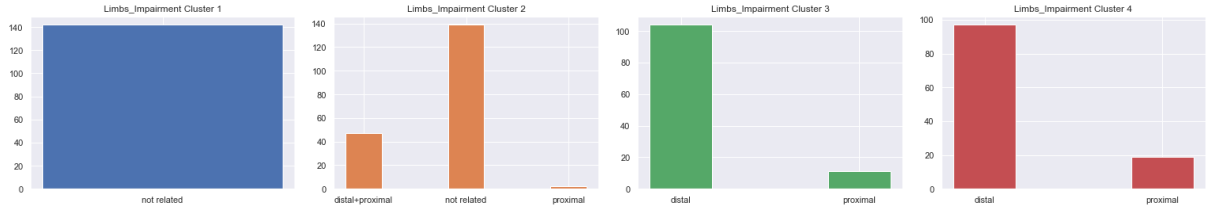
Figure B.1: Temporal Features Median Evolution across Clusters - Complete Feature Set configuration (Part 1 of 2)

(a) %VC **(b)** %FVC



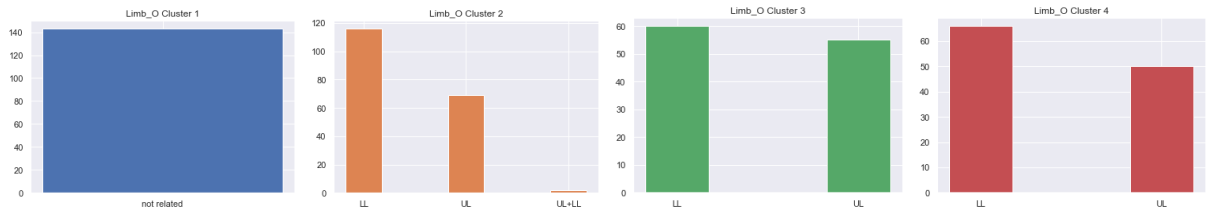
(c) %MIP **(d)** %MEP

Figure B.2: Temporal Features Median Evolution across Clusters - Complete Feature Set configuration (Part 2 of 2)



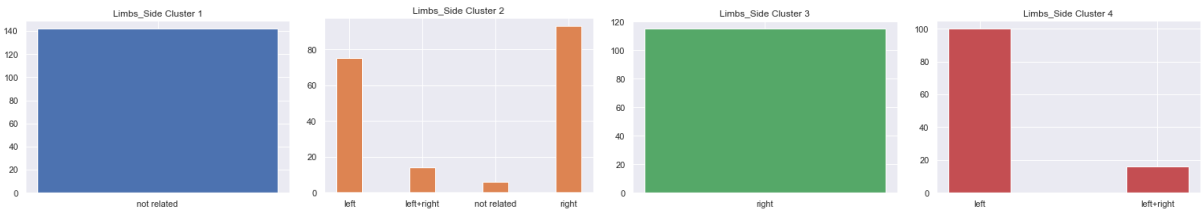
(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure B.3: Limbs Impairment Distribution by Cluster - Complete Feature Set configuration



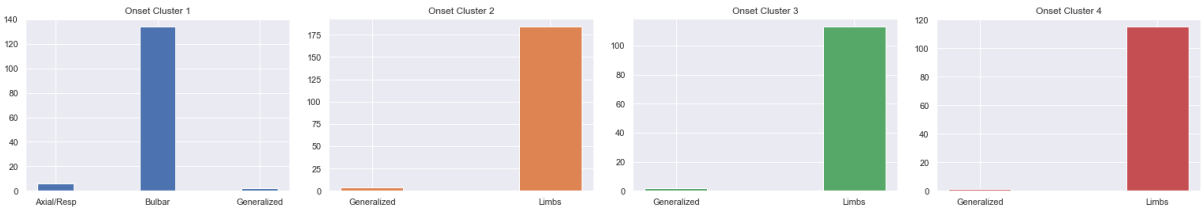
(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure B.4: Limbs Onset Distribution by Cluster - Complete Feature Set configuration



