Data-driven predictive maintenance and time-series applications

Kefalas, M.

Chapter 8

An Automated Machine Learning Approach for Electromyography Data

In the previous chapters we mostly dealt with AI-based tools for data-driven applications in predictive maintenance (PdM). In this chapter, we will deal with AI-based time-series applications in the medical domain. Even though the work presented here is not directly relatable to prognostics and health management (PHM) and PdM we still believe that it is a valuable addition to the work discussed in this thesis, as it shows that tools developed for industry can lend themselves to other fields as well. This idea of knowledge transfer across different scientific fields and disciplines is, generally, rather important in the process of knowledge creation, the emergence of new fields and the overall progress of science [215]. It has been shown that knowledge exchange across scientific areas can drive forward and further develop science (see e.g., [7]).

In more detail, in this chapter[1], we will deal with a case study in the field of Neurology, in which we will use methods from time-series representations and other methods (such as in Chapter 4) e.g., feature selection, to ultimately classify patients as either being healthy or not. The approach is automated and limits as many arbitrary choices as possible, providing at the same time valuable diagnostic information without having to rely heavily on clinical expertise. We should note here that the term "automated" refers to the fact that the method can be used in a *generic* way in different domains (domain-agnostic), as we will also note later on. Thus, it should *not* be confused with the notion of "AutoML" (see Section 4.3.5).

[1]

8.1 Introduction

Needle or intramuscular electromyography (EMG) is a common technique used in clinical neurophysiology to record the electrical activity of muscles at different levels of activation [47]. As the EMG signals of patients with both nerve diseases (neuropathies) and muscle diseases (myopathies) differ from those in healthy controls, EMG can be used to diagnose various neurological disorders. The most commonly used method to interpret the EMG is qualitative, based on visual inspection of the signal in real-time by an experienced examiner. A major drawback of this method is that it is highly subjective and prone to errors. In particular, for the diagnosis of myopathies, EMG has been called "one of the most challenging areas in electrodiagnostic medicine" [47]. In theory, a neuropathic EMG with fibrillation potentials, positive sharp waves, high-amplitude and long duration motor unit potentials (MUPs), and a reduced interference pattern should be clearly distinguishable from a myopathic EMG containing smaller, short-duration polyphasic MUPs and a full interference pattern. In practice, however, the diagnostic yield of qualitative EMG analysis for distinguishing between both abnormal-myopathic and neuropathic-myopathic is disappointingly low.

In the past decades, several quantitative EMG (qEMG) methods such as turns-amplitude analysis have been developed in an attempt to increase the diagnostic yield of the EMG. However, so far sensitivity and specificity of various qEMG techniques have remained similar to visual inspection [80, 211]. Similarly, another quantitative technique called the clustering index method yielded a sensitivity of 92% for neurogenic and 61% for myopathic patients [226]. Furthermore, most qEMG methods were published several decades ago and are based on assumptions with regard to MUP morphology and physiology. Interpretation of the EMG in patients with Inclusion Body Myositis (IBM), a myopathy, is particularly challenging, as it may contain both myopathic and neurogenic features [106]. As IBM may also mimic motor neuron disease clinically, inappropriate interpretation of the EMG can lead to an incorrect diagnosis. A retrospective study of mislabeled IBM patients found that routine EMG commonly pointed to a neurogenic disorder called Amyotrophic Lateral Sclerosis (ALS): it showed fibrillations and positive sharp waves, as well as excessive amounts of polyphasic long-duration "neurogenic" MUPs in the majority of mislabeled patients [48]. This is highly unfortunate as ALS, a neuropathy, is a progressive, fatal disease, whereas life expectancy is not significantly affected in IBM [45].

Recent advances in computer processing power and machine learning techniques enable a "big data" approach that processes a large number of features without any underlying assumptions about the nature of the signal. We have previously shown that such an approach, developed for the automotive industry but applied to electroencephalography (EEG) signals, could distinguish between Parkinson's disease patients with good cognition from those with poor cognition with an accuracy of 91% [122].

Generally, a first approach towards a(n) (automatic) classification of specific diseases, either myopathic or neuropathic, is the differentiation between a normal EMG assessment of a healthy individual and an abnormal EMG assessment of a patient with a myopathic *or* neuropathic disease.
In this work, our contributions lie in the following:

1. We aim to evaluate an automated time-series classification algorithm for usage in differentiating EMG time-series of healthy individuals and EMG time-series of patients with either neuropathic *or* myopathic diseases by considering the two types of disease as *one* disease class (binary classification).
2. Our approach is generic and limits as many arbitrary choices as possible, providing at the same time valuable diagnostic information without having to rely heavily on clinical expertise.

