Temporal modeling of deterioration patterns and clustering for disease prediction of ALS patients

Dan Halbersberg
Ben Gurion University of the Negev
Be'er Sheva, Israel
halbersb@post.bgu.ac.il

Boaz Lerner
Ben Gurion University of the Negev
Be'er Sheva, Israel
boaz@bgu.ac.il

***Abstract*—Amyotrophic lateral sclerosis (ALS) is a neurodegenerative disease, lasting from the day of onset until death. Factors such as the progression rate and pattern of the disease vary greatly among patients, making it difficult to achieve accurate predictions about ALS. To accurately predict ALS disease state and deterioration, we propose a novel approach that combines: a) sequence clustering based on dynamic time warping for separation among patients with diverse ALS deterioration patterns, b) sequential pattern mining for discovery of deterioration changes that patients of the same type may have in common, and c) deterioration-based patient next-state prediction. Using a clinical dataset, we demonstrate the advantage of the proposed approach in terms of classification accuracy and deterioration detection compared to other classification methods and temporal models such as long short-term memory.**

***Index Terms*—Amyotrophic lateral sclerosis (ALS), prediction, deterioration, temporal models, sequential pattern mining, sequence clustering, classification**

I. Introduction

Amyotrophic lateral sclerosis (ALS) is a fatal neurodegenerative disease of unknown origin affecting the human motor system with a highly uncertain pathogenesis [13]. The inner workings and mechanisms of this disease are mainly unknown [4], but recent progress aims at better understanding of ALS pathogenesis to enable extension of life expectancy and improvement in the life quality of those afflicted.

The medical condition of an ALS patient is evaluated at clinic visits using ten items describing physical functionalities [3] such as speaking, writing, or walking. Each is given a grade from 0 for complete loss of function to 4 for full functionality. The sum over these items, called the ALS functional rating scale (ALSFRS), represents the patient's total functionality having a value between 0 and 40 [3]. Measuring the ALSFRS at each clinic visit helps in understanding the patient's medical condition, tracks his disease deterioration, and improves the assessment of the influence of treatment.

Our research concentrates on two questions: 1) can temporal modeling significantly improve prediction of the next patient state? and 2) can sequence clustering and/or sequential pattern mining improve the prediction of the disease next state?

This study develops an approach to: cluster ALS patients based on their deterioration patterns, find significant patterns of ALS deterioration in sequential data of patients for each grade of functionality, and predict a patient's ALS state based on his previous states and those of patients whose deterioration pattern is similar. It proposes a new prediction framework that combines: a) sequence clustering based on dynamic time warping for separation among patients with diverse ALS deterioration patterns, b) sequential pattern mining for discovery of frequent deterioration changes that patients of the same type may have in common, and c) frequent deterioration-based disease-state prediction.

The motivation for this approach is to discover deterioration patterns. As opposed to regular sequence pattern mining, we do not search for the patterns over the row data, but over deteriorations in ALS functions. To the best of our knowledge, this has never been done before. Demonstrated for ALS, the proposed framework is extended to clinical data of other diseases such as Parkinson's and Alzheimer's.

Section II provides background and a related-work survey. Section III describes the proposed framework, while Section IV outlines the framework's evaluation criteria. Section V reports the results of demonstrating the framework for a clinical dataset, and Section VI summarizes and discusses limitations and future work.

II. Background and related work

This section outlines sequential and temporal modeling, in general, and ALS prediction and progression rate modeling, in particular. In sequential (longitudinal) data, as opposed to classical time-series data, the collection of observations does not have to be taken at successive equally spaced points in time, and only the order within the sequence is important. In addition, the sequence usually has a lower dimensionality with fewer observations in each sequence [7].

A. Modeling ALS progression

Numerous studies have dealt with the task of predicting the state or deterioration rate of ALS and, just until very recently, most of them used non-temporal methods such as regression, decision tree, and random forest (RF). [8] developed multiple models for different progression rates based on clustering patients into groups of fast and slow progression based on the difference in ALSFRS values between the first and last visit divided by the time between them. After a new patient was assigned to a cluster, they applied the relevant Weibull model. In another non-temporal approach, [10] used RF to predict ALS progression from the fourth to the twelfth month

based on data of the first three months. In addition to feeding the RF with fixed data about static variables per patient, like gender and site of onset, as opposed to temporal variables that can change over time, they fed the RF with the slope between the first visit and the last known visit.

