Learning Prognostic Models Using Disease Progression Patterns: Predicting the Need for Non-Invasive Ventilation in Amyotrophic Lateral Sclerosis

Andreia S. Martins, Marta Gromicho, Susana Pinto, Mamede de Carvalho, and Sara C. Madeira

Abstract—Amyotrophic Lateral Sclerosis is a devastating neurodegenerative disease causing rapid degeneration of motor neurons and usually leading to death by respiratory failure. Since there is no cure, treatment's goal is to improve symptoms and prolong survival. Non-invasive Ventilation (NIV) is an effective treatment, leading to extended life expectancy and improved quality of life. In this scenario, it is paramount to predict its need in order to allow preventive or timely administration. In this work, we propose to use itemset mining together with sequential pattern mining to unravel disease presentation patterns together with disease progression patterns by analysing, respectively, static data collected at diagnosis and longitudinal data from patient follow-up. The goal is to use these static and temporal patterns as features in prognostic models, enabling to take disease progression into account in predictions and promoting model interpretability. As case study, we predict the need for NIV within 90, 180 and 365 days (short, mid and long-term predictions). The learnt prognostic models are promising. Pattern evaluation through growth rate suggests bulbar function and phrenic nerve response amplitude, additionally to respiratory function, are significant features towards determining patient evolution. This confirms clinical knowledge regarding relevant biomarkers of disease progression towards respiratory insufficiency.

Index Terms—Amyotrophic lateral sclerosis, pattern mining, disease progression patterns, non-invasive ventilation, prognostic models

1 Introduction

Amyotrophic Lateral Sclerosis (ALS) is a neurodegenerative disease with no cure and a short life expectancy after symptom onset, with most patients dying within 3 to 5 years due to respiratory failure [3]. Most available treatments focus on symptom management and only a few were shown to increase life expectancy by a few months. In this context, Non-invasive Ventilation (NIV) showed most promising results in increasing both life expectancy and quality of life, with evidence suggesting an earlier initiation provides greater benefit [37]. However, not all patients are promptly eligible and there is some debate on the best timing to initiate NIV.

In this scenario, it is important to investigate whether it is possible to effectively predict if ALS patients will be eligible for NIV in the near future based on their evolution, enabling timely NIV administration, promoting patient wellbeing and improving survival. We thus formulated the following prognostic prediction problem, schematized in Fig. 1: given a specific ALS patient static data collected at diagnosis and temporal data from disease follow-up, can we effectively use these clinical evaluations to predict if this patient will require NIV within k days of last evaluation?

Our goal is therefore to learn a prognostic model that predicts whether a patient will require NIV within k days of the last evaluation using static data collected at diagnosis and longitudinal data collected at patient's follow-up. However, instead of using the original data directly as features [25], [28], [29], [30], our goal is to learn patterns from data and use these patterns as features when learning the prognostic model. We thus tackle the learning of both disease presentation patterns and disease progression patterns, discovered in static and temporal data, respectively. This enables us to take disease progression into account in the predictions promoting not only model predictability but also model interpretability. Unlike when prognostic models are learnt directly from the original static and/or temporal features, which can only be done by using training examples with the same number of time points for all patients or introducing artificial missing values, in the proposed approach we enable the use of all available time points for each patient without adding missing values. This enables us to use more learning instances to train the models, while finding putatively interesting disease presentation and progression patterns in the process. In this context, we propose to use pattern mining and sequential pattern mining algorithms to extract patterns from the original static and

- *Andreia S. Martins and Sara C. Madeira are with LASIGE and the Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa in Lisbon, 1649-004 Lisboa, Portugal. E-mail: amartins@lasige.di.fc.ul.pt, sacmadeira@ciencias.ulisboa.pt.*
- *Marta Gromicho, Susana Pinto, and Mamede de Carvalho are with the Instituto de Medicina Molecular and Instituto de Fisiologia, Faculdade de Medicina, Universidade de Lisboa in Lisbon, 1649-004 Lisboa, Portugal. E-mail: {martalgms, susana.c.pinto}@gmail.com, mamedealves@medicina.ulisboa.pt.*

Manuscript received 17 January 2021; revised 5 April 2021; accepted 28 April 2021. Date of publication 7 May 2021; date of current version 7 October 2022.
(Corresponding author: Sara C. Madeira.)
Digital Object Identifier no. 10.1109/TCBB.2021.3078362

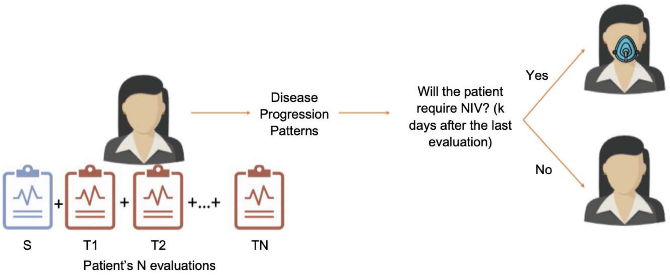


Fig. 1. Problem formulation: given static data collected at diagnosis and temporal data from patient's evaluation over time, can we predict the need for Non-invasive Ventilation (NIV) within *k* days of last evaluation?

temporal features, respectively. New datasets are then built from these patterns, considering them as features, and used to generate and evaluate predictive models, learnt using state-of-the-art classification algorithms, such as Random Forests [38], Support Vectors Machines (SVMs) [39], Naive Bayes and XGBoost [1]. Random Undersampling and SMOTE [2] are used to deal with class imbalance. As case study, we use the Portuguese ALS Dataset and predict the need for NIV within 90, 180 and 365 days (short, mid and long-term predictions) with promising results concerning predictability and interesting results concerning interpretability. The progression patterns used by the prognostic models as features confirm clinical knowledge regarding relevant biomarkers of disease progression towards respiratory insufficiency.

This paper is organized as follows: Section 2 provides background on ALS and Pattern Mining together and discusses the related work; Section 3 proposes the methodology for learning prognostic models using disease progression patterns, explaining data transformation, pattern extraction and evaluation, as well as classification; and Section 4 presents and discusses the results obtained for the predictive models, as well as pattern evaluation. Finally, Section 5 provides concluding remarks and discusses future work.

2 Background

2.1 Amyotrophic Lateral Sclerosis

Amyotrophic Lateral Sclerosis (ALS) is a neurodegenerative disease characterized by progressive degeneration of lower and upper motor neurons, at the spinal and bulbar level [3]. It spreads from an initial focal weakness, eventually affecting skeletal muscles, including the diaphragm. In this context, respiratory failure is a common cause of death among ALS patients, within 3-5 years of symptom onset. The majority of patients present a spinal onset, with common initial symptoms including weakness and fasciculations of the limbs. Difficulty with word articulation (dysarthria) is a common symptom linked to a bulbar onset, which represents about a third of all cases [4], [5]. Respiratory onset typically presents with specific initial symptoms, such as dyspnea, orthopnea and sleeping problems. However, it makes up less than 3 percent of all cases, with respiratory impairment being considered an end-stage event in disease progression [5], [6], [7]. Up to 50 percent of ALS patients also develop mild cognitive impairment, such as memory loss and behaviour changes [8].

Recent studies reported an incidence of 8-9 ALS cases per 100,000 inhabitants worldwide [41], and 10 in 100,000 in Portugal [40]. Peak onset age is of 58-63 years in the case of sporadic ALS. In the familial disease, disease onset typically occurs 5-10 years earlier [9], [10]. ALS has been linked to mutations in several genes, such as *SOD1* and *C9orf72*. Even though only about 10 percent of patients have a familiar history of ALS, gene mutations associated with familial ALS are seen in 5-10 percent of cases labelled as sporadic [11], [12], [13].

ALS has no cure, and most treatments focus on symptom management. The drug Riluzole has shown to increase life expectancy by up to 14.8 months [4], [13]. Edaravone, not approved in the EU, has a possible mild effect in slowing down functional decay in a very restricted number of patients [14]. Non-invasive ventilation (NIV) has shown an increased survival time when compared to Riluzole, mainly in patients with normal to mildly impaired bulbar function. Despite lower feasibility in patients with bulbar onset, a large cohort study suggested NIV also increases life expectancy in these cases [13], [15]. As such, NIV is advisable to most patients at the onset of respiratory symptoms [16]. However, current guidelines do not prescribe NIV access to all patients readily. Thus, several efforts have been made to determine the best time to initiate NIV, and the most relevant evaluation measures to determine eligibility [5].

2.2 Pattern Mining

In this work, we use Itemset Mining together with Sequential Pattern Mining [42] to discover both disease presentation patterns and disease progression patterns, from static and temporal data, respectively. These patterns are then used to extract new features for classification. In order to construct training sets to learn our prognostic models, where frequent patterns are used as features, we used pattern mining algorithms to find static/sequential patterns and output the id's of the transactions/sequences (patients) where these patterns occurred. To this aim, AprioriTID-Close [17], [18] was used to retrieve closed patterns from the static features, while Fournier08 [20] was used to retrieve closed sequential patterns from the temporal features. The latter was chosen over other algorithms, such as PrefixSpan [19], as it takes the time stamps of each transaction into account, rather than just item occurrence order, hence yielding more comprehensible patterns. Closed patterns were chosen to summarize the patterns, preventing the extraction of large amounts of overlapping patterns.

Follows a brief introduction on the used pattern and sequential pattern mining algorithms, which might be skipped according to reader's background. All pattern mining algorithms used are available in the SPMF library [43].