8.2 Related Work

Electromyography (EMG) is the study of the electric activity of the muscle and assists in the diagnosis of neuromuscular disorders. EMGs are used to detect and describe different disease processes affecting the motor unit (MU), the smallest functional unit of the muscle. The motor unit action potentials (MUPs) are recorded using a needle electrode at slight voluntary contraction during an EMG. The MUP reflects the electrical activity of a single anatomical motor unit. It represents the compound action potential of those muscle fibers within the recording range of the electrode. EMGs can detect neuromuscular disorders due to the structural reorganization of the MU because of disorders affecting peripheral nerve and muscle [174]. Current clinical practice is based on expert visual inspection of MUP traces and simultaneous real-time assessment of their audio characteristics. This subjective assessment, even if satisfactory, may not be sufficient to describe less apparent deviations or mixed patterns of abnormalities [187]. Therefore, for an automated EMG signal classification to be effective, a systematic and thorough treatment of EMG signals must be carried out. Because of this, a number of computer-based quantitative EMG analysis algorithms have been developed [213].
In this view, authors of [58] developed an EMG-based classifier for neuromuscular disorders using a Multi-Layer Perceptron (MLP). The authors compared the performance of five different feature extraction techniques from the EMG signals (autoregressive, root mean square, mean absolute value, zero crossing, and waveform length) across five different classification tasks: healthy-unhealthy, healthy-myopathy, healthy-neuropathy, myopathy-neuropathy, and healthy-myopathy-neuropathy. Their results showed that the autoregressive feature extraction from the EMG signal returned the best results in four out of five groups, and they achieved the highest accuracy (86.3%) when classifying healthy-myopathy-neuropathy. In [13], a dataset of 50 healthy, 50 neurogenic, and 50 myopathic subjects is generated using an EMG simu-

lation software. The feature set consists of 8 features regarding signal amplitude and phase alongside statistical metrics, such as mean and variance. The classification utilizes four different algorithms with a 97.78% classification accuracy using support vector machines (SVM). In [165] the authors use an openly available clinical database consisting of recordings of ten healthy subjects, seven myopathic, and eight patients with ALS. They use five feature extraction techniques (waveform length, zero crossings, slope sign changes, Willison amplitude, and root mean square). The study reports a 100% accuracy rate for normal subjects, 94% for myopathies and 96% for patients with ALS using the linear discriminant analysis (LDA) classifier. In [101] the authors introduce a novel method for automatic classification of subjects with or without neuromuscular disorders. This method is based on multiscale entropy of recorded surface electromyograms (sEMG) and support vector classification. They achieved a diagnostic yield of 81.5% for healthy/patient classification and 70.4% for healthy/myopathy/neuropathy classification. In [53] the authors describe a method for the classification of neuromuscular disorders. The approach involves isolating single MUPs, computing their scalograms, taking the maximum values of the scalograms in five selected scales, and averaging across MUPs to give a single 5-dimensional feature vector per subject. The SVM analysis reduces the vector to a single decision parameter, called the wavelet index, allowing the subject to be assigned to one of three groups: myogenic, neurogenic, or normal. In [165] Naik et al. present an ensemble empirical mode decomposition algorithm that decomposes a single-channel EMG into a set of noise-canceled intrinsic mode functions, which are then linearly separated by the FastICA algorithm. Five time-domain features extracted from the separated components are then classified using the LDA, and the classification results are fine-tuned with a majority voting scheme. The authors achieved a diagnostic yield of 98% on a clinical EMG database to discriminate between the normal, myopathic, and ALS subjects. More recently, Subasi et al. [214] present a bagging ensemble classifier for the automated classification of EMG signals. They use statistical values of the discrete wavelet transform coefficients and use those as features in a bagging ensemble of SVM, achieving a 99% accuracy for diagnosing neuromuscular disorders.