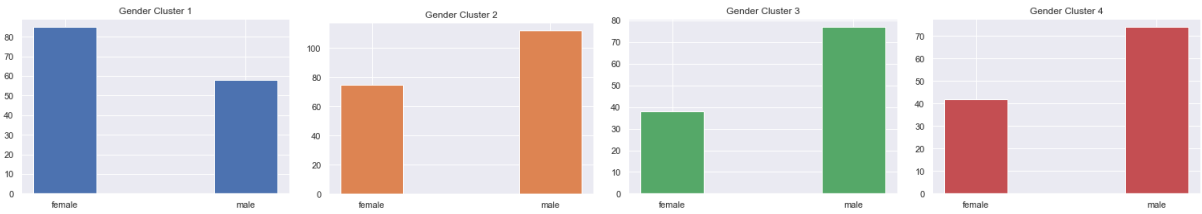
(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure B.5: Limbs Side Distribution by Cluster - Complete Feature Set configuration



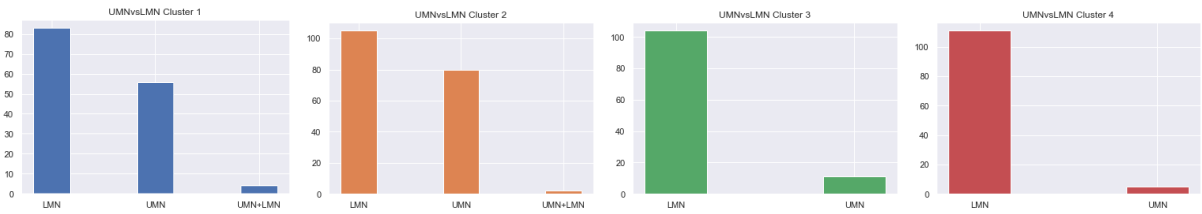
(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure B.6: Onset Distribution by Cluster - Complete Feature Set configuration



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure B.7: Gender Distribution by Cluster - Complete Feature Set configuration

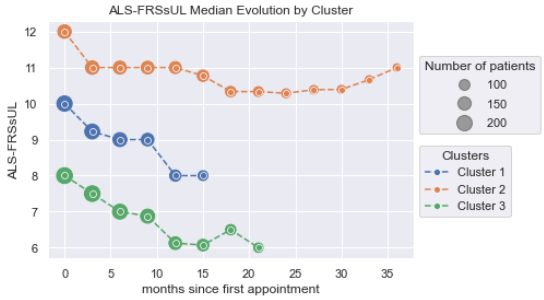


(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

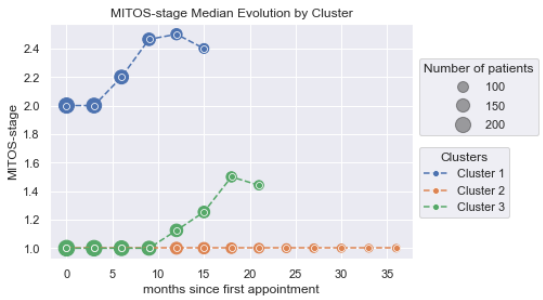
Figure B.8: UMNvsLMN Distribution by Cluster - Complete Feature Set configuration