Following this line, [20] used slopes together with baseline (first visit) data from the first visit only to develop separate RF models for long-term and short-term predictions, generating two different classifiers. They concluded that short-term predictions are consistent with those of other models like RF or the generalized linear model, but long-term predictions of other models tend to overfit, while the RF method does not. They found that the four most important variables are the time of prediction since baseline, the time from symptom onset to baseline, the ALSFRS score at baseline, and the slope of the ALSFRS score at baseline. Nevertheless, when using non-temporal models with longitudinal data, it is neither possible nor easy to exploit the data from all time points and, usually, some data are thrown away, aggregated, or partly used, and a linearity assumption between visits is made. Therefore, it is not recommended to use non-temporal models in such cases [17].

B. Sequence clustering

When it comes to sequence analysis, one important task is sequence clustering, in which sequences that are somehow related are grouped together. In ALS, separating between patients based on the sequences of states is important to stakeholders, such as physicians and drug companies, who wish to select small homogeneous groups of patients for clinical trials. Moreover, sequence clustering in ALS is also important for prediction since each group of patients may represent a different deterioration pattern and, hence, perhaps should be treated separately.

When sequences are of the same length, the simple k-means algorithm for sequence clustering can be applied. However, this is not the case when each sequence has its own length. For example, different patients may have a different number of visits. One way to bring all sequences to the same length according to [7] is to apply a dimensionality reduction method like piecewise aggregate approximation (PAA), where the average value of each w-sized window is calculated to reduce the dimension to w. [15] proposed the symbolic aggregate approximation (SAX) method that converts PAA to a symbol string by mapping all PAA values below the smallest breakpoint to symbol "a", all values between the smallest and second smallest breakpoints to symbol "b", etc. Nevertheless, the user must specify in advance the cardinality of the symbols, i.e., the size of the alphabet.

Once the sequences are transformed to a new representation of equal length, a simple Euclidean distance or MINDIST can be used to cluster them respectively via partition or hierarchical clustering [15]. If the sequences are not too long (e.g., in ALS, they have less than 10 visits), then SAX can skip the PAA step directly to discretization and symbolic representation of the sequences. Interestingly, [15] showed that working with the symbolic approximation produces better time-series clustering results than working with the original data despite the need for an extra user-defined parameter (symbol cardinality).

Another approach is dynamic time wrapping (DTW) that measures the similarity between two temporal sequences. It has long been used for speech processing as a method that allows an elastic shifting of the time axis to accommodate sequences that are similar but out of phase [11]. Let Q and C be two time series of lengths n and m, respectively, where $Q = \{q_1, q_2, ..., q_i, ..., q_n\}$ and $C = \{c_1, c_2, ..., c_j, ..., c_m\}$. To align the two sequences using DTW, an n-by-m matrix is introduced, where the (i, j) element of the matrix contains the distance between the two points q_i and c_j:

$$d(q_i, c_j) = (q_i - c_j)^2. \quad (1)$$

A warping path W is a contiguous set of matrix elements that defines a mapping between Q and C [12]. The warping path is typically subject to several constraints: a) the corners where the path begins and ends, b) continuity of the path (adjacent cells), and c) the points in W that have to be monotonically spaced in time.

DTW can also be extended for multi-dimensional data, where each sequence is not just a single signal, but at each time point, each sequence consists of multiple measurements [18] (e.g., a patient is represented by a sequence of visits in the clinic, where in each visit, several indicators/tests were taken). Mainly, there are two ways DTW can be generalized to the multi-dimensional case: dependent or independent warping. In independent warping, the distance between two multi-dimensional time series is the sum of distances over each dimension separately. In dependent warping, where we assume mutual dependence between all dimensions, the distance is redefined as the cumulative squared Euclidean distances of M data points instead of the single data point used in the one-dimensional case (Eq. 1): $d(q_i, c_j) = \sum_{m=1}^{M}(q_{i,m} - c_{j,m})^2$.

C. Sequential mining

While in supervised sequential mining such as long short-term memory (LSTM), the task is to learn rules that separate between two or more classes [6], unsupervised sequential mining, also called sequential pattern mining [14], is more difficult [14], and our proposed framework is such.

1) Sequential pattern mining: Sequential pattern mining discovers frequent sub-sequences as patterns in a sequence database. It is a sub-class of frequent pattern mining that was first introduced by [1]. In this mining, elements or events are ordered, but without a concrete notion of time.

