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

Andreia S. Martins
LASIGE, Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa
Lisbon, Portugal
amartins@lasige.di.fc.ul.pt

Marta Gromicho
Instituto de Medicina Molecular, Instituto de Fisiologia, Faculdade de Medicina, Universidade de Lisboa
Lisbon, Portugal
martalgms@gmail.com

Susana Pinto
Instituto de Medicina Molecular, Instituto de Fisiologia, Faculdade de Medicina, Universidade de Lisboa
Lisbon, Portugal
susana.c.pinto@medicina.ulisboa.pt

Mamede de Carvalho
Instituto de Medicina Molecular, Instituto de Fisiologia, Faculdade de Medicina, Universidade de Lisboa
Lisbon, Portugal
mamedealves@medicina.ulisboa.pt

Sara C. Madeira
LASIGE, Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa
Lisbon, Portugal
sacmadeira@ciencias.ulisboa.pt

ABSTRACT

Amyotrophic Lateral Sclerosis (ALS) is a devastating neurodegenerative disease causing rapid degeneration of motor neurons and usually leading to death by respiratory failure 3 to 5 years after symptom onset. Since no cure is available, the goal of treatment is to improve symptoms and prolong survival. Non-invasive Ventilation (NIV) is known to be 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 the 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 proposed prognostic models showed promising results. Pattern evaluation through growth rate suggests that 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.

KEYWORDS

Amyotrophic Lateral Sclerosis, Pattern Mining, Disease Progression Patterns, Non-Invasive Ventilation, Prognostic Prediction

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 3-5 years due to respiratory failure. Most available treatments focus on symptom management and only a few were shown to increase life expectancy by a few months. Non-invasive Ventilation (NIV) is the treatment which has shown most promising results in increasing both life expectancy and quality of life, with evidence suggesting an earlier initiation provides greater benefit. However, not all patients are promptly eligible and there is some debate on the best timing to initiate NIV. In this context, it makes sense to investigate whether it is possible to effectively predict if patients will or not be eligible for NIV in the near future, based on their evolution. We thus formulated the following problem (schematized in Figure 1): given static data collected at diagnosis and temporal data from a patient's follow-up, can we effectively use these clinical evaluations to predict if this patient will require NIV within k days of the last evaluation?

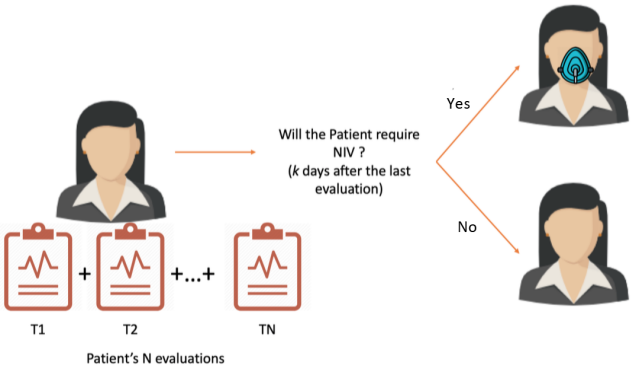

Figure 1: Problem formulation: given the patient's evaluations over time, can we predict the need for Non-invasive Ventilation (NIV) within k days of the last evaluation?

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 directely as features, the goal is to learn patterns from data (both disease presentation patterns and disease progression patterns, discovered in static and temporal data, respectively) and use them as features, enabling to take disease progression into account in the predictions and promoting model interpretability. Unlike when training models using the original features, which usually can only be trained with considerering the same number of time points for all patients to avoid introducing artificial missing values, in the proposed approach we can use all available time points for each patient without adding missing values. We can thus use more learning instances to train the models, while finding putatively interesting 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 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 Forest, SVM and XGBoost [1]. SMOTE [14] is used to deal with class imbalance.

This paper is organized as follows: Section 2 provides an introduction to ALS and to the Pattern Mining algorithms used; Section 3 explains the procedures performed for data transformation, pattern extraction and evaluation, as well as classification; Section 4 provides the results obtained for each predictive model, as well as a summary of 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 [31]. It spreads from an initial focal weakness, eventually affecting most 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, stiffness and cramping of the limbs. Difficulty with swallowing, chewing and word articulation are initial symptoms linked to a bulbar onset, which represents about a third of all cases [4, 25]. Respiratory onset typically presents with specific initial symptoms, such as dyspnea, orthopnea and sleeping problems. However, it makes up less than 3% of all cases, with respiratory impairment being considered an end-stage event in disease progression [15, 20, 25]. Up to 50% of ALS patients also develop mild cognitive impairment, such as memory loss and behaviour changes [28].