(a) ALS-FRS-R

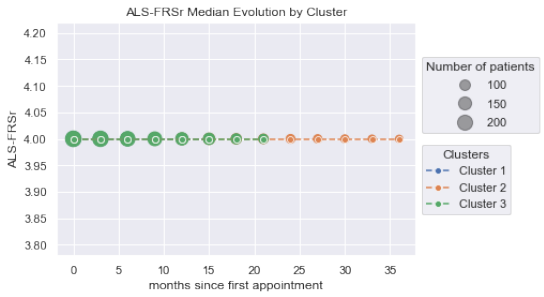
(b) ALS-FRSb



(c) ALS-FRSsUL



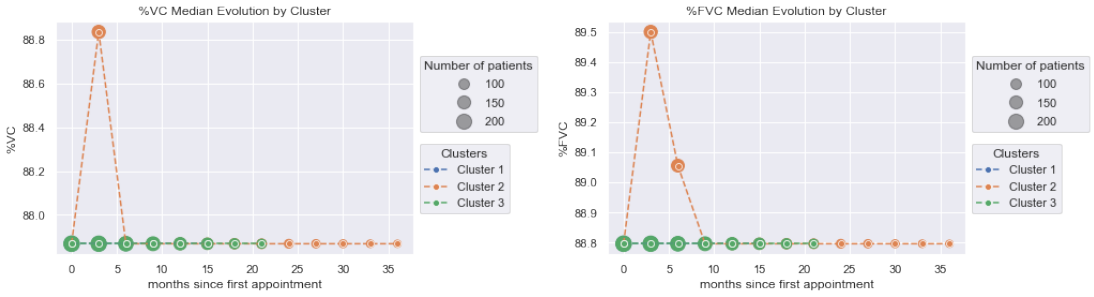
(d) MiToS



(e) ALS-FRSr

(f) R

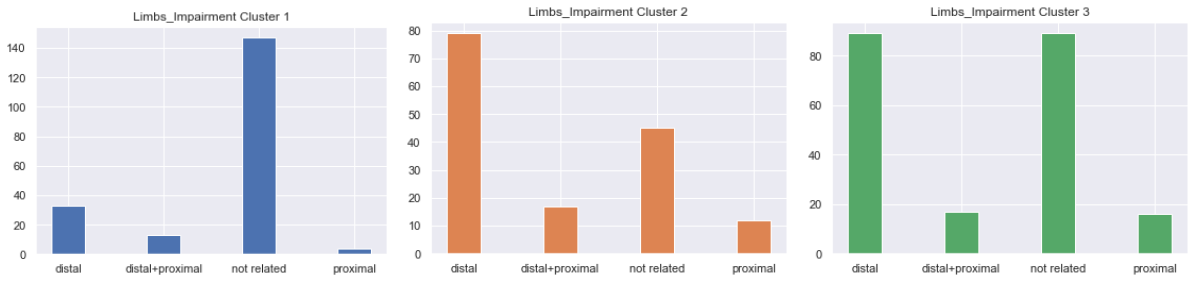
Figure B.9: Temporal Features Median Evolution across Clusters -Temporal Only configuration (Part 1 of 2)



(a) %VC **(b)** %FVC

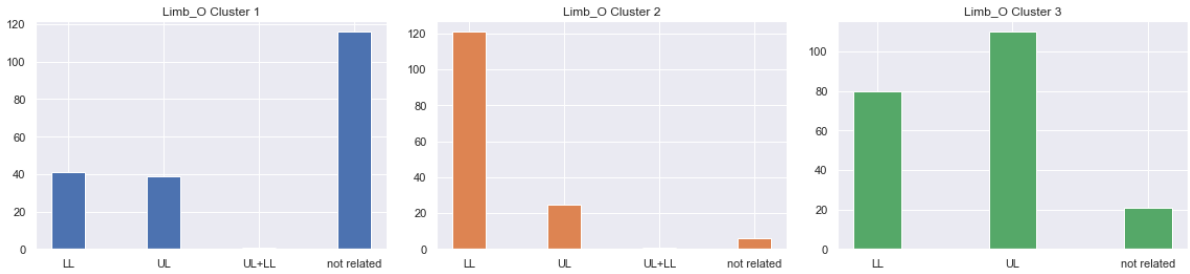
(c) %MIP **(d)** %MEP

Figure B.10: Temporal Features Median Evolution across Clusters - Temporal Only configuration (Part 2 of 2)



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3

Figure B.11: Limbs Impairment Distribution by Cluster - Temporal Only configuration