2.2.1 Itemset Mining

Itemset Mining, or Pattern Mining (PM), is a set of unsupervised learning techniques meant to find objects of interest that occur together within a transaction database [21], where each tuple corresponds to one transaction, containing an identifier and an itemset. A k-itemset is a subset of size k of the collection of existing items. In our case, each item in the collection corresponds to a pair *(feature, value)*, such as

"Gender = Male". As such, each transaction consists of a patient identifier and an itemset that contains items with the information contained in all (non-missing) static features for the patient. We focus on closed itemsets: frequent patterns that occur with a frequency (or support) above a predefined threshold and are not contained within supersets with the same support. This minimizes the risk of extracting a large number of patterns, many of which subsets of each other. There are many algorithms that compute these type of itemsets, using different candidate generation methods and transaction database traversal, resulting in varying levels of efficency and speed [21]. In this work, we use the implementations of AprioriTID algorithms [17], [18] available in the SPMF library to compute both frequent and closed itemsets.

AprioriTID [17] uses a level-wise search of the itemsets, using the set of frequent k-itemsets to compute the frequent $k+1$-itemsets. It makes use of the Apriori property to prune candidate itemsets at each level. This property is based on two observations: 1) If a k-itemset is frequent, then all of its subsets must also be frequent; and 2) If a k-itemset is not frequent, then none of its supersets can be frequent [21]. In practice, this means that for a certain candidate $k+1$-itemset, support is only computed if all of its k-subsets are frequent. AprioriTIDClose [18] discovers the closed frequent itemsets first, using a closure mechanism based on the Galois connection. According to its original proposal, the frequent itemsets can then be extracted from the closed ones. As such, SPMF implementation outputs only the closed frequent itemsets.

2.2.2 Sequential Pattern Mining

Sequential Pattern Mining (SPM) refers to a set of techniques meant to find interesting patterns spanning several time points or object positions in a dataset [21], [42]. It is an extension of Itemset Mining, in the sense that each sequence corresponds to an identifier and a set of transactions performed by the same entity over different time periods (one per transaction). The time point may or not be specified. In this work, each sequence is associated to a patient, with each transaction being an evaluation performed during the patient's follow-up and each item a pair *(exam, value)*.

There are several algorithms to perform SPM [42], with varying degrees of efficiency and complexity. As with Itemset Mining, different algorithms can retrieve closed sequential patterns, and some algorithms can further take time constraints into account, allowing the retrieval of sequences with specific lengths or gaps between time points [20].

PrefixSpan [19] considers item occurrence order, but not time intervals between transactions. This means two sequences where the same itemset occur at different time points are considered to have the same pattern [22].

Hirate-Yamana's algorithm [22] takes into account the transaction timestamps and allows for sequential pattern retrieval with specific time intervals between transactions. It is based on PrefixSpan, functioning in a similar manner. The suffix of each item occurrence is still kept at each projected database. However, the respective timestamp changes to translate the new time intervals between transactions. There are also four constraints that when specified, each sequence must satisfy, apart from minimum support. In the context of this work, only one is relevant: $t_{i,i+1} \leq max\_interval \forall i \in \{1, m-1\}$, where $max\_interval$ is the maximum time interval allowed between two adjacent itemsets in a sequence. It will be referred to as *temporal gap* or *allowed time interval*.

The Fournier08 algorithm [20] is an extension of Hirate-Yamana, but has integrated the BIDE+ algorithm closure checking [23]. This feature allows for a closed sequence set to be maintained without a closed candidate set, and therefore, the complete frequent sequence set does not have to be explored. It also implements the BackScan pruning from the BIDE+ algorithm [23].

2.3 Related Work

The pressing need to find effective treatments to neurodegenerative diseases, such as ALS, combined with the complex nature of these diseases, lead to development of several computational approaches aimed to assist clinicians in diagnosis, prognosis and decision making. Myszczynska *et al.* [24] reviewed how Machine Learning algorithms can be applied to available biomedical data in order to build insightful strategies, highlighting the heterogeneity among patients and complexity of disease mechanisms as main challenges. Grollemund *et al.* [25] surveyed the approaches applied to ALS datasets, emphasizing the challenges, such as small sample sizes and disease heterogeneity, described the achievements and discussed future work.

In the particular case of the Portuguese ALS Dataset, Carreiro *et al.* [30] pioneered the use of prognostic models to predict the need for NIV based on clinically defined time windows. They further explored the use of SPM to extract features encoding temporal dependencies [26]. Standard classifiers were used to predict the need of NIV within k days of the last evaluation, using both a fixed and variable number of time points. Dataset size and a limited number of time points per patient impacted negatively the results. Preliminary results can be found here [27]. Pires [28] also used the original features for prognostic prediction and explored the use of several evaluations per patient. However, the temporal features were always considered independent by the models used, which were trained with the same number of evaluations for all patients. Predictive models for one, two and three evaluations were trained. The predictive models using the last evaluation always outperformed the others. Following this work, Pires *et al.* [29], [36] explored different approaches for patient stratification, showing that learning group specific prognostic models can in fact improve predictability, but no temporal data were used.

More recently, Matos *et al.* [32] proposed a biclustering-based classifier. Biclustering [35] was used to find groups of patients with coherent values in subsets of clinical features (biclusters), then used as features together with static data. Despite the interesting results, only presentation patterns discovered in static data were used as features and no temporal data were considered. Soares *et al.* [33] proposed a triclustering-based classifier to analyse temporal data. Triclustering [34] was used to find groups of patients with coherent values in subsets of clinical features in a subset of evaluations (triclusters). The triclusters are then used as

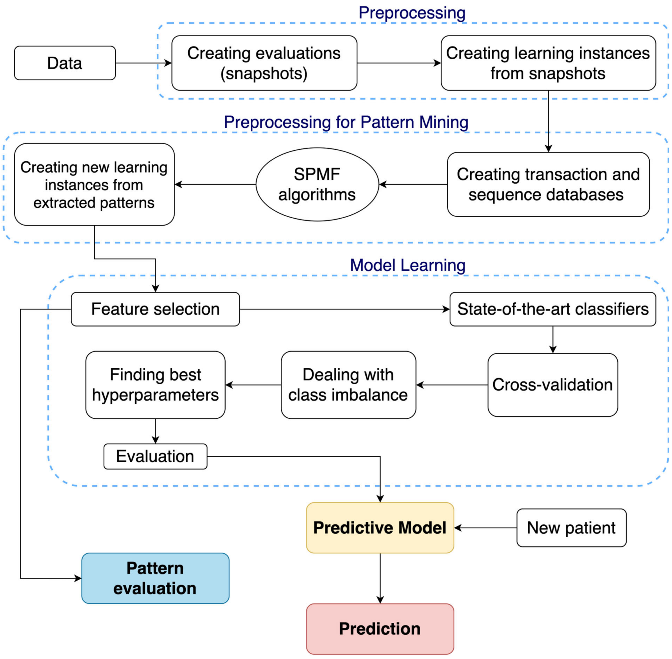

Preprocessing
Data
Creating evaluations (snapshots)
Creating learning instances from snapshots
Preprocessing for Pattern Mining
Creating new learning instances from extracted patterns
SPMF algorithms
Creating transaction and sequence databases
Model Learning
Feature selection
State-of-the-art classifiers
Finding best hyperparameters
Dealing with class imbalance
Cross-validation
Evaluation
Predictive Model
New patient
Pattern evaluation
Prediction

Fig. 2. Workflow of the methodology to learn prognostic models using using disease progression patterns. After the model is learnt a prediction can be made for a new patient providing values for the original features.

binary features. The results are promising, showing the benefits of using patterns for the sake of interpretability. However, the approach can only be trained using the same (predefined) number of evaluations for all patients and static features are not incorporated in the prognostic model. The tricluster-based patterns identify temporal patterns where the same set of features have coherent values in a set of temporal evaluations, whereas the sequential patterns used in this work can unveil patterns containing different sets of features in each temporal evaluation.

3 Methods

This section describes the methodology proposed for prognostic prediction using both disease presentation patterns and disease progression patterns, discovered in static and temporal data, using itemset mining and sequential pattern mining, respectively. We first provide a brief description of the Portuguese ALS dataset used in this work highlighting its challenges. Then, we describe the main steps of the proposed methodology to learn prognostic models using disease progression patterns: 1) Data preprocessing; 2) Preprocessing data for pattern mining; 3) Training set creation and model learning; and 4) Pattern evaluation. Fig. 2 presents the methodology workflow, highlighting the key steps, discussed below with the help of a toy example.

3.1 About the Portuguese ALS Dataset

The dataset used in this work is the Portuguese ALS dataset, containing demographic information and clinical data, such as respiratory and functional test results, for a cohort of 1,375 patients, followed at the ALS clinic of the Translational Clinic Physiology Unit, Hospital de Santa Maria, IMM, Lisbon. Observations range from 1995 until March 2020 and consist of 5 tables, one containing 66 static features, and 4 containing a total of 103 temporal features (including dates). Since some features have dependencies on others, are highly correlated or have a high percentage of missing values, only 21 features are kept after preprocessing. Previous versions of this dataset with less patients and time points were analysed by Carreiro *et al.* [26], [30] and Pires *et al.* [28].