ALS has a prevalence of 3-5 cases per 100000 people and can appear at any age [25, 28]. 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 [13, 17]. ALS has been linked to mutations in several genes, such as *SOD1* and *C9orf72*. Even though only about 10% of patients have a familiar history of ALS, gene mutations associated with familial ALS are seen in 5-10% of cases labelled as sporadic [10, 26, 27].

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, 27]. Edaravone, not yet approved in the EU, has shown to slow down the loss of physical function by 33% relatively to the placebo treatment [3]. 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 a lower feasibility in patients with a bulbar onset, a large cohort study has suggested NIV also increases life expectancy in these cases [21, 27]. As such, NIV is advisable to most patients, at the onset of respiratory symptoms [11]. However, current guidelines do not concede NIV access to all patients readily. Given this, several efforts have been made to determine the best time to initiate NIV, and the most relevant evaluation measures to determine eligibility [25].

2.2 Pattern Mining

In this work, we use itemset mining together with sequential pattern mining to discover both disease presentation patterns and disease progression patterns, discovered in static and temporal data, respectively. These patterns are then used to extract new features for classification. In order to construct our training sets, we needed the pattern mining algorithms to output the id's of the sequences or transactions where the patterns occured. To this aim, AprioriTID-Close [2, 18] was used to retrieve closed patterns from the static features, while PrefixSpan [19] and Fournier08 [9] were used to retrieve closed sequential patterns from the temporal features. The use of two algorithms for sequential pattern retrieval enables to assess the effect of temporal constraints in classification: PrefixSpan only takes item occurrence order into account, while Fournier08 also considers the time stamps. Closed patterns were chosen as to minimize the risk of extracting a large amount of patterns, many of which would be subsets of each other.

Follows a brief introduction on the pattern and sequencial pattern mining algorithms used in this work and mentioned above, which might be skiped according to reader's background. All pattern mining algorithms used are available in the SPMF library [8].

2.2.1 Itemset Mining. Itemset Mining (IM) is a set of unsupervised learning techniques meant to find objects of interest that occur together within a transaction database [30], 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 *(attribute, value)*, such as *"Gender = Male"*. As such, each transaction consists of a patient identifier and an itemset that contains all known static features for the patient. We focus on closed itemsets: they appear with a frequency (or support) above a preset threshold and are not contained within supersets with the same support. This minimizes the risk of extracting too many patterns, many of which would be 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 [30]. In this work, the implementations of the AprioriTID algorithm in the SPMF library were used to compute both frequent and closed itemsets.

The **AprioriTID** algorithm 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. The property is based on two observations: (1) If a k-itemset is frequent, then all of its subsets must also be frequent; (2) If a k-itemset is not frequent, then none of its supersets can be frequent [30]. In practice, this means that for a certain candidate k+1-itemset, support is only computed if all of its k-subsets are frequent. The AprioriTIDClose algorithm [18] discovers the closed frequent itemsets first, using a closure mechanism based on the Galois connection. According its original proposal, the frequent itemsets can then be extracted from the closed ones. The 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 [30]. 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 and each item a pair *(exam, value)*.

There are several algorithms that perform SPM, with varying degrees of efficiency and complexity. As with Itemset Mining, there are also different algorithms to retrieve closed sequential patterns. There are also algorithms that take time constraints into account, allowing for the retrieval of sequences with specific lengths or gaps between time points [9]. PrefixSpan [19] looks into item occurrence order, but not into time intervals between transactions. This means it would see two sequences containing the same itemsets at different time points and consider they observe the same pattern [12].

The **Hirate-Yamana** algorithm [12] takes into account the transaction timestamps and allows for sequential pattern retrieval with specific time intervals between transactions. Since it is based on PrefixSpan, this algorithm functions in a similar matter. At each projected database, the suffix of each item occurrence is still kept. However, the respective timestamp changes to translate the new time intervals between transactions. There are also 4 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 [9] is an extension of Hirate-Yamana, but has integrated the BIDE+ algorithm closure checking. This feature allows for a closed sequence set to be mantained without a closed candidate set, and therefore, the complete frequent sequence set does not have to be explored. It has also implemented the BackScan pruning from the BIDE+ algorithm [29].