Let $I = \{i_1, i_2, ..., i_k\}$ be a set of all items. A subset of I is called an itemset. A sequence $\alpha =< a_1, a_2, ..., a_m > (a_i \subseteq I)$ is an ordered list of itemsets. Each itemset in a sequence represents a set of events occurring at the same timestamp (e.g., a_2 can be $< i_2, i_4 >$), while different itemsets in a sequence occur at different times. For example, a patient visit sequence could have data from several test results in one visit followed by other test results or different examinations, and so on. A sequence $\alpha =< a_1, a_2, ..., a_m >$ is a sub-sequence of

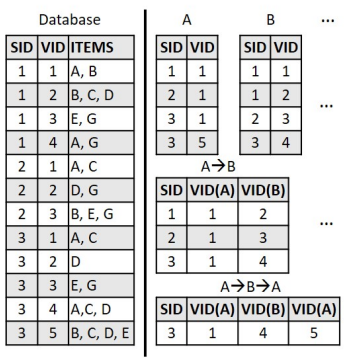

Fig. 1: Example of temporal joining according to SPADE.

another sequence $\beta =< b_1, b_2, ..., b_n >$ denoted as $\alpha \subseteq \beta$ if and only if $\exists\ i_1, i_2, ..., i_m$ such that $1 \leq i_1 \leq i_2 \leq ... \leq i_m \leq n$ and $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}$,..., and $a_m \subseteq b_{i_m}$. We also call β a super-sequence of α, and β contains α. Given a sequence database D, the support (Sup) of a sequence α is the number of sequences in D that contain α. If the support of a sequence α satisfies a pre-specified minimal support threshold, α is a frequent sequential pattern.

Among the most common algorithms that find frequent sequential patterns are the two a-priori-based algorithms: the generalized sequential pattern (GSP) [19] that uses the downward-closure property of sequential patterns (i.e., if a set is frequent, then so are all its subsets) and adopts a multiple pass (candidate generate-and-test) approach; and the sequential pattern discovery using equivalent class (SPADE) [21], which is an extension of vertical format-based frequent itemset mining methods. Figure 1 shows how SPADE creates sequential patterns. The left-hand side of the figure is the sequential database, where SID is the sequence ID (e.g., patient ID), VID is the visit ID, and ITEMS are the events associated with the visit (e.g., A and B are decreases in a patient's speech and salivation capabilities, respectively). The right-hand side of Figure 1 demonstrates how SPADE grows sequence patterns by joining subsequences. For example, pattern $A \rightarrow B$, i.e., a visit that includes event A followed by a visit that includes event B (top of figure), is a joining of Tables A and B (middle of figure) for the cases in which a patient has both events A and B, $SID(A) = SID(B)$, and event A occurred before event B, $VID(A) < VID(B)$. Similar, $A \rightarrow B \rightarrow A$ (bottom of figure) is the joining of A to $A \rightarrow B$. The support of the pattern is the number of rows that result from the joining (e.g., $Sup(A \rightarrow B) = 3$). This method banishes the need to rescan the database multiple times.

2) Sequential supervised learning: Statistical learning problems in most medical applications such as ALS involve sequential supervised data in which the sequences consist of (x, y) pairs and not just x (as in sequential pattern mining). In sequential supervised data, samples (pairs in the sequences) are not independent, i.e., nearby (x, y) values are likely to be related to each other [6]. The sequential supervised learning problem can be formulated as follows: let $\{(\vec{x}_i, \vec{y}_i)\}_{i=1}^{N}$ be a set of N training samples (we use arrows above variables to denote vectors). Each sample is a pair of sequence $\vec{x}_i, \vec{y}_i)$, where $\vec{x}_i =< x_{i,1}, x_{i,2}, ..., x_{i,T_i} >$, $\vec{y}_i =< y_{i,1}, y_{i,2}, ..., y_{i,T_i} >$, and T_i is the length of sequence i. For example, in ALS data, $y_{i,j}$ and $x_{i,j}$ represent the label and the predictor of the j visit of patient i in the clinic, respectively. Note that, usually, $y_{i,j}$ is a scalar (whether a discrete or a real number), while $x_{i,j}$ can be a vector of predictors (i.e., $\vec{x}_{i,j}$).