Data is in a 1-row-per-patient format, which means the temporal features span several columns. Also, the tests in one evaluation set are not usually performed within the same day, or may not be performed at all (due to the patient's condition, scheduling or costs). Hence, there is not a clear distinction between consecutive evaluations, but rather pairs *(date,test)*. This means the dataset must be transformed in order to make use of the temporal features, by grouping exam sets by date. This way, each evaluation will consist of a set of tests performed during a relatively small time interval. To build predictive models, a class stating if each patient did actually need NIV within k days of the last appointment must also be created, since the dataset only states the NIV status and/or NIV initiation date. This class was named *Evolution*. The transformation into evaluation sets would not be enough if using the original features, since each patient has now several corresponding rows, one per evaluation. A certain number of time points would have to be selected and data rearranged to a 1-row-per-patient format. Due to the use of sequential patterns as new features, all time points are included, but extra transformation stages are still required, such as the creation of sequence databases for pattern mining and subsequent training sets. These steps will have the side effect of only retaining the Evolution class' last value for each patient, most likely leading to class imbalance. It is also expected that the number of features grows when comparing to the original set, which can impair classifier performance.

The data preprocessing required can be divided into two stages: 1) preprocessing, where the original data is transformed into several datasets, each for a specified prediction window, and 2) preprocessing for Pattern Mining, where these datasets are used for pattern extraction and generation of new learning instances. Finally, predictive models are built and evaluated, as shown in the workflow in Fig. 2 showing all these key steps.

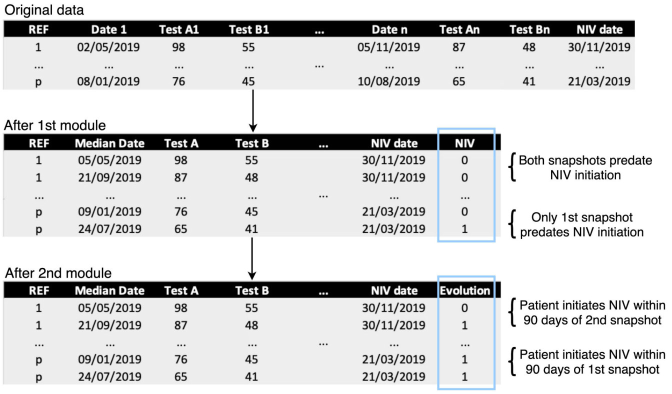
3.2 Data Preprocessing

Data transformation is performed as described by Carreiro *et al.* [30] and Pires *et al.* [28], and consists of two modules, exemplified in the sections below using a toy example.

The first module (Creating Snapshots) transforms the original data into a set of snapshots (1 per row) by selecting the desired features and grouping exams taken in close dates. The second module (Creating Learning Instances) takes the snapshot set as input and creates a set of learning instances, according to the selected prediction window, k: 90, 180 or 365 days. It also generates the class, *Evolution*, for each instance (patient). An example of the transformations performed at this stage are presented in Fig. 3.

3.2.1 *Creating Snapshots*

The first module takes as input the original datasets, one per feature group. It transforms them into a set of snapshots (1



Original data

REF	Date 1	Test A1	Test B1	...	Date n	Test An	Test Bn	NIV date
1	02/05/2019	98	55		05/11/2019	87	48	30/11/2019
...	...	...	...	...	...	...	...	...
p	08/01/2019	76	45		10/08/2019	65	41	21/03/2019

After 1st module

REF	Median Date	Test A	Test B	...	NIV date	NIV
1	05/05/2019	98	55		30/11/2019	0
1	21/09/2019	87	48		30/11/2019	0
...	...	...	...	...	...	...
p	09/01/2019	76	45		21/03/2019	0
p	24/07/2019	65	41		21/03/2019	1

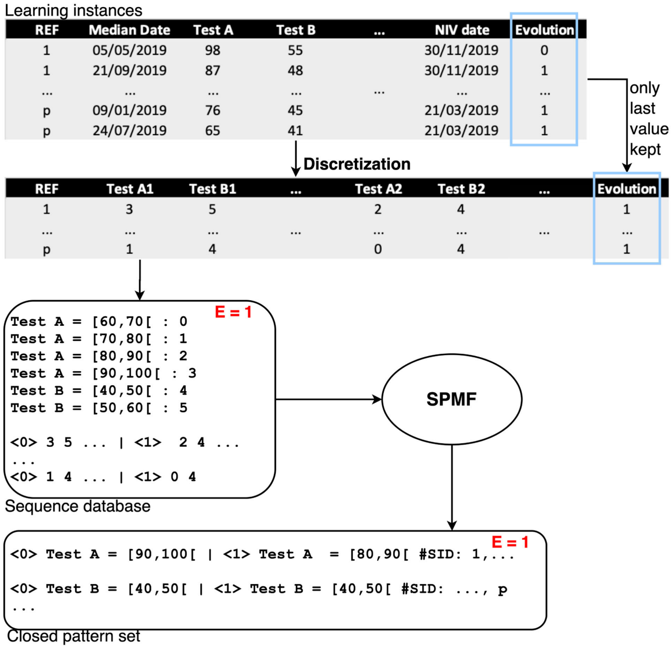
Both snapshots predate NIV initiation
Only 1st snapshot predates NIV initiation

After 2nd module

REF	Median Date	Test A	Test B	...	NIV date	Evolution
1	05/05/2019	98	55		30/11/2019	0
1	21/09/2019	87	48		30/11/2019	1
...	...	...	...	...	...	...
p	09/01/2019	76	45		21/03/2019	1
p	24/07/2019	65	41		21/03/2019	1

Patient initiates NIV within 90 days of 2nd snapshot
Patient initiates NIV within 90 days of 1st snapshot

Fig. 3. Example of data transformations performed while preprocessing the Portuguese ALS dataset. The resulting dataset contains class-labelled snapshots, that is, learning instances, where each patient is characterized by a set of features and the Evolution class. Note that the last snapshot is not eligible as learning instance (evolution is already 1 in the previous instance) and is only kept for the sake of comprehension.



Learning instances

REF	Median Date	Test A	Test B	...	NIV date	Evolution
1	05/05/2019	98	55		30/11/2019	0
1	21/09/2019	87	48		30/11/2019	1
...	...	...	...	...	...	...
p	09/01/2019	76	45		21/03/2019	1
p	24/07/2019	65	41		21/03/2019	1

REF	Test A1	Test B1	...	Test A2	Test B2	...	Evolution
1	3	5		2	4		1
...	...	...	...	...	...	...	...
p	1	4		0	4		1

```
Test A = [60,70[ : 0
Test A = [70,80[ : 1
Test A = [80,90[ : 2
Test A = [90,100[ : 3
Test B = [40,50[ : 4
Test B = [50,60[ : 5

<0> 3 5 ... | <1>  2 4 ...
...
<0> 1 4 ... | <1> 0 4
```

```
<0> Test A = [90,100[ | <1> Test A  = [80,90[ #SID: 1,...

<0> Test B = [40,50[ | <1> Test B = [40,50[ #SID: ..., p
...
```

Fig. 4. Example of the data transformations performed during preprocessing for pattern mining and pattern extraction.

per row) by selecting the desired features and grouping exams performed in close dates. Since each test result is associated with a date, all exams referring to a patient are sorted by date and grouped using an Agglomerative Hierarchical Clustering approach with constraints [30]. Two constraints are taken into account: 1) each exam can only occur once in a snapshot; and 2) all exams grouped within the same snapshot must have occurred in dates where the NIV status is the same. That is, exams performed before and after NIV initiation cannot belong to the same snapshot.

The output of this module is a set of snapshots. Each snapshot contains one column per exam, a NIV status (1 if NIV was initiated, 0 if not), a NIV initiation date, if available, the first and last dates featured in the evaluation, as well as the median date. The median date is considered as the snapshot's date, and will be used in subsequent stages. Note that, since each patient typically performs an exam several times, most patients will have more than one shapshot associated to them. The set of snaphots computed for each patient are used as the set of evaluations in the follow-up.

3.2.2 *Creating Learning Instances*

The second module receives the snapshot set as input, determines which snapshots are eligible to be used as learning instances and generates the *Evolution* class by comparing NIV statuses between snapshots. Since the *Evolution* class states whether or not a patient required NIV within k days from the last evaluation, each learning instance to be created requires the knowledge of NIV status from two snapshots, and the length of the prediction window, k. Following previous work with the Portuguese ALS dataset, the time windows chosen are 90, 180 or 365 days [26], [28], [30], targeting short, medium and long term predictions. Evolution occurs when a patient's NIV status switches from 0 to 1 within the specified time window. Assuming two snapshots are n days apart: if $n < k$, the patient evolves if the starting NIV status is 0 and finishing status is 1 (E=1); if both status are 0, there is no Evolution (E=0); and if $n > k$, the patient also does not evolve (E=0), in the predefined time window.

As previously mentioned, only eligible snapshots can be used to generate learning instances. If the patient has already initiated NIV at a given snapshot, it is not necessary to explore the following snapshots: the patient has already evolved to require NIV. If there is no information on NIV status before the selected time window, there is no use to keep the snapshot as well: we cannot know if the patient evolved or not. Note that a learning instance will consist of the starting snapshot, if eligible, and the newly generated class obtained using NIV information from future snapshots. Therefore, a snapshot will not be eligible if: 1) the starting NIV status is 1; or 2) the starting NIV status is 0, but unknown after k days.