2.3 Related Work

Carreiro et al. [5] explored the use of Sequential Pattern Mining to extract features that encode temporal dependencies, employing standard classifiers to predict the need of NIV within k days of the last evaluation, using both a fixed and variable number of time points. In [7], they also used Sequential Pattern Mining for classification, but in diagnosis. Pires [22] also used the original features for prognosis 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 models using the last evaluation always outperformed the others. Pires et al. [23] explored patient stratification improving these results, but no temporal data was used.

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 dataset and its challenges. Then, we discuss the required preprocessing and the details about model learning. Figure 2 presents the methodology workflow, highlighting the key steps, discussed below.

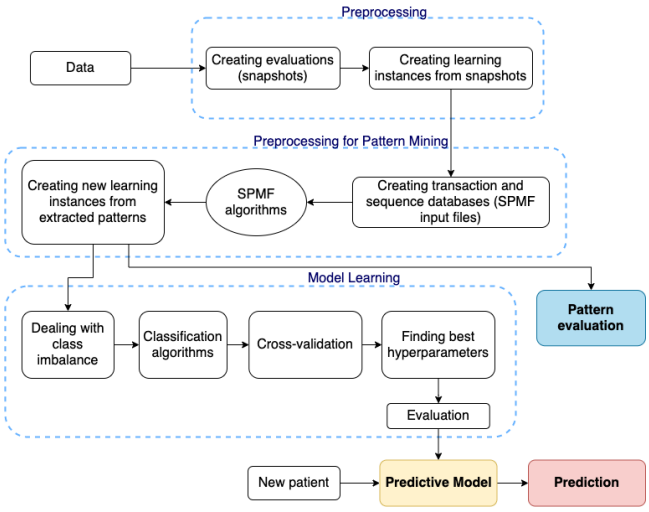

Figure 2: Workflow of the methodology used for prognostic prediction using disease progression patterns.

3.1 About the Portuguese ALS dataset

The dataset used for this work is the Portuguese ALS dataset, which contains demographic information and clinical data, such as respiratory and functional test results, for a cohort of 1375 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. [5, 6] and Pires et al. [22].

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 will need NIV within k days must also be created, since the dataset only states the NIV status and/or NIV initiation date. This class will be named *Evolution*. The transformation into evaluation sets would not be enough if using the original features, since now each patient has 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 input files for the pattern mining algorithms and transformation of the output files into 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 Figure 2 showing all these key steps.

3.2 Data Preprocessing

Data transformation is performed as described in [6] and [22], and consists of two modules. The first module 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 takes the snapshot set and creates a set of learning instances, according to the selected time window, k: 90, 180 or 365 days. It also generates the class, *Evolution*, for each instance.

3.2.1 Creating Snapshots. The 1st module takes the original datasets, one per feature group. It transforms them into a set of snapshots (1 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 ordered by date and grouped through Agglomerative Hierarchical Clustering. This is performed taking two constraints into account:1) Each exam can only appear once in a snapshot; and 2) all exams contained 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 latter is considered the snapshot's date, and will be used as reference for future stages. Note that, since each patient typically performs an exam several times, most patients will have more than one shapshot associated to them.

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 will require NIV within k days from the last evaluation, each learning instance created requires knowing the NIV status from 2 snapshots, and how long is the prediction window k. Like in previous work performed on the Portuguese ALS dataset, the time windows chosen are of 90, 180 or 365 days [5, 6, 22]. 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 $n < k$, then the patient evolves if the starting NIV status is 0 and finishing status is 1 (E=1), and if both status are 0, then there is no Evolution (E=0); if $n > k$, then the patient does not evolve (E=0).

As mentioned before, only eligible snapshots can be used to generate learning instances. If the patient has already initiated NIV at a given snapshot, it's not necessary to explore the following snapshots: the patient has already evolved to require NIV. If there is no information on NIV status past the selected time window, there is no use to keep the snapshot as well: we can't 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 by comparison with 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 of a snapshot and class label. However, using these instances for classification equates to using only one evaluation for prognosis prediction. Instead, we are looking to make these predictions based on the patients' complete evaluation history, and moreso, to use patterns found in the data for the purpose of prediction. This means that additional data processing steps have to be made. The datasets obtained are therefore used to extract patterns, and will require additional data processing steps. All transformation steps were done using Python, with pattern extaction being performed with the SPMF library. After that, new datasets will be generated and used to train and evaluate several classifiers.

First, we have to transform the sets of learning instances into sequence and transaction databases (for temporal and static features, respectively), so that patterns can be extracted. Second, we have to generate new learning instances from the collected patterns.