One family of algorithms that does sequential supervised learning are the graphical models. The most common graphical model that handles temporal data is the hidden Markov model (HMM) and its Bayesian representation, the dynamic Bayesian network (DBN) [16], which extends the Bayesian network to model state-space data. However, both HMM and DBN assume that all variables are discrete, and that the time interval between observations is equal (i.e., state-space models). Another temporal model that falls under the definition of a graphical model is the LSTM [9]. An LSTM unit is a recurrent network unit that excels at remembering values for either long or short durations of time. The key to this ability is that it uses no activation function within its recurrent components. Thus, the stored value is not iteratively squashed over time, and the gradient term does not tend to vanish when back propagation through time is applied to train it.

III. Method

In this paper, we suggest to predict the patient's next state using a framework that consists of three stages; first, a sequence clustering based on an independent multi-dimensional DTW, aiming to group patients with similar sequences; second, to mine for common deterioration patterns that describe similar patients; and third, to train a classifier based on cluster-specific patient common patterns.

First, we randomly split the data into training (80%) and testing (20%) sequences (patients). In the first stage of the framework, each two training sequences are compared based on a multi-dimensional DTW, and a similarity matrix ($N \cdot N$) is constructed, where N is the number of training sequences and each entry (i, j) is the DTW distance between sequences i and j. This similarity matrix is used for growing a hierarchical clustering (by complete linkage). We select the number of clusters (K) based on the dendrogram so that each cluster will have a minimal number of sequences (patients) (see Section V-C below). The output of this stage is K clusters (groups), each of which contains similar sequences.

In the second stage of our proposed framework, a sequential pattern mining algorithm (SPADE) [21] is used to find patterns in the training data. Since we are interested in finding patterns that describe deterioration patterns, we convert our dataset into an event-deterioration sequence database as follows: Initially, we define an event dictionary in which each of ten events B-K is defined as a decrease in a specific ALS function, e.g., B is a decrease in Speech. Event A represents that the patient's state remains completely the same (i.e., all ALS functions remain unchanged in the visit). This dictionary is then used to transform each multi-dimensional sequence (Table I) into

TABLE I: An example of multivariate patient data with 11 visits (rows) and 10 ALS functions (columns). The blue font represents a spurious deterioration (which should be ignored), and the red font represents a real deterioration (supported by the further visits).

Visit number	Speech	Salivation	Swallowing	Writing	Cutting food	Dressing	Turning in bed	Walking	Climbing stairs	Respiratory	Time from onset
1	4	4	4	4	4	3	3	2	1	4	0
2	4	4	4	4	4	2	3	2	1	4	90
3	4	3	4	4	4	4	3	2	1	4	180
4	4	4	4	4	4	3	3	2	1	4	240
5	4	4	4	4	3	3	3	2	1	4	360
6	4	4	4	4	4	3	2	2	0	4	450
7	4	3	3	3	1	1	2	1	0	4	630
8	4	3	3	3	1	1	2	1	0	4	780
9	3	3	3	2	1	0	1	1	0	4	900
10	3	3	3	2	1	0	1	1	0	4	990
11	3	2	2	2	1	0	1	1	0	4	1080

an event-sequence representation (Table II). The last visit (11 in the case of Table I) is not used for pattern mining since it will be used later for the prediction of the disease next state.

TABLE II: The example in Table I represented using the dictionary.

Visit number	Items	Number of items
2	A	1
3	A	1
4	G	1
5	A	1
6	H,J	2
7	C,D,E,F,G,I	6
8	A	1
9	B,E,G,H	4
10	A	1

Next, we apply PAA and SAX [15] to reduce the target variable cardinality/dimension (we consider the ALSFRS variable that ranges from 0 to 40 as a continuous variable that needs discretization). Discretization can help us find deterioration patterns that are unique to each class. To achieve this, we apply SPADE to each class separately over the event-deterioration sequence database. Other reasons to address each class separately are: 1) there is a higher chance that the pattern variables will separate classes, and 2) otherwise, unique patterns of the minority class will not have enough support. The output of the second stage is frequent deterioration patterns per class (each class is a state of the discretized target variable).

In the third stage, we train a classifier for each cluster separately, where patients in the cluster are each represented by a single record that includes all deterioration patterns as binary variables (whether the pattern exists or not in the patient's multi-dimensional sequence). Later, in the test, we first assign a test patient to his most similar cluster, and then predict his disease state using the cluster's classifier. Although transforming a patient's data into a single record may involve a loss of information, we expect the moments we extract (mean, slopes, etc.), together with the binary pattern features, to capture most of the essential information.