3.3 Preprocessing for Pattern Mining

After the initial preprocessing is completed we are left with a set of learning instances for each selected time window, with a learning instance comprising a snapshot and corresponding class label. However, using these instances for classification would correspond to using only one evaluation for prognosis prediction. Instead, our goal is to make these predictions based on the patients' complete evaluation history, and moreso, to use patterns found in data for the purpose of prediction. This means that additional data processing steps have to be made. First, we have to transform the sets of learning instances into sequence and transaction databases (for temporal and static features, respectively), and then perform pattern extraction, with the SPMF library. Fig. 4 shows an example of the transformation steps performed at this stage. After pattern mining, new datasets are generated and used to train/evaluate different models.

3.3.1 *Generating Input Files for Pattern Extraction*

As mentioned before, the learning instances generated through the preprocessing modules consist of class-labelled snapshots. This means that most patients will be associated with a set of snapshots, which in fact correspond to a set of transactions within a sequence. Therefore, our goal in this

TABLE 1
Summary of the Features and Values in the Portuguese ALS Dataset That Were Used for Itemset and Sequential Pattern Mining

Type	Feature	Values (post-discretization)
Static	Gender	Male, Female
	Body Mass Index at first visit (kg/m^2): BMI	$<20,\ [20,25[,\ [25,30[,\ \geq 30$
	Familiar history of motor neuron disease: MND familiar history	Yes, No, Unknown
	Age at onset (years)	$<30,\ [30,50[,\ [50,70[,\ \geq 70$
	Disease duration (months)	$\leq 6,\ ]6,12],\ ]12,18],\ ]18,36],\ >36$
	El Escorial Reviewed Criteria	definitive, probable, possible, suspected
	Predominantly affected motor neurons: UMN vs LMN	UMN, LMN, both
	Onset form	spinal, bulbar, respiratory/axial, mixed, FTD
	Hexanucleotide repeat expansion in C9orf72 gene: C9orf72	Yes, No, Unknown
Temporal	Functional Rating Scale: ALS-FRS	$<12,\ [12,24[,\ [24,36[,\ \geq 36$
	Bulbar subscore: ALS-FRSb	$<4,\ [4,8[,\ [8,12[,\ 12$
	Upper limb subscore: ALS-FRSsUL	
	Lower limb subscore: ALS-FRSsLL	
	Revised respiratory subscore: R	
	Maximal Sniff Nasal Inspiratory Pressure: SNIP (cmH2O)	$<40,\ [40,60[,\ \geq 60$
	Maximal Inspiratory Pressure: MIP(%)	
	Forced Vital Capacity: FVC (%)	$<40,\ [40,60[,\ [60,80[,\ \geq 80$
	Maximal Expiratory Pressure: MEP (%)	
	Phrenic Nerve Response Amplitude: PhrenMeanAmpl (mV)	$<0.4,\ \geq 0.4$
	Cervical Flexion: CervicalFlex	$<5,\ 5$
	Cervical Extension: CervicalExt	

step is to transform our current sets of learning instances into sequence databases, with one sequence per patient.

Since Pattern Mining algorithms require discrete data, as transactions are made of items, it is necessary to discretize continuous features, as well as attributes with many possible values. The original features were separated into two groups: static and temporal. Continuous features in both groups were discretized, and some categorical features had their values regrouped, according to studies in Matos [31] and feedback from clinicians. A total of 21 features were selected (apart from the patient identifier and class), originating 33 static and 40 temporal items of the form *Attribute = value*, such as *Gender = Female* or *ALS-FRS = [24,36[*. Table 1 shows the complete set of features and values used.

For the static features, all rows pertaining the same patient have the same information. Hence, only the first of these rows was kept. Items were generated for the features with a known value, meaning there is no need for missing value imputation. Input files were created for each of the prediction windows and *Evaluation* class values, with items identified by an integer key (as required by SPMF) and a caption added at the top of each file. Temporal data were reshaped into a 1-row-per-patient format, with a delimiter character separating each transaction, processed similarly to the ones described in the static case. The input files were generated as for the static case for all time windows.

3.3.2 Pattern Extraction in SPMF

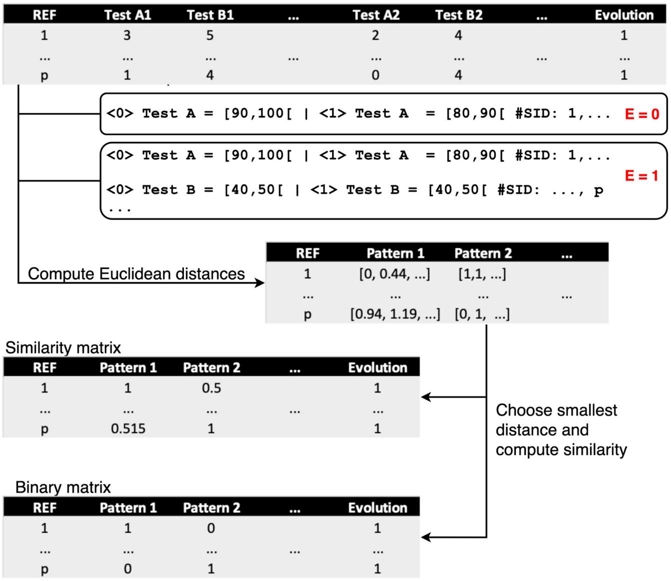
The generated input files are fed to the SPMF library, which outputs the extracted patterns after selecting the corresponding algorithm and parameters. AprioriTIDClose [18] was used for Itemset Mining, and Fournier08 [20] for Sequential Pattern Mining. Patterns were retrieved separately for each class value to prevent potential under-representations caused by class imbalance further on. The class label corresponds to the *Evolution* value of each patient's final snapshot. The support threshold considered was 25 percent, as it provided a fitting balance between the number of patients required for a pattern to emerge and the total amount of patterns retrieved. Maximum time interval allowed was set to 10. Each output file provides the list of eligible patterns, along with their absolute support and the list of sequences where each pattern is verified. It is important to note that the same pattern can be found in sequences at different starting points, thus the timestamps presented in the patterns are relative.

3.4 Training Set Creation and Model Learning

The SPMF output files are then used to build both binary and similarity matrices, to which the class is added, leading to new datasets to be used by the classifiers. This is performed by computing information about the emergent patterns, such as support, number of time points, number of items, class where they were obtained and list of patients that verifies them. The trivial sequences and itemsets, that is, the ones containing only one item, are discarded. The remaining patterns are then compared to the original sequence databases to compute the matrices, where each row correspond to a patient and each column to a pattern. This is performed for each pattern as follows: if the patient verifies, that is, if the patient values are all equal to the pattern values, then the corresponding element is 1; if not, then an array of distances is computed, one for each starting time point t in the patient sequence. The distance metric used was euclidean distance between patient P and pattern Q:

$$d_t(P,Q) = \sqrt{\sum_{\tau\in\mathcal{T}}\sum_{f\in\mathcal{F}}(p_{t+\tau,f} - q_{\tau,f})^2}, \tag{1}$$

where $\mathcal{T}$ is the set of time points featured in the pattern and $\mathcal{F}$ the set of features at each time point τ. Hence, the elements of P and Q are the previously discretized attribute



REF	Test A1	Test B1	...	Test A2	Test B2	...	Evolution
1	3	5		2	4		1
...	...	...	...	...	...	...	...
p	1	4		0	4		1

<0> Test A = [90,100[| <1> Test A = [80,90[#SID: 1,... E = 0
<0> Test A = [90,100[| <1> Test A = [80,90[#SID: 1,...
<0> Test B = [40,50[| <1> Test B = [40,50[#SID: ..., p
...
E = 1
Compute Euclidean distances

REF	Pattern 1	Pattern 2	...
1	[0, 0.44, ...]	[1,1, ...]	
...	...	...	...
p	[0.94, 1.19, ...]	[0, 1, ...]	

Choose smallest distance and compute similarity

Similarity matrix

REF	Pattern 1	Pattern 2	...	Evolution
1	1	0.5		1
...	...	...	...	...
p	0.515	1		1

Binary matrix

REF	Pattern 1	Pattern 2	...	Evolution
1	1	0		1
...	...	...	...	...
p	0	1		1

Fig. 5. Example of the data transformations performed to obtain the training sets.

values, now normalized between 0 and 1. Since many sequences won't contain values for all of the pattern's features, d_t is only computed if the patient has at least one feature value at each of the pattern's time points, with the remainder of the individual distances being replaced by the maximum possible value

$$p_{t+\tau,f} = nan \rightarrow (p_{t+\tau,f} - q_{\tau,f}) = \max\{|0 - q_{\tau,f}|,\ |1 - q_{\tau,f}|\}. \tag{2}$$

Otherwise, $d_t = \infty$. The smallest distance is then chosen and used to compute a similarity value between 1 and 0

$$s = \frac{1}{1 + \min d_t}. \tag{3}$$

Static patterns are treated similarly but only one distance needs to be computed. This value is added to the corresponding similarity matrix element. If $s = 1$, a 1 is also added to the corresponding value in the binary matrix, filled with 0s by default. Finally, the *Evolution* value in the patient's final snapshot is added as class label. Fig. 5 shows an example of the described transformations.

These datasets are then used to train and evaluate several prognostic models, using the state-of-the-art classifiers Random Forests and XGBoost, as well as Naive Bayes and Support Vector Machines. Model evaluation was performed using Sensitivity, Specificity and AUC scores, showing their ability in positive and negative examples, and in overall classification, respectively. Accuracy is also presented.