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 section 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. Continous attributes in both groups were discretized, and some categorical features had their values regrouped, according to both the studies performed by Matos [16] and feedback from clinicians. We are left with a total of 21 features, plus the patient identifier, which originate 33 static ans 40 temporal items. These are of the form *Attribute = value interval*, such as *ALS-FRS = [24,36[*, or *Attribute = value*, such as *Gender = Male*. The complete set of features and values used is in Table 1.

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 attributes to which we have a known value, meaning there is no need for missing value imputation. Input files were created for each of the prediction windows, with items identified by an integer key (as required by SPMF) and a legend added at the top of each file. The temporal data was 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. Two versions of this data were generated: one with timestamps (indicating the snapshot, to be used by the Fournier08 algorithm) and one without (to be used by Prefixspan). Input files were generated similarly to the static case for all considered prediction windows. Therefore, a total of 3 datasets was generated per time window: static data for Itemset Mining with AprioriTIDClose, temporal data with no timestamps for SPM with PrefixSpan and temporal data with timestamps for SPM with Fournier08.

3.3.2 Pattern Extraction in SPMF. The generated input files are given to the SPMF library, which outputs the extracted patterns after selecting the correspoding algorithm and parameters. AprioriTIDClose was used for Itemset Mining, PrefixSpan for SPM without time constraints and Fournier08 for SPM with time constraints. The support threshold considered was 24%, 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 at 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 time points, thus the timestamps presented in the patterns are relative.

3.4 Training Set Creation and Model Learning

The SPMF output files were then used to build a binary matrix, to which the class is added, making the new datasets to be used by the classifiers. This is done by creating reference files containing information about all patterns retrieved for each prediction window: support, number of time points, number of items and list of patients that verifies each pattern. The trivial sequences and itemsets, ie, the ones containing only one item, are discarded. A binary matrix is computed from the remaining patterns, where each row is a patient and each column is a pattern. If the patient verifies the pattern, then the value is 1 in the corresponding element, and 0 otherwise. The class is the *Evolution* value present in the patient's final snapshot. This results in two datasets for each time window, one for each type of patterns (with and without timestamps). The information about the resulting datasets is found in Table 2.

As expected, Fournier08 extracts less patterns than PrefixSpan for the same data, due to the latter not discerning between temporal instants. When timestamps are included, more patterns are found but less of them meet the support threshold. We can also see that the number of eligible patients decreases as the time window increases, as a lengthier medical history is required. 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.

These datasets are then used to train and evaluate several prognosis models, using the state-of-the-art classifiers Random Forests, Support Vector Machines and XG-Boost.

3.5 Pattern Evaluation

In order to assess the relevance of each pattern towards classification, an extra step is added to the preprocessing for pattern mining stage, where patterns are evaluated based on their support for both classes. For each pattern P, we compute the relative support for the positive and negative class. Then, we compute Growth Rate, measuring how support grows from class A to class B:

$$GrowthRate_{A\to B}(P) = \begin{cases} \dfrac{supp_B(P)}{supp_A(P)}, & supp_A(P) \neq 0 \\ 0, & supp_A(P) = 0 = supp_B(P) \\ \infty, & supp_A(P) = 0 \neq supp_B(P) \end{cases} \quad (1)$$

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

$$GrowthRate_{pos}(P) = GrowthRate_{neg\to pos}(P) > \alpha \quad (2)$$

And discriminative for the negative class if:

$$GrowthRate_{neg}(P) = GrowthRate_{pos\to neg}(P) > \alpha \quad (3)$$

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

This section presents and discusses the 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 also highlighting the importance of pattern evaluation for model interpretability.

4.1 Model Evaluation

The datasets previously described were used to build and evaluate classifiers. SMOTE was performed for class balancing, improving the results. Three algorithms were used: Random Forest, SVM with Gaussian kernel and XGBoost. Stratified 5×10-fold cross-validation