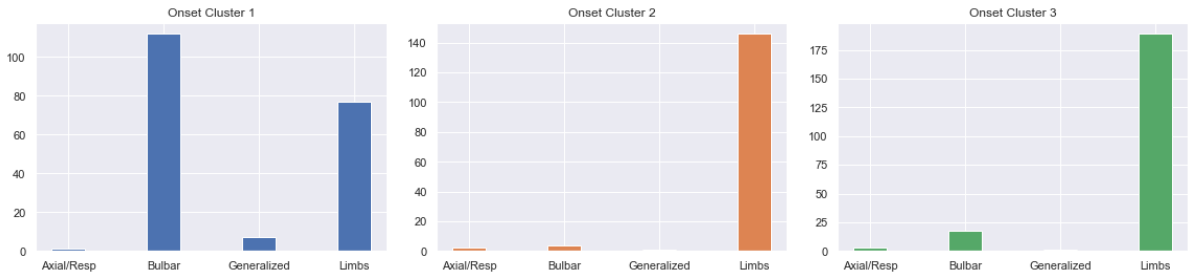


(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3

Figure B.12: Limbs Onset Distribution by Cluster - Temporal Only configuration

(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3

Figure B.13: Limbs Side Distribution by Cluster - Temporal Only configuration

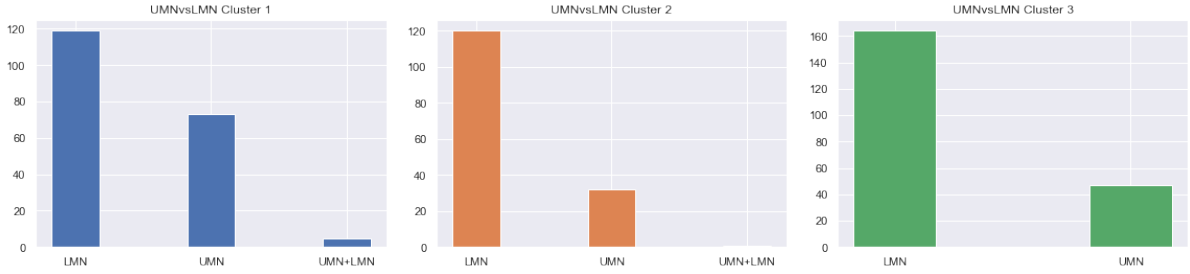


(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3

Figure B.14: Onset Distribution by Cluster - Temporal Only configuration

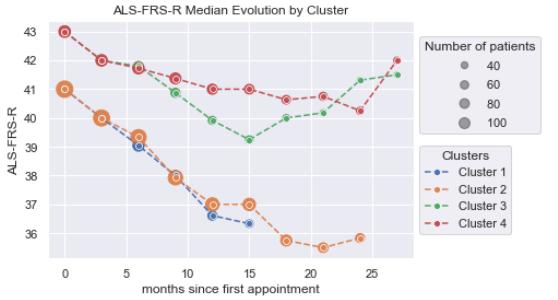
(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3

Figure B.15: Gender Distribution by Cluster - Temporal Only configuration

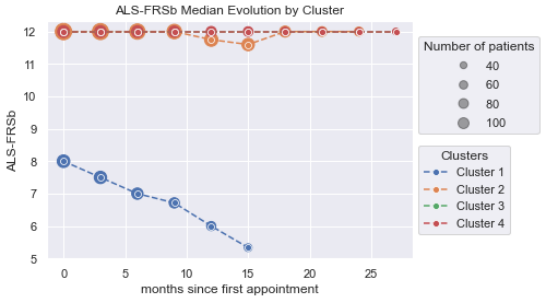


(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3

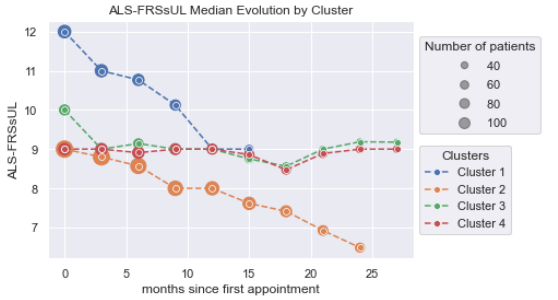
Figure B.16: UMNvsLMN Distribution by Cluster - Temporal Only configuration