As previously mentioned, we can expect class imbalance and a large number of features as consequences of the approach. As we only retain the class label of the last evaluation, the likelihood of a patient evolving to need NIV grows as we consider larger prediction windows. Hence, progressively disproportionate class distributions are expected. Moreso, a lengthier medical history is required to compute the class, resulting in a decreasing amount of learning instances. Therefore, a class balancing strategy should be implemented. In this work, Random Undersampling and SMOTE [2] were used. Also due to these reasons, the number of features can differ greatly between prediction windows. Since the support threshold used for pattern mining was fixed at 25 percent, less examples are required for a pattern to emerge as the prediction window grows. This prompts an increase in the amount of extracted patterns, and many of which may not be useful for classification. To address this issue, a feature selection step can be implemented. In this work, a Decision Tree was used to prune low-importance features and compute a feature rank.

3.5 Pattern Evaluation

In order to assess the relevance of each pattern towards classification, an extra step is carried out, where patterns are evaluated based on their support for both classes. For each prediction window, and considering the matching binary matrix, we compute, for each pattern P, the relative support for the positive $(Evolution = 1)$ and negative class $(Evolution = 0)$. Then, we compute Growth Rate, measuring how support grows from class A to class B as follows:

$$GrowthRate_{A\rightarrow B}(P) = \begin{cases} \dfrac{supp_B(P)}{supp_A(P)}, & supp_A(P) \neq 0 \\ 0, & supp_A(P) = 0 = supp_B(P) \\ \infty, & supp_A(P) = 0 \neq supp_B(P). \end{cases}. \tag{4}$$

For a given threshold α, we consider that pattern P is discriminative for the positive class if

$$GrowthRate_{pos}(P) = GrowthRate_{neg\rightarrow pos}(P) > \alpha, \tag{5}$$

and discriminative for the negative class if

$$GrowthRate_{neg}(P) = GrowthRate_{pos\rightarrow neg}(P) > \alpha. \tag{6}$$

The higher the Growth Rate, the more discriminative the pattern is, for one of the classes. If both rates are 1, the support is the same in both classes, and thus the pattern is not discriminative.

4 Results and Discussion

As case study we used the proposed methodology to learn prognostic models in the Portuguese ALS dataset, described in Section 3.1. We target the learning of prognostic models using disease progression patterns in order to predict the need for NIV in ALS using short, medium and long time windows. This section presents and discusses our results from two perspectives: model evaluation and pattern evaluation, promoting not only a discussion from the point of view of model effectiveness regarding prognostic prediction, but further highlighting the importance of pattern evaluation for the sake of model interpretability.

4.1 Model Evaluation

Targeting prognostic prediction, several models were evaluated in order to explore the settings under which the proposed approach can be of most value. We established a set of benchmarks to study how it fares without the presence of

TABLE 2
Information Regarding the Datasets Generated to Learn the Benchmark Models for Each Prediction Window, Using a Minimum Support of 25 percent per Class in SPMF Algorithms

	No. Patients	No. Features		Class distribution
		Original	Patterns	

90d	940	21
		9 S + 12 T
			44
			40 S + 4 T
				Y: 551 (58.62%) {110}
				N: 389 (41.38%) {78}

180d	895	21
		9 S + 12 T
			50
			43 S + 7 T
				Y: 637 (71.17%) {127}
				N: 258 (28.83%) {52}

365d	840	21
		9 S + 12 T
			56
			48 S + 8 T
				Y: 654 (77.86%) {130}
				N: 186 (22.14%) {38}

The number of patients and patterns correspond to the number of learning instances and features, respectively. In all cases the class is Evolution. *The numbers in {} brackets are the number of tuples on the test folds.*

temporal data, in comparison to the use of original features. Then, we implemented the approach using the patients' complete evaluation history and evaluated its performance when compared to the benchmarks.

4.1.1 Benchmarks

In order to assess whether the proposed approach described in Section 3 is beneficial in comparison to those not using temporal data, we established two benchmarks: Benchmark 1) classification using the original data from the last available time point, and Benchmark 2) classification using only patterns found in the last time point. This was carried out by selecting the last available snapshot from the datasets outputted in Section 3.2, for each patient.

Benchmark 1 was performed using both original and post-discretization values. Since missing value imputation was required we used the mean and mode values for numerical and categorical features, respectively.

Benchmark 2 consisted in applying the proposed approach only to the last (most recent) snapshots. From the latter, result static patterns that remain unchanged from those mined with the complete dataset, as well as a reduced set of patterns mined from temporal features.

Table 2 shows information regarding the datasets used to learn Benchmark 1 and 2. It is worth noting that the features we call *temporal* in these datasets, while comprised of original features whose values evolve through time, only pertain to the values found in the last time point.

As expected, the number of eligible patients decreases as the time window increases, as a lengthier medical history is required to compute the class label. Moreso, the class distribution becomes more disproportionate as the prediction window grows. This makes sense, since it is more likely for a patient to evolve within a year than within a few months. The imbalance is further magnified by the fact that only the class from the last evaluation is retained, meaning that a large portion of the negative class will comprise of patients still in treatment. To tackle this issue, a combination of Random Undersampling and SMOTE was performed. First, majority class instances were removed until the desired ratio was achieved, followed by the generation of minority class examples through SMOTE. Several ratios were explored, aiming for performance improvement with minimal loss of information. The ratios chosen were 45/55, 35/65 and 30/70 percent for 90, 180 and 365 days, respectively.

TABLE 3
Parameter Values Tested Through Grid Search for Each Classifier

Classifier	Parameter	Tested values
Random Forest	Max. features	{1, 2, 5, 10, 20}
	Nr. estimators	{100, 200, 500, 1000}
SVM Gaussian	C (complexity)	{0.01, 0.1, 1, 10, 100}
	Gamma	{0.01, 0.1, 1, 10, 100}
XGBoot	Learning rate	{0.1, 0.2, 0.3, 0.4, 0.5}
	Nr. estimators	{100, 200, 500, 1000}

We trained and evaluated four models using Random Forests, SVMs with Gaussian kernel, XGBoost and Naive Bayes, using stratified 5 × 5-fold cross-validation (CV) and a Grid Search to determine the best hyperparameters for each classifier. Table 3 shows the parameter tested.

Tables 4 and 5 present the benchmark results. Some benchmark approaches have trouble at classifying minority class examples, mainly SVMs using original feature values. This worsens for mid and long-term predictions, which was expected given the original class imbalance. Using the original features results in higher sensitivities and lower specificities, while using the patterns as features yields slightly more balanced outputs. When comparing between classifiers, we see that Random Forests and XGBoost present consistent results throughout, while SVMs and Naive Bayes show more variability. There are some significant differences in AUC scores. Wilcoxon Signed-Ranks tests yielded p-values of 0.068 (with mostly negative ranks) when comparing the AUC's obtained by the binary and similarity matrices to those outputted by the models learnt with with discrete values. Hence, the models trained with the latter perform significantly better, for typical significance levels. However, when comparing the AUC's obtained by the matrices to those outputted by the models learnt with original values, the tests yielded p-values above 0.715, meaning there are no significant differences between the results obtained through these datasets. Still, these matrices contain patterns mostly containing static feature items, so we expect that the addition of temporal data will improve results.

4.1.2 Binary and Similarity Matrices

Datasets were constructed using the binary and similarity matrices built from the Portuguese ALS dataset for three prediction windows (90, 180 and 365 days), using the proposed methodology described in Section 3. Their amount of learning instances and class distributions, showned in Table 6, are identical to those in the benchmark datasets. Hence, we continue to use Random Undersampling and SMOTE as sampling strategy. Random Undersampling is performed until the prediction-window specific ratio is met (45/55, 35/65 and 30/70 percent for 90, 180 and 365 days, respectively) and then minority class tuples are generated through SMOTE until a balanced training set is obtained.

A decreasing amount of learning instances, paired with a growing class imbalance, also results in an increasing

TABLE 4
Model Evaluation Results Obtained Using Class-Labelled Patient Snapshots as Training Sets, Built From the Last Available Time Points and Over a Stratified 5x5-Fold Cross-Validation Scheme (Benchmark 1)