Type	Feature	Values (post-discretization)
Static	Gender	Male, Female
	Body Mass Index: BMI	< 20, $[20, 25[$, $[25, 30[$, ≥ 30
	Familiar history of motor neuron disease: MND familiar history	Yes, No, Unknown
	Age at onset (years)	< 30, $[30, 50[$, $[50, 70[$, ≥ 70
	Disease duration (months)	≤ 6, $]6, 12]$, $]12, 18]$, $]18, 36]$, > 36
	El Escorial Reviewed Criteria	definitive, probable, possible, suspected
	Affected motor neurons: UMN vs LMN	UMN, LMN, both
	Onset form	spinal, bulbar, respiratory/axial, mixed, FTD
	Expression of mutations in C9orf72 gene: C9orf72	Yes, No, Unknown
Temporal	Functional Rating Scale: ALS-FRS	< 12, $[12, 24[$, $[24, 36[$, ≥ 36
	Bulbar subscore: ALS-FRSb	< 4, $[4, 7]$, $[8, 11]$, 12
	Upper limb subscore: ALS-FRSsUL	
	Lower limb subscore: ALS-FRSsLL	
	Revised respiratory subscore: R	
	Maximal Sniff Nasal Inspiratory Pressure: SNIP	< 40, $[40, 60[$, ≥ 60
	Maximal Inspiratory Pressure: MIP(%)	
	Forced Vital Capacity: FVC (%)	< 40, $[40, 60[$, $[60, 80[$, ≥ 80
	Maximal Expiratory Pressure: MEP (%)	
	Phrenic Nerve Response Amplitude: PhrenMeanAmpl	< 0.4, ≥ 0.4
	Cervical Flexion: CervicalFlex	< 5, 5
	Cervical Extension: CervicalExt	

Table 1: Summary of the features and values used for Itemset and Sequential Pattern Mining. The attribute *REF* is a patient identifier and therefore not included in the databases used for pattern extraction.

Time Window	Nr. instances (patients)	Nr. features		Class distributions
		SPs by PrefixSpan	SPs by Fournier08	
90d	940	37 static 202 temporal 239 total	37 static 161 temporal 198 total	Y: 551 (58.62%) N: 389 (41.38%)
180d	895	39 static 216 temporal 255 total	39 static 169 temporal 208 total	Y: 637 (71.17%) N: 258 (28.83%)
365d	840	37 static 191 temporal 228 total	37 static 155 temporal 192 total	Y: 654 (77.86%) N: 186 (22.14%)

Table 2: Information regarding each generated dataset, using a minimum support of 24% in the SPMF algorithms. In all cases the class is *Evolution*.

(CV) and a Grid Search was performed to determine the best hyperparameters for each classifier. Table 3 shows the tested values.

Algorithm	Parameter	Tested values
RF	Max. features	{1,2,5,10,20}
	Nr. estimators	{100,200,500,1000}
SVM G	C (complexity)	{0.01, 0.1, 1, 10, 100}
	Gamma	{0.01, 0.1, 1, 10, 100}
XGB	Learning rate	{0.1, 0.2, 0.3, 0.4, 0.5}
	Nr. estimators	{100,200,500,1000}

Table 3: Parameter values tested through Grid Search for each evaluated classifier.

The predictive models were evaluated through Sensitivity, Specificity and AUC scores, which respectively determine the performance within positive and negative examples, and overall classification ability. The evaluation results are presented in Table 4.

All models provide results with AUC scores over 70% and decently-balanced Sensitivity and Specificity, which seems promising given the simple nature of the approach. While all algorithms performed quite similarly, Random Forest achieved the highest AUC in the majority of cases.

As for evaluating the effect of using time constraints, the metrics' values do not show any clear distinction. The Wilcoxon Signed-Ranks test for the AUC scores output p-values of 0.109, 0.109 and 1.000 for the time windows of 90, 180 and 365 days, respectively,

Time Window	Classifier	Sensitivity	Specificity	AUC
SPs by Prefixspan				
90d	RF	67.6 ± 6.6	70.3 ± 7.0	76.1 ± 4.4
	SVM	66.3 ± 6.6	71.7 ± 5.9	75.3 ± 4.0
	XGB	67.8 ± 5.9	72.3 ± 7.0	**76.3 ± 4.5**
180d	RF	81.6 ± 4.5	82.1 ± 5.2	**89.8 ± 3.2**
	SVM	76.9 ± 4.9	82.8 ± 4.9	88.4 ± 3.2
	XGB	79.3 ± 5.4	81.6 ± 5.8	88.9 ± 3.3
365d	RF	88.2 ± 3.3	87.1 ± 4.2	**94.2 ± 2.3**
	SVM	90.6 ± 3.3	83.8 ± 4.3	93.5 ± 2.1
	XGB	85.5 ± 5.0	86.2 ± 3.9	93.2 ± 2.0
SPs by Fournier08				
90d	RF	65.9 ± 5.8	71.5 ± 6.6	**75.6 ± 4.4**
	SVM	64.2 ± 5.5	72.3 ± 6.1	74.5 ± 3.8
	XGB	67.7 ± 5.6	70.6 ± 5.8	74.7 ± 4.2
180d	RF	79.7 ± 4.6	83.5 ± 4.4	**89.1 ± 3.1**
	SVM	85.5 ± 4.2	76.0 ± 5.2	88.0 ± 3.3
	XGB	79.2 ± 5.4	80.5 ± 5.3	88.0 ± 2.8
365d	RF	85.7 ± 4.6	88.8 ± 3.8	**93.9 ± 2.4**
	SVM	90.1 ± 2.9	84.3 ± 4.3	93.7 ± 2.1
	XGB	85.6 ± 4.8	86.6 ± 4.6	93.3 ± 2.1