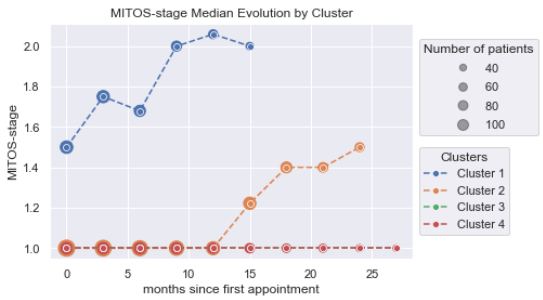
(a) ALS-FRS-R



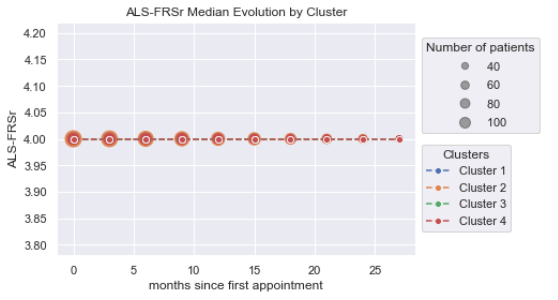
(b) ALS-FRSb



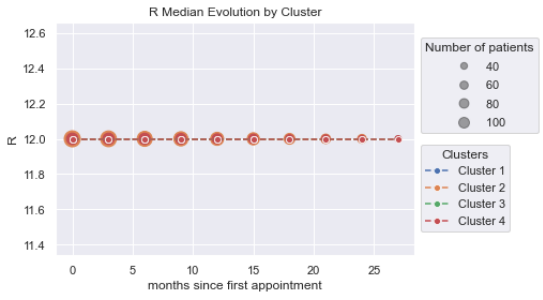
(c) ALS-FRSsUL



(d) MiToS

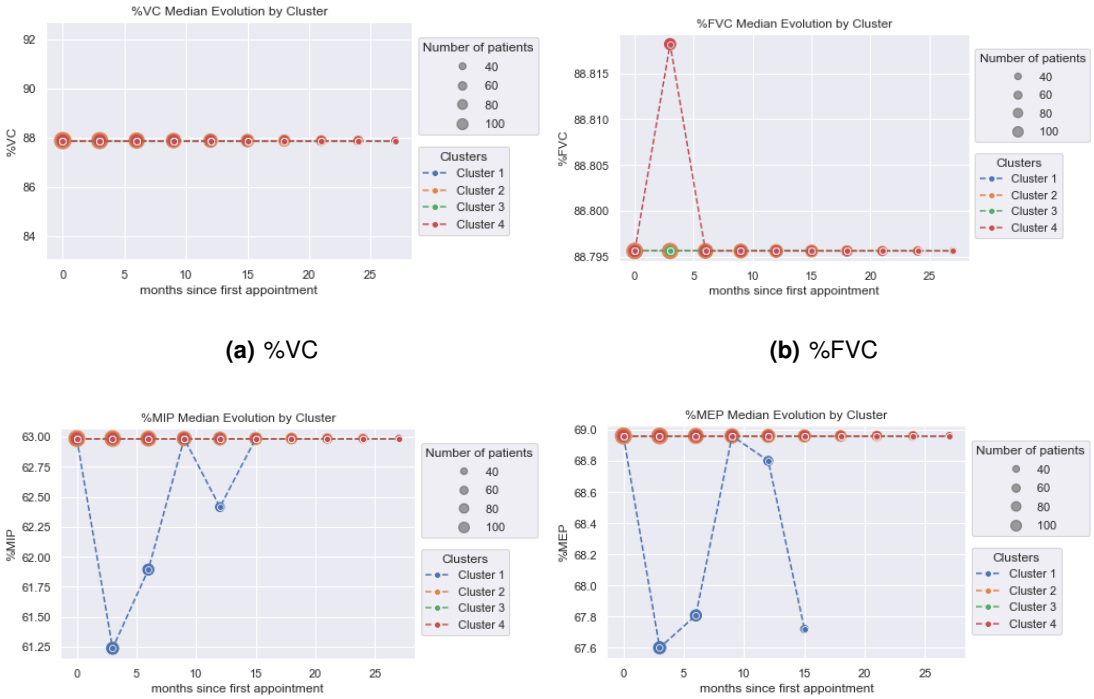


(e) ALS-FRSr



(f) R

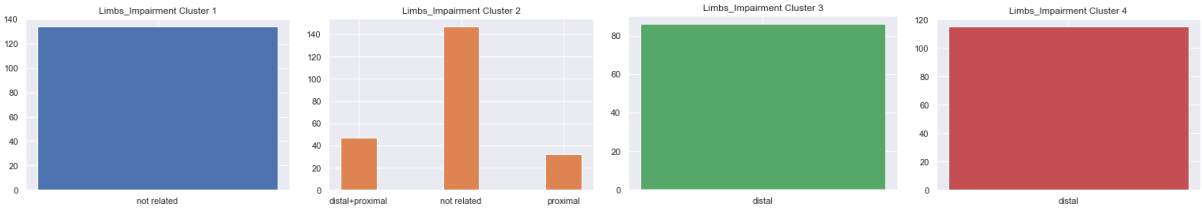
Figure B.17: Temporal Features Median Evolution across Clusters - First Record Only configuration (Part 1 of 2)



(a) %VC **(b)** %FVC

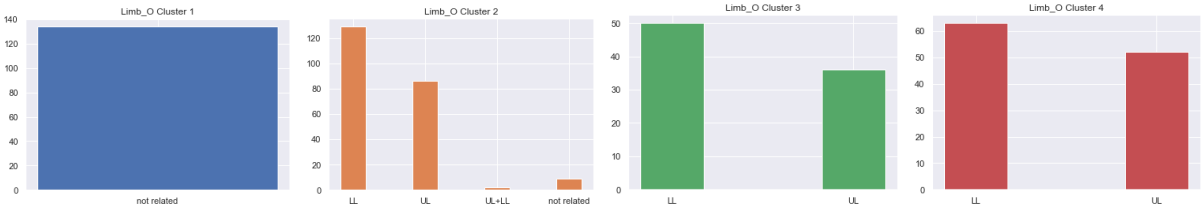
(c) %MIP **(d)** %MEP

Figure B.18: Temporal Features Median Evolution across Clusters - First Record Only configuration (Part 2 of 2)