IV. Evaluation

We use the *PROACT ALS dataset* [2], collected by the non-profit organization Prize4life from 17 late-stage industry and academic clinical trials. It contains 3,171 patients with 22,089 clinic visits, 42 variables per visit, and an average time between visits of 132 days. A visit consists of patients' static data (e.g., gender, age at onset), which do not change over time, temporal/longitudinal data that include laboratory test results (e.g., bilirubin, glucose), and ten observed variables that measure patients' functionality (e.g., walking, writing). Each of these ALS functions takes a value between 0 and 4, and their sum (ALSFRS), our target variable (for classification), takes values between 0 and 40. Although the time between patient visits is not constant, in most cases, it ranges from three to six months. Since life expectancy with ALS is relatively short (3–5 years), a patient will have on average only seven visits.

A. Evaluation metric

Evaluation of ALS predictions usually relies on the mean absolute error (MAE), which gives a good sense of how far the predictions are in terms of points from the ALSFRS score. In this paper, we report on MAE, but also on the accuracy and F1 measures for predicting the last recorded ALSFRS (i.e., last visit). Note, that these predictions were made for the separate testing set (20%). Moreover, since a model that predicts that a patient's next state is equal to the previous one will have high accuracy (due to the imbalance property, where in most cases, successive ALS functions have the same value), we also measure our algorithm's ability to detect a decrease in ALSFRS (a binary variable). Thus, we propose to measure the performance for two target variables separately. To make our results more reliable, we repeat each experiment five times (i.e., five folds) with a random split to train and test sets, and report on the average results.

B. Research questions and experimental design

Recall that our research questions are: 1) can temporal modeling significantly improve prediction of the next patient state? and 2) can sequence clustering and/or sequential pattern mining improve the prediction of the disease next state?

To answer these questions, we report here on a series of experiments designed to measure the impact of each of our developed elements (i.e., clustering and pattern mining) separately and together. In the first experiment (used as a

reference/baseline), we train a state-of-the-art classifier (RF) on the PROACT dataset (without clustering, pattern mining, or any other temporal modeling technique). We denote it henceforth as the naïve approach (Experiment 1). The input for this classifier, that aims to predict the ALSFRS at the last visit, and whether that variable has decreased from the previous visit is all the information from the previous visit (all ten ALS functions, the ALSFRS, and the time from onset) and the slopes which are the differences between the ALS functions of the previous visit and the first visit divided by the corresponding time intervals. The motivation for this is that the slopes represent the trend of each ALS function, and the last (previous visit) is expected to be the most correlated to the target variable since it is the closest in terms of the time interval. If the naïve classifier is a regression model, then the model is in the form of:

$$\begin{aligned} y_{i,T} =& \beta_1 \cdot y_{i,T-1} + \beta_2 \cdot (y_{i,T-1} - y_{i,1}) \quad + \\ & \vec{\beta}_3 \cdot \vec{x}_{i,T} + \vec{\beta}_4 \cdot (\vec{x}_{i,T-1} - \vec{x}_{i,1}) + \beta_5 \cdot t, \end{aligned}$$

where $y_{i,T}$ is the ALSFRS of patient i in the last visit (T), β_1 is the coefficient of the last known ALSFRS (in T-1), β_2 is the coefficient of the slope between the previous ALSFRS and the first one, $\vec{\beta_3}$ are the coefficients (vector) of all predictors (all 10 ALS functions) of the last known (previous) visit, $\vec{\beta_4}$ are the coefficients of all the slopes (similar to β_2) of all predictors (all 10 ALS functions), and β_5 is the coefficient of the $time\_from\_onset$ variable for which the prediction is needed.

Table III shows an example feature vector for the patient in Table I. The slope of Speech is $1/990$ since Speech deteriorates from a value 4 (in the 1st visit) to a value 3 (in the 10th visit) in 990 days. The "Last decreased" variable takes the value "No" because the ALSFRS did not decrease between the 9th and 10th visits, which are distanced $t = 90$ days apart, and the target variable "Is decreased" also takes the value "No" because there was no decrease in Speech between the 10th and 11th visits). We call this process $flattening$.