The work presented above is by no means exhaustive. To the best of our knowledge, though, there has not been much research in hyperparameter tuning in the selected algorithms in this context. The use of hyperparameter optimization techniques would, for example, enhance the model performance further [44]. What is more, it is evident that most of the studies only consider a limited number of features as input to the classifiers (i.e., Hudgin's set of features [95]). An automatic approach to finding relevant time-series representations would create and give insights to new features, or rather biomarkers [122], and would assist in avoiding time-consuming feature engineering processes. In addition, most studies have been done on a specific muscle (e.g., biceps brachii) and not on an arbitrary set of muscles. This could affect the general applicability of the classification task if, for example, a different muscle is put to the test.

This chapter addresses such shortcomings by using a fully automated pipeline to limit arbitrary choices. The pipeline contains units for feature extraction, feature selection, a machine learning model, and hyperparameter optimization. Furthermore, the data used are collected from routine clinical practice rather than in an artificial research setting. Finally, we focus on presenting the machine learning approach in detail.

8.3 Data

The EMG data contain 380 muscle recordings from 65 muscles (at rest or at maximum contraction) based on 65 patients with IBM ($n = 20$), ALS ($n = 20$) and healthy (control group) ($n = 25$). As IBM is relatively rare, we used all available consecutive recordings from 2004-2019. As multiple muscles were examined per patient, we have the EMG of 122 muscles of healthy subjects and 258 muscles of ALS/IBM patients. All recordings were age-matched. These recordings were made within routine clinical care.

The data were collected by the department of clinical neurophysiology of the Leiden University Medical Center (LUMC), a tertiary referral center for neuromuscular diseases[2]. The EMGs were performed with concentric needle electrodes and recorded using Medelec Synergy electromyography equipment[3]. In general, the assessment takes place in three phases: with the muscle at rest, during slight activation, and during (near-) maximal activation. Recording at maximal muscle activation is commonly avoided when the EMG signal appears to be normal at near-maximal activation levels, as the EMG becomes increasingly painful when the muscle is fully activated. The EMG machine routinely stores the last 40 seconds of the examination as 200 consecutive segments of $0.2s$ each (we shall refer to the segment as a trace hereafter). For this study, the longest artifact-free series of consecutive $0.2s$ segments from every muscle recording were selected rigorously by clinicians through visual inspection. This means that for all pairs of patient and muscle, the number of traces varies, and is at *most* 200.

The diagnosis was based on established clinical criteria; in brief: the criteria for IBM were the presence of both typical clinical features and muscle biopsy showing atrophy, inflammation, and rimmed vacuoles. Criteria for ALS were typical clinical features, EMG abnormalities, and progressive neurological decline. Finally, criteria for healthy subjects were defined as subjects with atypical complaints of muscle cramps, pain, or fear of a neuromuscular disease without clinical weakness upon neurological examination and no signs of muscle weakness during a follow-up period of at least two years.

For all the patients and muscles, the data were recorded with two sampling rates, namely 4800Hz and 5000Hz comprising of 16642 and 14279 traces, respectively.

Formally, let $p \in \{1, 2, .., 65\}$ denote the patient, $m \in \{1, 2, .., 65\}$ the muscle, and $t \in$

[2] `https://www.spierziektencentrum.nl/location/lumc/`

[3] Oxford Instruments, Abingdon, Oxfordshire, UK

$\{1, 2, .., Tr_{(p,m)}\}$ the trace. Here, $Tr_{(p,m)}$ stands for the number of traces for each patient and muscle, which depends on the longest artefact-free segment of the muscle recording.
An EMG trace can then be denoted as:

$$\mathbf{s}_t^{(p,m)} := (s_1^t, s_2^t, \ldots, s_{l_t}^t)^\intercal \in \mathbb{R}^{l_t} \quad \forall(p, m, t), \tag{8.1}$$

where l_t is variable and depends on the sampling rate (here 4800Hz or 5000Hz) and duration of the trace (here 0.2s). We can also denote the muscle recording for the tuple (patient, muscle) ($\forall(p, m)$) as:

$$\mathbf{S}^{(p,m)} := [\mathbf{s}_1^{(p,m)}, \mathbf{s}_2^{(p,m)}, \ldots, \mathbf{s}_{Tr_{(p,m)}}^{(p,m)}]^\intercal \in \mathbb{R}^N, \tag{8.2}$$

where $N = l_1 + \cdots + l_{Tr_{(p,m)}}$.
As stated in Section 8.1, our approach is a binary classification task. It aims to differentiate between a normal EMG assessment from a healthy individual and an abnormal EMG assessment from a patient with a myopathic (IBM) or neuropathic (ALS) disease. In this view, the classification targets, labeled by experts, are for each patient $p : \mathcal{T}^p = \{\text{DISEASE}, \text{CTRL}\}$, where DISEASE includes *both* ALS and IBM and CTRL represents healthy controls. It goes without saying that a muscle recording of a patient belonging to a particular class receives the same class label. In Section 8.4 the data pre-processing is described.