	Clf	Sensitivity	Specificity	AUC	Accuracy
			Original values		
90d	RF	68.7 ± 3.9	59.8 ± 6.1	**70.2 ± 3.6**	65.0 ± 3.6
	SVM	92.8 ± 2.1	8.8 ± 3.3	56.1 ± 2.7	58.0 ± 2.0
	XGB	68.7 ± 4.5	58.8 ± 7.1	69.0 ± 3.5	64.6 ± 3.4
	NB	67.8 ± 3.9	51.5 ± 9.4	63.9 ± 4.1	61.1 ± 3.5
180d	RF	78.3 ± 3.7	56.1 ± 6.1	**74.6 ± 2.7**	71.9 ± 2.2
	SVM	95.1 ± 1.5	4.1 ± 2.8	54.3 ± 5.1	68.9 ± 1.1
	XGB	75.7 ± 2.9	57.5 ± 6.3	72.7 ± 3.3	70.5 ± 2.4
	NB	60.8 ± 4.0	66.8 ± 7.1	67.4 ± 4.3	62.5 ± 3.0
365d	RF	83.2 ± 2.2	51.5 ± 9.3	**75.4 ± 4.1**	76.1 ± 2.3
	SVM	97.4 ± 1.3	4.2 ± 2.4	61.9 ± 4.4	76.7 ± 1.3
	XGB	78.8± 2.2	53.3 ± 8.1	73.3 ± 4.0	73.1 ± 2.0
	NB	57.5 ± 6.2	68.5 ± 8.5	68.5 ± 5.3	59.9 ± 4.7
Discretized values					
90d	RF	66.3 ± 4.3	56.7 ± 7.3	67.0 ± 4.7	62.3 ± 3.6
	SVM	59.6 ± 3.8	68.0 ± 6.1	**68.9 ± 3.8**	63.1 ± 3.5
	XGB	64.2 ± 5.8	59.0 ± 5.8	66.7 ± 4.6	62.0 ± 3.7
	NB	53.5 ± 4.0	73.1 ± 6.8	68.1 ± 4.0	61.6 ± 3.4
180d	RF	76.6 ± 4.2	55.8 ± 6.9	73.1 ± 3.9	70.6 ± 3.2
	SVM	66.3 ± 4.6	68.7 ± 7.2	**74.5 ± 3.2**	66.9 ± 3.1
	XGB	73.3 ± 3.9	58.6 ± 6.7	73.1 ± 3.3	69.1 ± 3.0
	NB	60.5 ± 6.9	71.9 ± 7.6	72.4 ± 4.6	63.8 ± 4.8
365d	RF	80.9 ± 2.9	52.1 ± 8.7	**74.8 ± 4.8**	74.5 ± 2.8
	SVM	66.7 ± 3.8	71.2 ± 8.1	**74.8 ± 4.0**	67.7 ± 2.8
	XGB	77.5 ± 3.1	55.3 ± 10.3	73.6 ± 4.3	72.5 ± 3.2
	NB	52.4 ± 8.9	78.9 ± 8.2	74.0 ± 5.0	58.3 ± 6.3

We present the average and standard deviation over all folds. The best AUC for each dataset appears in bold.

amount of emerging patterns as we broaden the prediction window, given the fixed support threshold. As we do not expect all patterns to be relevant for classification (a large amount of features could actually impair classifier performance rather than improve it), it makes sense to prune some of these patterns. Consequently, the feature importances of a Decision Tree were used for the purpose of feature selection. Several thresholds were tested for each prediction window, with the best performing ones being chosen: 0.01, 0.005 and 0.001 for 90, 180 and 365 days, respectively. Table 6 also shows information regarding the remaining features.

Table 7 shows the results obtained when using these datasets to train and evaluate the proposed prognostic models using disease progression patterns, using cross-validation and grid-search as in the benchmarks.

All models learnt using patterns as features showed promising results. While all algorithms present similar AUC scores within the same dataset, SVM achieved the highest AUC and Specificity in the majority of cases. Wilcoxon Signed Ranks tests for the AUC scores outputted p-values between 0.066 and 0.465 when comparing to the benchmarks, with mostly positive ranks. When comparing to Benchmark 1, the majority of them are above 0.1 for the binary matrix and below 0.1 for the similarity matrix. When comparing to Benchmark 2, all p-values but one are below 0.068. This means there is a significant performance improvement in respect to Benchmark 2 for typical significance thresholds, but only significant improvement in respect to Benchmark 1 when using the similarity matrix. The former is expected, due to the larger pool of patterns that can be more informative than those found only in last evaluations.

As for evaluating the effect of using a similarity metric, the metrics' values do not show any clear distinction for AUC, with Wilcoxon Signed-Ranks tests outputting p-values of 0.144, 0.273 and 0.273 for 90, 180 and 365 days, respectively. However, models using the similarity matrices consistently yield higher Sensitivity and Specificity. Therefore, a similarity matrix should be used.

4.2 Pattern Evaluation

For each prediction window, the extracted patterns that remained after feature selection (features used for classification) were evaluated through GrowthRate, computed for both classes. Table 8 shows the top 5 discriminative patterns in the datasets generated through Fournier08, where the positive class corresponds to $Evolution = 1$.

When analysing the patterns kept by feature selection for each class, we observed that:

- In general, there are more discriminative patterns for the negative class, and they tend to have higher growth rates than those for the positive class. This is most likely due to relative support being used, since the negative class is also the minority one. This tendency might be atenuated or reversed if absolute support were used instead.
- Temporal patterns are, in general, more discriminative than static patterns. Also, most static patterns are featured for the positive class;
- The patterns {*R = [8,12[, PhrenMeanAmpl < 0.4*}, {*ALS-FRS = [24,36[, MIP < 40*} and {*ALS-FRS = [24,36[PhrenMeanAmpl < 0.4*} are consistently the most discriminative for the positive class when present, with Growth Rates comparable to those found in the negative class;
- The negative class patterns have a strong presence of items *R = 12* and *ALSFRSb = 12*, while the positive class patterns include lower scores for these tests. This makes sense considering the class *Evolution*: a patient with strong respiratory function (*R = 12*) is less likely to require ventilation in the short term. This is also evidenced by the many discriminative patterns for the negative class containing *R = 12* and *ALSFRSb = 12* throughout several evaluations, meaning that bulbar function may also be a relevant factor. Phrenic nerve response amplitude may also be relevant, as *PhrenMeanAmpl < 0.4* only appears in the discriminative patterns for the positive class, and *PhrenMeanAmpl ≥ 0.4* in patterns for the negative class. A similar case happens for *MIP* and *MEP*, with lower values only presenting for the positive class. *ALS-FRS*, as well as its subscores for the upper and lower limbs (*ALS-FRSsUL* and *ALS-FRSsLL*, respectively) appear with similar values in discriminative patterns for both classes, meaning they may not be as relevant.

TABLE 5
Model Evaluation Results Obtained Using Class-Labelled Binary and Similarity Matrices as Training Sets, Built From the Last Available Time Points and Over a Stratified 5x5-Fold Cross-Validation Scheme (Benchmark 2)

	Clf	Sensitivity	Specificity	AUC	Accuracy
		Binary Matrix			
90d	RF	62.2 ± 5.5	50.5 ± 6.0	59.9 ± 3.3	57.4 ± 2.7
	SVM	65.9 ± 6.5	58.0 ± 6.3	**67.0 ± 3.1**	62.7 ± 3.0
	XGB	63.0 ± 4.4	51.6 ± 6.5	62.4 ± 3.1	58.3 ± 2.6
	NB	62.5 ± 6.2	58.2 ± 5.5	65.6 ± 3.3	60.7 ± 3.1
180d	RF	76.6 ± 2.9	48.1 ± 5.1	65.3 ± 3.1	68.4 ± 2.3
	SVM	75.7 ± 4.6	56.5 ± 5.5	**70.0 ± 2.3**	70.2 ± 2.7
	XGB	79.5 ± 3.6	46.2 ± 4.3	66.3 ± 3.3	69.9 ± 3.4
	NB	71.2 ± 4.6	53.0 ± 5.4	67.4 ± 3.5	65.9 ± 3.4
365d	RF	80.9 ± 3.1	40.8 ± 10.3	66.2 ± 5.6	72.0 ± 3.0
	SVM	79.7 ± 4.7	50.4 ± 8.1	**71.2 ± 6.0**	73.2 ± 4.4
	XGB	83.8 ± 2.9	40.4 ± 8.6	66.0 ± 5.9	74.2 ± 3.1
	NB	74.7 ± 3.7	56.3 ± 7.9	70.3 ± 5.8	70.6 ± 3.5
Similarity Matrix					
90d	RF	65.8 ± 4.8	55.8 ± 7.4	65.2 ± 3.4	61.7 ± 3.0
	SVM	64.7 ± 6.8	61.3 ± 5.9	**67.3 ± 3.1**	63.3 ± 3.6
	XGB	65.9 ± 4.2	54.4 ± 5.4	65.0 ± 4.0	61.2 ± 2.5
	NB	58.2 ± 6.8	55.7 ± 7.0	59.8 ± 4.1	57.2 ± 3.8
180d	RF	79.9 ± 3.3	51.4 ± 5.6	**70.5 ± 3.0**	71.7 ± 2.2
	SVM	70.0 ± 4.3	58.6 ± 5.5	69.3 ± 2.2	66.7 ± 2.8
	XGB	79.2 ± 2.9	50.2 ± 5.6	69.7 ± 2.8	70.8 ± 2.7
	NB	59.0 ± 5.5	56.4 ± 5.2	61.6 ± 3.6	58.2 ± 4.1
365d	RF	83.4 ± 3.4	44.8 ± 8.2	68.5 ± 5.1	74.9 ± 3.0
	SVM	72.8 ± 3.2	57.3 ± 10.7	**69.6 ± 5.7**	69.4 ± 3.5
	XGB	83.8 ± 3.0	44.1 ± 9.6	67.1 ± 5.5	75.0 ± 3.2
	NB	65.8 ± 5.1	57.1 ± 11.2	65.8 ± 5.8	63.9 ± 3.9

We present the average and standard deviation over all folds. The best AUC for each dataset appears in bold.