The second experiment includes sequential pattern mining based on SPADE (Section III), applied to each class separately to detect patterns that are unique to that class. All patterns, from all classes, that are above the minimal support (Section II-C) are transformed into a set of binary variables, each of which indicates if a pattern exists in patient i's data (although we search for patterns only in the training data, the qualified patterns are post-detected in both training and test patients regardless of the class, i.e., we search for a pattern that was detected for class c also in data of class $j \neq c$). Next, this new set of binary variables (a new feature vector per patient) is appended to the feature vector from Experiment 1, creating a new expanded feature vector per patient (for both training and test data). Similar to Experiment 1, we train a classifier based on this new derived data that includes pattern information, allowing evaluation of the contribution of sequential pattern mining to the prediction of the next patient state.

The third experiment includes grouping patients using hierarchical clustering based on DTW. Once the training patients are divided into clusters, we train a classifier for each cluster separately (Section III) based on the feature vectors created for Experiment 2 per patient. The question that remains is how to assign a testing patient to the most probable cluster. In partition clustering methods (e.g., k-means), it is easy to assign a new test record (patient) to a cluster based on the minimal distance to all cluster centers. This is more challenging when it comes to hierarchical clustering, as there are no clear definitions for the centers (each patient has a different sequence length). To overcome this, we propose a method where each testing patient is assigned to a cluster based on the minimal average distance to all of the cluster's (training) patients. For that, we denote ns_c as the number of sequences (patients) in cluster c. A new testing sequence P_j of patient j is assigned to cluster c_j according to,

$$c_j \in \arg\min_c \frac{1}{ns_c} \sum_{i=1}^{ns_c} DTW(P_j, P_i^c),$$

where P_i^c is the i^{th} patient (sequence) in cluster c.

In the final experiment (Experiment 4), we also apply the state-of-the-art temporal model LSTM that in contrast to our proposed method, which aggregates data and thus may lose vital information, uses all the patient's data and thus presumably does not need any pre-processing stages (i.e., our pattern mining and clustering).

V. Results

In the following experiments, the target variable (ALSFRS) was discretized using SAX into five ordinal states/classes, A–E (see Section III). In order for the sequential pattern mining to be effective, we filtered out patients with less than four visits to the clinic (so we were left with $2,590$ patients). Also, in all the experiments, we used RF with 500 trees as our classifier. To adjust the classifier to treat the first target variable (the five classes of ALSFRS) as an ordinal response, we applied a cost matrix, where each entry in the matrix equals the error size $|j - i|$. In addition, due to the imbalance nature of the second target variable, i.e., $is\_decreased$ (a patient is not likely to change ALSFRS values between two successive visits), we applied another cost matrix for the binary response with a 20:1 ratio. The values in the cost matrix were selected experimentally.

A. Experiment 1–Naïve classifier

Table IV shows the average confusion matrix (CM) over five folds (as described in Section IV-A) for the binary target variable. It can be seen from Table IV that the class imbalance is more than 1:4, and thus that the RF model makes more accurate predictions for the majority class (no decrease).

B. Experiment 2–Pattern mining

In this experiment, we added the pattern variables (in addition to the first experiment's variables). We experimentally selected $min\_sup = 0.4$ (a user-defined parameter) for SPADE and filtered out patterns with less than two items

TABLE III: An example of a feature vector for the naïve approach derived from the patient presented in Table I, where green, blue, black, and red represent slope variables, data from the previous visit, general variables, and target variables, respectively.

Slope				Previous visit				General variables		Target 1	Target 2
Speech	...	Dyspnea	ALSFRS	Speech	...	Dyspnea	ALSFRS	Last decreased	t	ALSFRS	Is decreased
1/990		3/990	2/990	3		1	2	No	90	1	No

TABLE IV: RF avg CM for Exp. 1 and the binary target variable.

		True Class (Y)	
Predicted Class (X)		not decreased	decreased
	not decreased	380.6	112.9
	decreased	11.8	12.7
		97%	**10%**

TABLE V: RF avg CM for Exp. 2 and the binary target variable.

		True Class (Y)	
Predicted Class (X)		not decreased	decreased
	not decreased	376.4	106.8
	decreased	16	18.8
		96%	**15%**

(patterns with a single item are not informative). Table V shows the average CM over five folds.

It can be seen from Table V that although the total accuracy remains almost the same (~76%), the accuracy in classifying the minor class was increased by 50% (from 10% to 15%). This can be explained by the fact that we searched for patterns per class including those that are unique for the minority class.