8.4 Data Pre-processing

For data pre-processing, we first downsampled all 5000Hz traces to 4800Hz[4]. This was done for consistency as well as for computational purposes. In addition, we renamed certain muscle groups for consistency between recordings (genioglossus → tongue). These pre-processing steps can be considered on a trace level and they transform Equations (8.1) and (8.2) from before to Equations (8.3) and (8.4), respectively, as:

$$\mathbf{s}_t^{(p,m)} := (s_1^t, s_2^t, \ldots, s_l^t)^\intercal \in \mathbb{R}^l \quad \forall(p, m, t), \tag{8.3}$$

where $l = 960$ at a trace duration of $0.2s$ and sampling rate of 4800Hz, and $\forall(p, m)$,

$$\mathbf{S}^{(p,m)} := [\mathbf{s}_1^{(p,m)}, \mathbf{s}_2^{(p,m)}, \ldots, \mathbf{s}_{Tr_{(p,m)}}^{(p,m)}]^\intercal \in \mathbb{R}^{l \cdot Tr_{(p,m)}}, \tag{8.4}$$

In the next steps, we move from the trace level to the muscle level. For this, we designed a unique ID that takes into account the patient identifier, the muscle examined, and the side

[4]We used the resample function of the signal module of the scipy package `https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.resample.html`

examined ($\{\text{Left}, \text{Right}\}$). With this unique ID we grouped together traces belonging to the same patient identifier, the muscle examined, and the side examined. We then reconstructed a 5-second time-series by stitching together consecutive 0.2s segments of each unique ID, which at 4800Hz results in 24000 data points per examined muscle. By creating time-series of equal length, we aimed to avoid bias caused by differences in the sample length and reduce the amount of processing time required. We used the last 5 seconds available from each recording under the assumption that the part of the recording from the muscle at near-maximal contraction is the most likely to contain valuable information for the classification. Nine (9) recordings had fewer than 24000 data points, in which case the entire recording was used. Finally, we discarded 98 recordings with 960 data points *in total*, which correspond to an entire EMG duration of $0.2s$ (at 4800Hz).

Taking Equations (8.3) and (8.4) into account, we denote EMG traces for each patient p, muscle m, and examination side $s \in \{\text{Left}, \text{Right}\}$ as follows:

$$\mathbf{s}_t^{(p,m,s)} := (s_1^t, s_2^t, \ldots, s_l^t)^\intercal \in \mathbb{R}^l. \tag{8.5}$$

And the concatenation of all traces for each patient, muscle, and side is:

$$\mathbf{S}^{(p,m,s)} := [\mathbf{s}_1^{(p,m,s)}, \ldots, \mathbf{s}_{Tr_{(p,m,s)}}^{(p,m,s)}]^\intercal \in \mathbb{R}^N, \tag{8.6}$$

where $N = l \cdot Tr_{(p,m,s)}$ and $l = 960$ is the trace length of $0.2s$ duration and 4800Hz sampling rate.

8.5 Machine Learning Pipeline

The pipeline used in this chapter was initially developed for applications in the automotive industry for time-series classification problems with vehicle onboard data [123, 121]. Later it has been applied to EEG (electroencephalogram) data to predict cognitive function in Parkinson's disease patients potentially eligible for DBS (deep brain stimulation) [122]. The (automated) pipeline has been continuously developed further and consists of the following steps:

1. Feature Extraction from time-series,
2. Feature Selection,
3. Modeling, and
4. Hyperparameter Optimization of the classifier.

The input of this fully automated pipeline are labeled time-series (here: EMG data). The output are performance measures after optimizing the hyperparameters.