TABLE 7
Model Evaluation Results Obtained Using Class-Labelled Binary and Similarity Matrices as Training Sets, With a Random Undersampling/SMOTE Balancing Strategy and Over a Stratified 5x5-Fold Cross-Validation Scheme

	Clf	Sensitivity	Specificity	AUC	Accuracy
		Binary Matrix			
90d	RF	66.6 ± 4.3	53.8 ± 5.0	65.1 ± 3.5	61.3 ± 3.0
	SVM	64.4 ± 4.2	66.0 ± 5.4	**70.4 ± 3.0**	65.0 ± 2.6
	XGB	65.2 ± 4.6	55.3 ± 6.0	64.7 ± 3.5	61.1 ± 3.3
	NB	65.3 ± 4.2	63.7 ± 4.8	69.8 ± 3.1	64.6 ± 2.8
180d	RF	81.1 ± 3.3	48.4 ± 7.0	72.5 ± 2.3	71.6 ± 2.2
	SVM	67.7 ± 3.5	67.8 ± 6.9	**75.2 ± 2.7**	67.7 ± 2.4
	XGB	77.6 ± 3.2	49.6 ± 6.1	70.7 ± 2.9	69.6 ± 2.4
	NB	71.0 ± 4.2	58.8 ± 5.7	71.8 ± 2.2	67.5 ± 2.7
365d	RF	86.8 ± 3.0	47.0 ± 8.8	75.9 ± 3.8	72.0 ± 3.0
	SVM	72.6 ± 4.6	67.8 ± 6.9	**76.9 ± 3.6**	78.0 ± 2.2
	XG	85.3 ± 2.6	48.3 ± 7.9	75.6 ± 3.6	71.5 ± 3.6
	NB	73.2 ± 2.7	54.9 ± 7.9	73.2 ± 4.8	77.1 ± 2.5
Similarity Matrix					
90d	RF	71.2 ± 5.1	60.1 ± 5.9	71.9 ± 3.5	66.6 ± 3.2
	SVM	66.8 ± 5.4	66.2 ± 5.9	**72.5 ± 3.1**	66.5 ± 3.0
	XGB	69.4 ± 4.1	59.3 ± 5.2	70.7 ± 3.4	65.2 ± 2.4
	NB	65.0 ± 4.6	61.6 ± 5.9	68.6 ± 3.2	63.6 ± 3.1
180d	RF	83.4 ± 3.5	52.7 ± 9.0	**77.2 ± 2.4**	74.6 ± 2.3
	SVM	67.2 ± 5.6	70.2 ± 7.1	76.3 ± 2.8	68.0 ± 3.2
	XGB	81.2 ± 3.8	56.1 ± 7.8	76.2 ± 2.5	74.0 ± 2.7
	NB	60.0 ± 5.9	68.7 ± 5.8	69.2 ± 3.0	62.5 ± 3.6
365d	RF	88.9 ± 3.3	50.9 ± 7.0	**78.8 ± 4.2**	80.5 ± 3.2
	SVM	77.8 ± 3.8	62.0 ± 9.3	77.2 ± 4.6	74.3 ± 3.5
	XGB	86.7 ± 3.3	53.7 ± 8.7	78.3 ± 4.1	79.4 ± 3.1
	NB	59.2 ± 5.2	72.0 ± 8.4	71.2 ± 5.7	62.0 ± 3.9

We present the average and standard deviation over all folds. The best AUC for each dataset appears in bold.

TABLE 6
Information Regarding the Datasets Generated for Each Prediction Window, Using a Minimum Support of 25 percent per Class in SPMF Algorithms

	No. Patients	No. Patterns (Before FS)	No. Patterns (After FS)	Class distribution
90d	940	359	34	Y: 551 (58.62%)
		40 S + 319 T	13 S + 21 T	N: 389 (41.38%)
180d	895	666	71	Y: 637 (71.17%)
		40 S + 626 T	16 S + 55 T	N: 258 (28.83%)
365d	840	1014	83	Y: 654 (77.86%)
		48 S + 966 T	22 S + 61 T	N: 186 (22.14%)

The number of patients and patterns are the number of learning instances and features, respectively. The class is Evolution.

Additionally, we can further observe that the majority of patterns is considerably small, usually containing less than 6 items and/or temporal instances. Since no stratification strategies are used, large patterns that can exist among patients with slower progress do not meet the required support. It is also important to note that many items are not featured in these patterns. This can be due to the discretization process considering interpretability over even distribution, meaning some value ranges may not have enough support to be featured. Furthermore, since most patients display high test values at onset, namely high *ALS-FRS* subscores, the items containing these values will be more prevalent. Other tests had a high number of missing values to begin with, indicating a lower probability of them appearing in the extracted patterns.

5 Conclusion and Future Work

In this work, we proposed a prognostic prediction approach to predict the need of Non-Invasive Ventilation in ALS patients, making use of their complete evaluation history through Pattern Mining techniques. The extracted patterns served as features for building and evaluating several predictive models using time windows of 90, 180 and 365 days.

Model evaluation showed promising results for the three prediction windows. Both approaches used for classification presented similar results in model learning, with slightly better outcomes obtained when using a similarity matrix instead of a binary matrix. Thus, it could be relevant to explore more insightful metrics to compare patterns. Pattern

TABLE 8
Top 5 Discriminative Patterns for Each Prediction Window, Extracted With Fournier08

GR (class)	Pattern
90 days	
2.73 (+)	< 0 > R = [8,12[, PhrenMeanAmpl < 0.4
1.97 (+)	< 0 > ALS-FRS = [24,36[, MIP < 40
1.93 (-)	< 0 > R = 12, CervicalExt = 5
1.93 (+)	< 0 > ALS-FRS = [24,36[, PhrenMeanAmpl < 0.4
1.92 (+)	< 0 > MIP < 40, MEP < 40
180 days	
4.98 (+)	< 0 > R = [8,12[, PhrenMeanAmpl < 0.4
2.39 (-)	< 0 > ALS-FRSb = 12, R = 12 | < 1 > ALS-FRSb = 12 | < 2 > ALS-FRSb = 12, R = 12 | < 3 > R = 12
2.36 (-)	< 0 > ALS-FRSsUL = 12, R = 12 | < 1 > R = 12
2.27 (+)	< 0 > ALS-FRS = [24,36[, MIP < 40
2.18 (-)	< 0 > ALS-FRSb = 12, R = 12 | < 1 > ALS-FRSsLL = [8,12[
365 days	
3.82 (-)	< 0 > ALS-FRSb = 12, ALS-FRSsUL = 12, R = 12
3.45 (+)	< 0 > ALS-FRS = [24,36[, MIP < 40
3.06 (-)	< 0 > ALS-FRS ≥ 36, ALS-FRSb = 12, ALS-FRSsLL = [8,12[
2.62 (+)	< 0 > ALS-FRS = [24,36[, PhrenMeanAmpl < 0.4
2.57 (-)	< 0 > ALS-FRSb = 12, R = 12 | < 1 > R = 12 | < 2 > ALS-FRSsUL = [8,12[, R = 12

< t > *denotes the time point.*

selection can play a significant role in further studies, whether in feature selection or pattern extraction. The pattern mining process is highly dependent on the selected parameters and discretization strategy chosen. Therefore, it could be beneficial to explore alternative strategies, including other pattern extraction techniques. A natural follow-up would be to make use of state-of-the-art pattern-based biclustering algorithms [35], following a recent work by Henriques and Madeira [44], assessing the impact of discriminative patterns with varying coherence and quality on associative classification and proposing a bicluster-based classifier.

Furthermore, the proposed approach does not take into account the heterogeneity found among ALS patients, namely in progression rate. Stratification strategies as the ones explored by Pires *et al.* [29], [36] can also be implemented, thus learning specialized prediction models for groups of patients displaying similar progression. However, this strategy can be highly impactful to the pattern extraction process, as slow progressors can be expected to exhibit longer patterns and in larger amount, while fast progressors have few evaluations and therefore a decreased likelihood of a large number of patterns being found. To minimize this issue, different prediction windows may also be explored according to the progression rate, as more snapshots could be made eligible. Further, since class is computed by comparing the NIV status between evaluations, it is highly dependent on the chosen prediction window. Therefore, shorter time windows can favour class distribution among fast-progressing patients, with the same occurring with larger time windows for slow-progressing patients.

Ethics Approval and Consent to Participate

The study was conducted in accordance with the Declaration of Helsinki and was approved by the local (Faculty of Medicine, University of Lisbon) ethics committee. Informed consent to participate in the study was obtained from all participants. Data access was granted in the context of project NEUROCLINOMICS2 (PTDC/EEI-SII/1937/2014), where the authors' institutions participate.

Acknowledgments

This work was supported in part by Fundao para a Ciłncia e a Tecnologia , FCT, the Portuguese Public Agency for Science, Technology and Innovation, through projects NEUROCLINOMICS2, PREDICT, and AIpALS under Grants PTDC/EEI-SII/1937/2014, PTDC/CCI-CIF/29877/2017, and PTDC/CCI-CIF/4613/2020 and in part by the LASIGE Research Unit under Grants UIDB/00408/2020 and UIDP/00408/2020.

References

[1] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[3] S. Zarei *et al.*, “A comprehensive review of amyotrophic lateral sclerosis,” *Surg. Neurol. Int.*, vol. 6, 2015, Art. no. 171.

[4] R. H. Brown and A. Al-Chalabi , “Amyotrophic lateral sclerosis,” *New Engl. J. Med.*, vol. 377, pp. 162–172, 2017.

[5] A. G. Solimando *et al.*, “Move toward to non invasive mechanical ventilation in amyotrophic lateral sclerosis: A clinical review,” *J. Clin. Genomic*, vol. 1, no. 1, 2018.

[6] S. Martin, A. Al Khleifat , and A. Al-Chalabi , “What causes amyotrophic lateral sclerosis?,” *F1000Res.*, vol. 6, 2017, Art. no. 371.

[7] G. Pfeffer, M. Povitz, G. J. Gibson, and P. F. Chinnery, “Diagnosis of muscle diseases presenting with early respiratory failure,” *J. Neurol.*, vol. 262, pp. 1101–1114, 2015.