C. Experiment 3–Clustering

This experiment represents our proposed framework since it includes all three elements (clustering, pattern mining, and classification). We added here the pre-processing step of hierarchical clustering (we selected $4 \leq K \leq 6$ based on the resultant dendrogram in each data fold, so that each cluster would have at least 80 sequences, i.e., patients) and used the variables of Experiment 2 in each cluster. Following [18]'s recommendation, we tried both dependent and independent multi-dimensional DTW, and found the independent method superior for our dataset. Thus, we report on results based on that method. Tables VI and VII show the average CMs for binary and multi-class target variables, respectively, over five folds and four–six clusters (a weighted average since the cluster sizes are different). We could have improved the prediction accuracy of the "decreased" class in Table VI by manipulating the cost matrix, but this will come at the expense of the majority class, in which we succeeded in keeping the false alarm rate between 2%–5% (Tables IV, V, VI, and VIII). For example, if we change the ratio in the binary cost matrix from 20 to 5 then the minority class accuracy would increase to ~50% at the expense of decreasing the majority class accuracy to 81%.

Once again, it can be seen from Table VI that the accuracy of the minor class was increased, but this time less drastically (15% vs. 18%). Table VII reveals that most errors are "mild" in terms of error severity (e.g., predicting A instead of B or vice versa), which is an advantage of our approach.

D. Experiment 4–LSTM

In this final experiment, we compared our proposed framework to a standard temporal model, the LSTM (we used two hidden layers with 200 neurons in each layer). Table VIII shows that the LSTM classifier has better accuracy than the naïve RF classifier (Experiment 1), but at the expense of being the worst in predicting the minority class (7%).

Next, we summarize in Table IX the results from all experiments and for both target variables.

Table IX reveals that, for both binary and multi-class responses, the results improve as we move from Experiments 1 to 3 (recall that each of these experiments adds another layer to the prediction framework). Although the differences with respect to accuracy, F1, and MAE are not statistically significant, a non-parametric Wilcoxson signed rank test [5] (with a 0.05 confidence level) shows that our proposed framework (Experiment 3) is superior to the naïve and LSTM classifiers with respect to minor-class accuracy. The average accuracy improvement of our proposed framework over the baseline naïve classifier (with respect to the minor-class accuracy) was ~80% (10.11% vs. 17.57%). Moreover, albeit the LSTM is usually ranked above the naïve classifier, it is almost always inferior to our enhanced frameworks (pattern mining and clustering) with respect to all measures.

Following is a list of the ten most important predictors measured based on the Gini impurity (starting from the most significant): Previous ALSFRS, Previous Dressing, Previous Climbing stairs, Previous Turning in bed, t, Pattern $< E, G, I >$ (i.e., decline in writing capability followed by decline in dressing, and finally in walking), Slope Salivation, Age, Pattern $< F, J >$, and Pattern $< J, B >$. Not surprisingly, the top four variables are from the previous visit, but encouraging is the fact that a few pattern variables are also among the important variables, which is further evidence for

TABLE VI: RF avg CM for Exp. 3 and the binary target variable.

		True Class (Y)	
Predicted Class (X)		not decreased	decreased
	not decreased	374.3	103.5
	decreased	18.1	22.1
		95%	**18%**

TABLE VII: RF avg CM for Exp. 3 and the ALSFRS target variable.

		True Class (Y)				
Predicted Class (X)		A	B	C	D	E
	A	210.4	33.6	3.2	0.7	0
	B	17.8	60.9	29.7	4.4	0
	C	0.5	7.2	13.5	6.1	0
	D	1.2	6	13.6	64.4	8.1
	E	0	0.3	0	7.3	29.1

TABLE VIII: LSTM avg CM for Exp. 4 and the binary target variable.

		True Class (Y)	
		not decreased	decreased
Predicted Class (X)	not decreased	385.1	116.9
	decreased	7.3	8.7
		98%	**7%**

TABLE IX: Accuracy, F1, and MAE for four experiments and the two target variables.

	Experiment	Accuracy	F1	MAE	Minor-class accuracy
Binary	1 - naïve classifier	75.92	**0.86**	0.24	10.11
	2 - pattern mining	76.29	**0.86**	**0.23**	14.97
	3 - clustering	**76.53**	**0.86**	**0.23**	**17.56**
	4 - LSTM	76.02	**0.86**	0.24	6.93
Multi-class	1 - naïve classifier	71.98	0.66	0.31	-
	2 - pattern mining	72.87	0.67	**0.30**	-
	3 - clustering	**73.03**	**0.68**	**0.30**	-
	4 - LSTM	72.97	0.65	0.32	-

the contribution of sequential pattern mining to our prediction framework (as was already demonstrated in Tables IV– IX).