Table 4: Model evaluation results obtained using class-labeled binary matrices as training sets, with a SMOTE balacing strategy and over a Stratified 5x10-fold cross-validation scheme. Each metric's result consists of the average over all folds, with uncertainty corresponding to the standard deviation. The best AUC for each dataset appears in bold.

GrowthRate (class)	Pattern
90 days	
1.93 (-)	<0> R = 12, CervicalExt = 5
1.93 (+)	<0> ALS-FRS = [24,36[, PhrenMeanAmpl < 0.4
1.80 (-)	UMN vs LMN = LMN, Onset form = spinal
1.78 (+)	<0> ALS-FRSb = [8,12[, ALS-FRSsUL = [8,12[
1.74 (+)	<0> ALS-FRSb = [8,12[, R = [8,12[
180 days	
2.71 (+)	<0> ALS-FRS = [24,36[, PhrenMeanAmpl < 0.4
2.06 (+)	<0> ALS-FRSb = [8,12[, R = [8,12[
1.92 (+)	<0> ALS-FRSb = [8,12[, ALS-FRSsUL = [8,12[
1.87 (-)	<0> ALS-FRSb = 12, ALS-FRSsLL = [8,12[, R = 12
1.86 (-)	<0> R = 12 | <3> R = 12
365 days	
3.45 (+)	<0> ALS-FRS = [24,36[, PhrenMeanAmpl < 0.4
2.06 (-)	UMN vs LMN = LMN, Onset form = spinal
2.04 (+)	<0> ALS-FRSb = [8,12[, R = [8,12[
2.00 (-)	<0> ALS-FRSb = 12, ALS-FRSsLL = [8,12[, R = 12
1.89 (-)	<0> ALS-FRSb = 12, R = 12 <1> ALS-FRSb = 12, R = 12

Table 5: Top 5 discriminative patterns for each prediction window, extracted with Fournier08. *<t>* denotes the time point. Patterns containing static features or a single time point are identical in both algorithms.

meaning there are no significant differences between the two approaches for significance thresholds of 0.1 or less. Therefore, sequential patterns extracted through Fournier08 would be preferrable to use, since they perform similarly with a smaller number of features (resulting in shorter running times), being more informative and easier to interpret.

4.2 Pattern Evaluation

For each prediction window, the extracted patterns were evaluated through GrowthRate, computed for both classes. Table 5 shows the 5 most discriminative patterns in the datasets generated through Fournier08, where the positive class corresponds to *Evolution = yes*.

When analysing the top 20 patterns for each class, we observed:

- In general, the discriminative patterns for the negative class have higher growth rates. 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;
- The pattern *ALS-FRS = [24,36[PhrenMeanAmpl < 0.4* is the most discriminative pattern for the positive class in all datasets, and the most discriminative overall for the larger time windows. With positive patterns being at a disadvantage, this is a clear standout;
- 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* throughout several evaluations. While *ALSFRSb = 12* only appears together with the respiratory subscore, lower values show up along with other items, meaning that bulbar function may 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. *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 many not be as relevant.

Additionally, we can further observe that the majority of patterns is considerably small, usually containing less than 5 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, with attributes such as *SNIP*, *MIP*, *MEP* not appearing at all. This is 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 CONCLUSIONS 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 spanning three prediction windows: 90, 180 and 365 days.

Model evaluation has shown promising results for all three prediction windows, with performance increasing with k. Both algorithms used to extract patterns presented similar results in model learning and pattern evaluation. With the selected support threshold, most of the retrieved sequential patterns contained only a few items and/or temporal instances, meaning there were few discrepancies between the two approaches. It can be noted that pattern selection can play a significant role in further study. The pattern extraction 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. 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 el al. [23, 24] can also be implemented, thus generating specific 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 many patterns being found. To minimize this issue, different prediction windows may also be explored according to progression rate, as more snapshots could be made eligible. Further, since class is determined by comparing the NIV status between evaluations, it's 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.