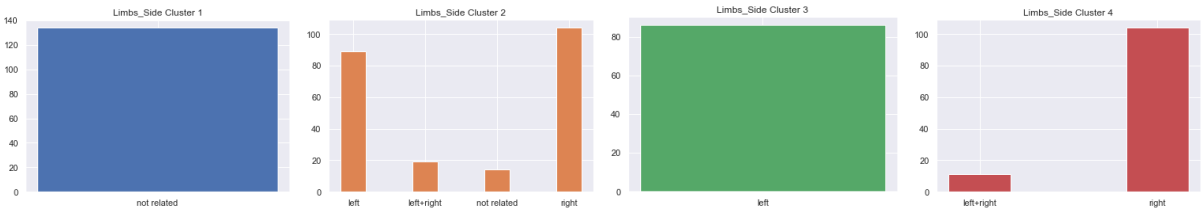
(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure B.19: Limbs Impairment Distribution by Cluster - First Record Only configuration



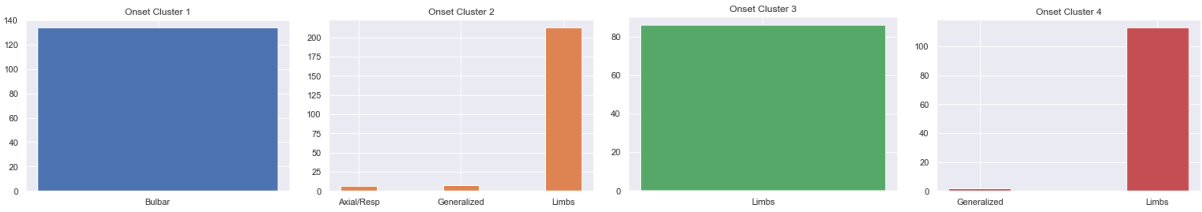
(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure B.20: Limbs Onset Distribution by Cluster - First Record Only configuration



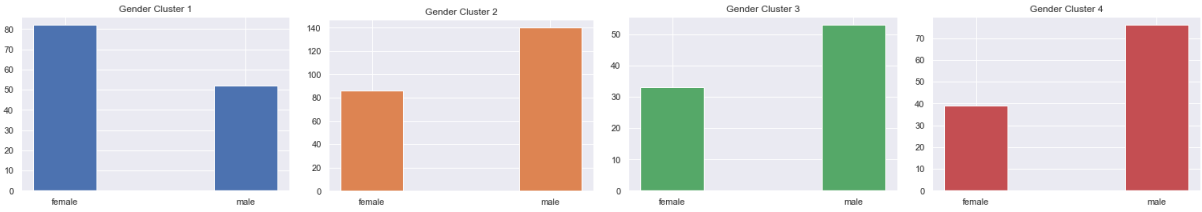
(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure B.21: Limbs Onset Distribution by Cluster - First Record Only configuration



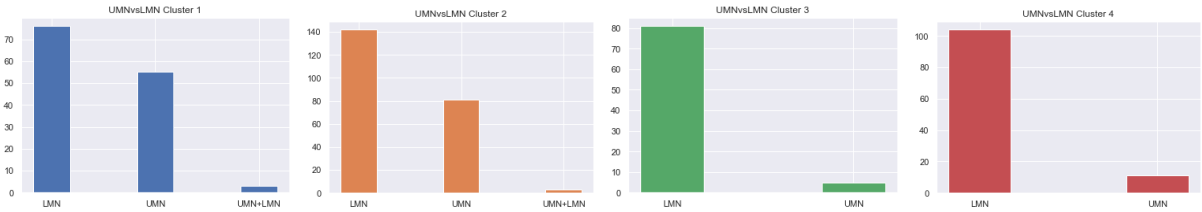
(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure B.22: Onset Distribution by Cluster - First Record Only configuration



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure B.23: Gender Distribution by Cluster - First Record Only configuration



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3 **(d)** Cluster 4

Figure B.24: UMNvsLMN Distribution by Cluster - First Record Only configuration