VI. Discussion and future work

Prediction of the next ALS patient state is difficult because the life expectancy of patients is relatively short (so only a few records per patient are available), ALS is highly heterogeneous (thus, different patients have various deterioration patterns), and as a rare disease, it has relatively small datasets (compared to other diseases).

We have suggested a new prediction approach that exploits sequence clustering and sequential pattern mining to better predict the next patient state. As opposed to regular sequence pattern mining, we do not search for the patterns over the row data, but over deteriorations in ALS functions. This, to the best of our knowledge, has never been done before, and the motivation for this is to discover deterioration patterns. Note however that the proposed approach is general and can fit any temporal data that have a target variable (i.e., sequential supervised learning), and any event of either an increase (improvement) or decrease (deterioration) nature. The results show that the contribution of both sequence clustering and sequential pattern mining are positive. In addition, the results show that our proposed framework does not fall behind dedicated temporal models such as the LSTM.

Future work can concentrate on several aspects: 1) While here, we focused on predicting one state ahead and only on deteriorations, an extension can be made to predict two or more periods ahead and also improvement in a patient's functionality, respectively; 2) Since the number of ALS patients is relatively low, shorter sequences can be derived from longer ones, but this should be done carefully to avoid bias as patients with a long sequence may be over-represented; 3) More temporal data (e.g., laboratory tests) can be added to the sequential pattern mining stage besides the ALS functions; and 4) The approach can be applied to other diseases.

References

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the 11th International Conference on Data Engineering*, pages 3–14. IEEE, 1995.

[2] N. Atassi, J. Berry, A. Shui, N. Zach, A. Sherman, E. Sinani, J. Walker, I. Katsovskiy, D. Schoenfeld, M. Cudkowicz, et al. The pro-act database design, initial analyses, and predictive features. *Neurology*, 83(19):1719–1725, 2014.

[3] B.R. Brooks, M. Sanjak, S. Ringel, J. England, J. Brinkmann, A. Pestronk, J. Florence, H. Mitsumoto, K. Szirony, J. Wittes, et al. The amyotrophic lateral sclerosis functional rating scale-assessment of activities of daily living in patients with amyotrophic lateral sclerosis. *Archives of Neurology*, 53(2):141–147, 1996.

[4] R.H. Brown and A. Al-Chalabi. Amyotrophic lateral sclerosis. *New England Journal of Medicine*, 377(2):162–172, 2017.

[5] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[6] T.G. Dietterich. Machine learning for sequential data: A review. In *Joint International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition*, pages 15–30. Springer, 2002.

[7] T.C. Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.

[8] R. Gomeni, M. Fava, and Pooled R. Amyotrophic lateral sclerosis disease progression model. *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration*, 15(1-2):119–129, 2014.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[10] T. Hothorn and H.H. Jung. Randomforest4life: A random forest for predicting ALS disease progression. *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration*, 15(5-6):444–452, 2014.

[11] F. Itakura. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):67–72, 1975.

[12] E. Keogh and C.A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.

[13] M. C Kiernan, S. Vucic, B. C Cheah, M. R Turner, A. Eisen, O. Hardiman, J.R. Burrell, and M.C. Zoing. Amyotrophic lateral sclerosis. *The Lancet*, 377(9769):942–955, 2011.

[14] A.D. Lattner and O. Herzog. Unsupervised learning of sequential patterns. In *ICDM 2004 Workshop on Temporal Data Mining: Algorithms, Theory and Applications*, 2004.

[15] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 2–11. ACM, 2003.

[16] K.P. Murphy. *Machine learning: A probabilistic perspective*. MIT Press, 2014.

[17] A. Nahon and B. Lerner. Temporal modeling of ALS using longitudinal data and long-short term memory-based algorithm. In *Proceedings of the 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 609–6014, 2018.

[18] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, and E. Keogh. Generalizing DTW to the multi-dimensional case requires an adaptive approach. *Data Mining and Knowledge Discovery*, 31(1):1–31, 2017.

[19] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology*, pages 1–17. Springer, 1996.

[20] A.A. Taylor, C. Fournier, M. Polak, L. Wang, N. Zach, M. Keymer, J. D Glass, and D.L. Ennist. Predicting disease progression in amyotrophic lateral sclerosis. *Annals of Clinical and Translational Neurology*, 3(11):866–875, 2016.

[21] M.J. Zaki. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1-2):31–60, 2001.