Acknowledgements. This work was partially supported by FCT funding to Neuroclinomics2 (PTDC/EEI-SII/1937/2014) and Predict (PTDC/CCI-CIF/29877/2017) research projects, and LASIGE Research Unit (UIDB/00408/2020).

REFERENCES

[1] [n.d.]. *Scalable and Flexible Gradient Boosting.* Retrieved Mar, 2020 from https://xgboost.ai

[2] R. Agrawal and R. Srikant. 1994. Fast Algorithms for Mining Association Rules. *Proc. 20th Int'l Conf. on Very Large Data Bases (VLDB)* (1994), 487–499. http://www.philippe-fournier-viger.com/spmf/apriori_longer.pdf

[3] R. Bhandari, A. Kuhad, and A. Kuhad. 2018. Edaravone: a new hope for deadly amyotrophic lateral sclerosis. *Drugs Today* 54, 6 (2018), 349–360. https://doi.org/10.1358/dot.2018.54.6.2828189

[4] R.H. Brown and A. Al-Chalabi. 2017. Amyotrophic Lateral Sclerosis. *N Engl J Med* 377 (2017), 162–172. https://doi.org/10.1056/NEJMra1603471

[5] A.V. Carreiro. 2016. *An integrative mining approach for prognostic prediction in neurodegenerative diseases.* Ph.D. Dissertation. Instituto Superior Técnico, Lisbon, Portugal. Advisor(s) S.C. Madeira.

[6] A.V. Carreiro, P.M.T. Amaral, S. Pinto, M. de Carvalho, and S.C. Madeira. 2015. Prognostic models based on patient snapshots and time windows: Predicting disease progression to assisted ventilation in Amyotrophic Lateral Sclerosis. *Journal of Biomedical Informatics* 58 (Dec. 2015), 133–144. https://doi.org/10.1016/j.jbi.2015.09.021

[7] A.V. Carreiro, S. Pinto, M. de Carvalho, S.C. Madeira, and C. Antunes. 2013. Classification of Clinical Data using Sequential Patterns: A case study in Amyotrophic Lateral Sclerosis. Conference Paper. *DMHH@SDM* (May 2013). https://www.researchgate.net/publication/236944617_Classification_of_Clinical_Data_using_Sequential_Patterns_A_case_study_in_Amyotrophic_Lateral_Sclerosis

[8] P. Fournier-Viger. 2019. *SPMF - An Open-Source Data Mining Library.* Retrieved Nov 18, 2019 from http://www.philippe-fournier-viger.com/spmf/

[9] P. Fournier-Viger, R. Nkambou, and E. M. Nguifo. 2008. A Knowledge Discovery Framework for Learning Task Models from User Interactions in Intelligent Tutoring Systems. In *MICAI 2008: Advances in Artificial Intelligence. Lecture Notes in Computer Science*, A Gelbhuk and E.F. Morales (Eds.). Vol. 5317. Springer, Berlin, Heidelberg, 765–778. https://doi.org/10.1007/978-3-540-88636-5_72

[10] L.I. Grad, G.A. Rouleau, J. Ravits, and N.R. Cashman. 2017. Clinical Spectrum of Amyotrophic Lateral Sclerosis (ALS). *Cold Spring Harb Perspect Med* 7 (2017). https://doi.org/10.1101/cshperspect.a024117

[11] O. Hardiman. 2011. Management of respiratory symptoms in ALS. *J Neurol* 258, 3 (March 2011), 359–365. https://doi.org/10.1007/s00415-010-5830-y

[12] Y. Hirate and H. Yamana. 2006. Generalized Sequential Pattern Mining with Item Intervals. *Journal of Computers* 1, 3 (June 2006). https://doi.org/10.4304/jcp.1.3.51-60

[13] M.C. Kiernan, S. Vucic, B.C. Cheah, M.R. Turner, A. Eisen, O. Hardiman, J.R. Burrell, and M.C. Zoing. 2011. Amyotrophic lateral sclerosis. *Lancet* 377 (March 2011), 942–955. https://doi.org/10.1016/S0140-6736(10)61156-7

[14] R. Kunert. 2017. *SMOTE explained for noobs - Synthetic Minority Over-sampling TEchnique line by line.* Retrieved Mar, 2020 from http://rikunert.com/SMOTE_explained

[15] S. Martin, A. Al Khleifat, and A. Al-Chalabi. 2017. What causes amyotrophic lateral sclerosis? [version 1; referees: 3 approved]. *F1000Research* 6(F1000 Faculty Rev), 371 (2017). https://doi.org/10.1136/jnnp-2018-319089

[16] J. Matos. 2019. *Biclustering Electronic Health Records to Unravel Disease Presentation Patterns.* Master's thesis. Faculty of Sciences, University of Lisbon.