[8] S. Vucic, J. D. Rothstein, and M. C. Kiernan, “Advances in treating amyotrophic lateral sclerosis: Insights from pathophysiological studies,” *Trends Neurosci.*,vol. 37, no. 8, pp. 433–442, 2014.

[9] M. C. Kiernan *et al.*, “Amyotrophic lateral sclerosis,” *Lancet*, vol. 377, pp. 942–955, 2011.

[10] P. R. Mehta *et al.*, “Younger age of onset in familial amyotrophic lateral sclerosis is a result of pathogenic gene variants, rather than ascertainment bias, ”*J. Neurol. Neurosurg. Psychiatry*, vol. 90, pp. 268–271, 2019.

[11] L. I. Grad, G. A. Rouleau, J. Ravits, and N. R. Cashman, “Clinical spectrum of amyotrophic lateral sclerosis (ALS),” *Cold Spring Harb Perspect Med.*, vol. 7, 2017, Art. no. a024117.

[12] C. Tard, L. Defebvre, C. Moreau, D. Devos, and V. Danel-Brunaud, ”Clinical features of amyotrophic lateral sclerosis and their prognostic value,” *Revue Neurologique*, vol. 173, no. 5, pp. 263–272, 2017.

[13] M. A. van Es *et al.*, “Amyotrophic lateral sclerosis,” *Lancet*, vol. 390, pp. 2084–2098, 2017.

[14] R. Bhandari, A. Kuhad, and A. Kuhad, “Edaravone: A new hope for deadly amyotrophic lateral sclerosis,” *Drugs Today*, vol. 54, no. 6, pp. 349–360, 2018.

[15] S. Piepers and L. H. van den Berg, “Evidence-based care in amyotrophic lateral sclerosis,” *Lancet Neurol.*, vol. 5, no. 2, pp. 105–106, 2006.

[16] O. Hardiman, “Management of respiratory symptoms in ALS,” *J. Neurol.*, vol. 258, no. 3 pp. 359–365, 2011.

[17] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, “Discovering frequent closed itemsets for association rules,” in *Proc. Int. Conf. Database Theory*, 1999, pp. 398–416.

[18] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in *Proc. Int. Conf. Very Large Data Bases*, 1994, pp. 487–499.

[19] J. Pei *et al.*, "Mining sequential patterns by pattern-growth: The prefixspan approach," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 11, pp. 1424–1440, Nov. 2004.

[20] P. Fournier-Viger , R. Nkambou, and E. M. Nguifo, "A knowledge discovery framework for learning task models from user interactions in intelligent tutoring systems," in *Proc. Mex. Int. Conf. Artif. Intell: Adv. Artif. Intell.*, 2008, pp. 765–778.

[21] M. J. Zaki and W. Meira, *Data Mining and Analysis: Fundamental Concepts and Algorithms*, 1st ed. New York, NY, USA: Cambridge Univ. Press, 2014.

[22] Y. Hirate and H. Yamana, "Generalized sequential pattern mining with item intervals," *J. Comput.*,vol. 1, no. 3, pp. 51–60, 2006.

[23] J. Wang and J. Han, "BIDE: Efficient mining of frequent closed sequences," in *Proc. Int. Conf. Data Eng.*, 2004, pp. 79–90.

[24] M. A. Myszczynska *et al.*, "Applications of machine learning to diagnosis and treatment of neurodegenerative diseases," *Nat. Rev. Neurol.*, vol. 16 , pp. 440–456, 2020.

[25] V. Grollemund *et al.*, "Machine learning in amyotrophic lateral sclerosis: Achievements, pitfalls, and future directions," *Front. Neurosci.*, vol. 13, 2019, Art. no. 135.

[26] A. V. Carreiro, "An integrative mining approach for prognostic prediction inneurodegenerative diseases," Ph.D. dissertation, Instituto Superior Tcnico, Lisbon, Portugal, 2006.

[27] A. V. Carreiro, S. Pinto, M. de Carvalho, S. C. Madeira, and C. Antunes, "Classification of clinical data using sequential patterns: A case study in amyotrophic lateral sclerosis, "in *Proc. Workshop Data Mining Healthcare Med. SIAM Int. Conf. Data Mining*, 2013, pp. 61–69.

[28] Sofia Pires, A supervised learning approach for prognostic prediction in als using disease progression groups and patient profiles, Master's thesis, Faculty Sci., Univ. Lisbon, Portugal, 2018.

[29] S. Pires, M. Gromicho, S. Pinto, M. de Carvalho, and S. C. Madeira, "Predicting non-invasive ventilation in ALS patients using stratified disease progression groups," in *Proc. IEEE Int. Conf. Data Mining Workshops*, 2018, pp. 748–757.

[30] A. V. Carreiro, P. M. T. Amaral, S. Pinto, 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, 2015.

[31] J. Matos, "Biclustering electronic health records to unravel disease presentation patterns," Master's thesis, Faculty Sci., Univ. Lisbon, Portugal, 2019.

[32] J. Matos *et al.*, "Unravelling disease presentation patterns in als using biclustering for discriminative meta-features discovery," in *Proc. Int. Work-Conf. Bioinf. Biomed. Eng.*, 2020, pp. 517–528.

[33] D. Soares, R. Henriques, M. Gromicho, S. Pinto, M. de Carvalho, and S. C. Madeira, "Towards triclustering-based classification of three-way clinical data: A case study on predicting non-invasive ventilation in ALS," in *Proc. Int. Conf. Practical Appl. Comput. Bio. Bioinf.*, 2020, pp. 112–122.

[34] L. Zhao and M. J. Zaki, "TRICLUSTER: An effective algorithm for mining coherent clusters in 3D microarray data," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2005, pp. 694–705.

[35] R. Henriques, F. L. Ferreira, and S. C. Madeira, "BicPAMS: Software for biological data analysis with pattern-based biclustering," in *BMC Bioinf.*, vol. 18, no. 1, 2017, Art. no. 82 .

[36] 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 *Proc. Int. Work-Conf. Bioinf. Biomed. Eng.*, 2020, pp. 529–541.

[37] P. M. Andersen *et al.*, "EFNS guidelines on the clinical management of amyotrophic lateral sclerosis (MALS)–revised report of an EFNS task force," *Eur. J. Neurol.*, vol. 19, pp. 360–375, 2011.

[38] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[39] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[40] 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, 2019.

[41] A. Chio *et al.*, "Global epidemiology of amyotrophic lateral sclerosis: A systematic review of the published literature," *Neuroepidemiology* , vol. 41, no. 2, pp. 118–130, 2013.

[42] P. Fournier-Viger , J. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, "A survey of sequential pattern mining," *Data Sci. Pattern Recognit.* , vol. 1, no. 1, pp. 54–77, 2017.

[43] P. Fournier-Viger *et al.*, "The SPMF Open-Source Data Mining Library Version 2," in *Proc. 19th Eur. Conf. Princ. Data Mining Knowl. Discov. Part III*, pp. 36–40, 2016.

[44] R. Henriques and S. C. Madeira, "FleBiC: Learning classifiers from high-dimensional biomedical data using discriminative biclusters with non-constant patterns," *Pattern Recognit.*, vol. 115, 2021, Art. no. 107900.

Andreia S. Martins received the BSc degree in physics in 2018 from Ciências - ULisboa, where she is currently working toward the MSc degree in data science. Her research interests include machine learning and data mining and their applications in biomedical and physical sciences.

Marta Gromicho received the PhD degree in evolutionary biology. She was involved in several national and international R&D projects in biomedical and environmental scientific areas. She is currently a senior researcher the Physiology lab, Instituto de Medicina Molecular, Lisbon, Portugal. Her research interests include amyotrophic lateral sclerosis, molecular biomarkers of disease onset, prognosis and progression, risk factors, clinical and genomic data integration, and epidemiology.

Susana Pinto is currently a physician, researcher, and an assistant professor of physiology, with large expertise in neuromuscular or neurodegenerative disorders, and especial interest in amyotrophic lateral sclerosis, including in the interplays genotype-phenotype-environment, deficiency-body function-activity-participation, and survival-quality of life, as well as respiratory evaluation and support, telemedicine and new technologies.

Mamede de Carvalho studies neurodegenerative and neuromuscular disorders, in particular applying neurophysiological methods to investigate disease mechanisms and adaptation, studying how the disorder of the motor system impact on patients function, and exploring the interplay between environmental features-genotype-phenotype in those diseases. He has an educational background on neurology and neurophysiology, and more than 30 years continuous experience in medical research. Different areas have been involved in his investigations. For these purposes, he has established a strong net of national and international collaborations, and developed a large unit covering different expertises.

Sara C. Madeira received the PhD degree in computer science and engineering from Técnico - ULisboa. She is currently an associate professor with Ciłncias - ULisboa, where she coordinates the Graduation in data science. She is currnetly a senior researcher with LASIGE, where she coordinates the Data and Systems Intelligence Research Line of Excellence. Her research interests include machine learning algorithms, with afocus on biclustering and triclustering, and their applications to complex data analysis in health and biomedical informatics. She was the PI of R&D projects NEUROCLINOMICS (PTDC/EIA-EIA/111239/2009) and NEUROCLINOMICS2 Unravelling Prognostic Markers in NEUROdegenerative diseases through CLINical and OMICS data integration (PTDC/EEI-SII/1937/2014), and is currently leading AIpALS advanced learnIng models using Patient profiles and disease progression patterns for prognostic prediction in ALS (PTDC/CCI-CIF/4613/2020).