[17] P.R. Mehta, A.R. Jones, and S. Opie-Martin *et al.* 2019. Younger age of onset in familial amyotrophic lateral sclerosis is a result of pathogenic gene variants, rather than ascertainment bias. *J Neurol Neurosurg Psychiatry* 90 (2019), 268–271. https://doi.org/10.1136/jnnp-2018-319089

[18] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. 1999. Discovering Frequent Closed Itemsets for Association Rules. *Database Theory - ICDT'99* (1999), 398–416. http://www.philippe-fournier-viger.com/spmf/pasquier99pdf

[19] J. Pei and J. Han et al. 2004. Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. *IEEE Transctions on Knowledge and Data Engineering* 16, 10 (Oct. 2004). http://www.philippe-fournier-viger.com/spmf/prefixspan.pdf

[20] G. Pfeffer, M. Povitz, G.J. Gibson, and P.F. Chinnery. 2015. Diagnosis of muscle diseases presenting with early respiratory failure. *J Neurol* 262 (2015), 1101–1114. https://doi.org/10.1007/s00415-014-7526-1

[21] S. Piepers and L.H. van den Berg. 2006. Evidence-based care in amyotrophic lateral sclerosis. *Lancet Neurol* 5, 2 (Feb. 2006), 105–106. https://doi.org/10.1016/s1474-4422(06)70330-1

[22] S. Pires. 2018. *A Supervised Learning Approach for Prognostic Prediction in ALS using Disease Progression Groups and Patient Profiles.* Master's thesis. Faculty of Sciences, University of Lisbon.

[23] Sofia Pires, Marta Gromicho, Susana Pinto, Mamede Carvalho, and Sara C Madeira. 2018. Predicting Non-invasive Ventilation in ALS Patients Using Stratified Disease Progression Groups. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW).* IEEE, 748–757.

[24] Sofia Pires, Marta Gromicho, Susana Pinto, Mamede de Carvalho, and Sara C Madeira. 2020. Patient Stratification Using Clinical and Patient Profiles: Targeting Personalized Prognostic Prediction in ALS. In *International Work-Conference on Bioinformatics and Biomedical Engineering.* Springer, 529–541.

[25] A.G. Solimando, S. Ciccol, G. Di Lernia, P. Leonel, and E. D'Errico *et al.* 2018. Move Toward to Non Invasive Mechanical Ventilation in Amyotrophic Lateral Sclerosis: A Clinical Review. *J Clin Genom* 1, 1 (2018). https://doi.org/10.1016/j.neurol.2017.03.029

[26] C. Tard, L. Defebvre, C. Moreau, D. Devos, and V. Danel-Brunaud. 2017. Clinical features of amyotrophic lateral sclerosis and their prognostic value. *Revue Neurologique* 173 (May 2017), 263–272. https://doi.org/10.1016/j.neurol.2017.03.029

[27] M.A. van Es, O. Hardiman, A. Chiò, A. Al-Chalabi, R.J. Pasterkamp, J.H. Veldink, and L.H. van den Berg. 2017. Amyotrophic lateral sclerosis. *Lancet* 390 (Nov. 2017), 2084–2098. https://doi.org/10.1016/S0140-6736(17)31287-4

[28] S. Vucic, J.D. Rothstein, and M.C. Kiernan. 2014. Advances in treating amyotrophic lateral sclerosis: insights from pathophysiological studies. *Trends in Neurociences* 37, 8 (Aug. 2014). https://dx.doi.org/10.1016/j.tins.2014.05.006

[29] J. Wang and J. Han. 2004. BIDE: Efficient Mining of Frequent Closed Sequences. *Proceedings. 20th International Conference on Data Engineering, Boston, MA, USA* (2004), 79–90. https://doi.org/10.1109/ICDE.2004.1319986

[30] M.J. Zaki and W. Meira Jr. 2014. *Data Mining and Analysis: Fundamental Concepts and Algorithms* (1st. ed.). Cambridge University Press.

[31] S. Zarei, K. Carr, L. Reiley, K. Diaz, O. Guerra, P.F. Altamirano, W. Pagani, D. Lodin, G. Orozco, and A. Chinea. 2015. A comprehensive review of amyotrophic lateral sclerosis. *Surg Neurol Int* 6, 171 (Nov. 2015). https://doi.org/10.3978/j.issn.2305-5839.2016.01.16