8.5.1 Time-Series Feature Extraction

The pipeline aims at being comprehensible, computationally efficient, and applicable to different time-series problems. To ensure this, our pipeline uses features computed from the time-series. Such features are computationally efficient to use and relatively easy to interpret.
In this work, we propose to extract an excessive number of features from the time-series and subsequently select the most significant ones for the problem at hand, based on some pre-defined feature selection criterion. Since those numerous features cover a broad range of time-series characteristics, this procedure allows the application of this pipeline to various problems with very different relevant features.
In this study, the feature extraction $\boldsymbol{\mathcal{F}}$ uses the EMG recordings of each patient and muscle of each side (see Section 8.4) as input and constructs a k-dimensional (k is the number of features) real-valued feature vector, $\boldsymbol{\mathcal{F}}\colon \mathbb{R}^N \to \mathbb{R}^k$:

$$\forall(p,m,s), \quad \mathbf{S}^{(p,m,s)} \mapsto \boldsymbol{\mathcal{F}}\left(\mathbf{S}^{(p,m,s)}\right).$$

Thus, each tuple (p,m,s) results in a feature vector which can be denoted as $\boldsymbol{\mathcal{F}}^{(p,m,s)}$. This feature vector represents the input for the feature selection procedure.
For this task, we used the `tsfresh` package (introduced under Feature Extraction in Section 4.4.2). For the importance of feature extraction, in general, we refer the reader to Section 4.3.3. In this work, `tsfresh` has been applied with its default settings. In the next step, from this generated feature space the most significant features are selected.

8.5.2 Feature Selection

The feature selection phase describes the selection of relevant features from the massive number of extracted features (in this case from `tsfresh`) for the classification task. For each tuple (p,m,s) of patient and muscle, we use $\boldsymbol{\mathcal{F}}_{sel}^{(p,m,s)} \in \mathbb{R}^{k'}$ to represent the vector resulting from feature selection (sel stands for "selected" and k' is the number of selected features). Numerous feature selection methods have been proposed, like the forward or backward selection [36]. To distinguish between relevant and non-relevant features, the so-called feature importance can be used as a measure. Feature importance describes the mean decrease of accuracy or the mean decrease of impurity when modeling with random forests. In a forward selection, features are added iteratively until the feature importance stagnates or deteriorates. Backward elimination uses all features in the beginning and gradually removes less important features. For the importance of feature selection, in general, and other feature selection methods, we refer the reader to Section 4.3.4.
In our pipeline, another feature selection algorithm called `boruta` [129] is used since it has shown better performances when compared to other methods [123]. The `boruta` algorithm

includes a random forest model, which is built on real features and shadow features. Shadow features are generated by randomly shuffling the values of each real feature vector. As soon as a real feature exposes a higher feature importance than the maximal feature importance overall shadow features, it is considered for selection. This procedure is repeated to guarantee that the selected features have a statistically significant meaning.

8.5.3 Modeling

In the modeling phase, a random forest model is trained with the selected features of the previous phase. We have implemented a random forest model due to its simplicity and its efficiency. Furthermore, random forests are known to achieve good performances in different domains. However, any other classifier can be implemented here. A random forest is an ensemble learning method. It is the conglomeration of several decision trees, with the resulting decision being the average outcome of all those decision trees [85] in the case of regression or by taking the majority vote in case of classification.
In this EMG study, we can summarize the input to the random forest model as $\{(\boldsymbol{\mathcal{F}}_{sel}^{(p,m,s)}, \mathcal{T}^{(p)})\}$, where $p \in \{1, \ldots, 65\}$, $m \in \{1, \ldots, 65\}$, $s \in \{\text{Left}, \text{Right}\}$.
We have 380 intramuscular EMG recordings, of which 258 belong to patients with a neuromuscular disorder and the remaining 122 to healthy individuals. Evidently, this dataset is not balanced. Thus, we also performed a balanced approach in addition to the previous modeling approach. In detail, we used a combination of over-sampling the minority class (healthy) and under-sampling the majority class (disease) by allowing the two classes to "meet" halfway (rounded down). In other words, if the difference is 20 data points (EMG recordings), we under-sample the majority class by 10 and over-sample the minority class by another 10. The under-sampling of the majority class happens randomly, whereas the oversampling of the minority class takes place using the well-known *Synthetic Minority Over-Sampling Technique* (`SMOTE`) [39]. The two modeling approaches will be called henceforth *Approach 1* and *Approach 2*. Table 8.1 shows an overview of the modeling approaches.
Since the classification task takes place on the EMG recordings of the muscles, it shall be known henceforth as *muscle-level*. Approach 1 and Approach 2 are the two variants of the muscle-level approach.

8.5.4 Hyperparameter Optimization

The optimization of hyperparameters enhances the performance of a machine learning algorithm. Table 8.2 shows the search space of the hyperparameter optimization (HPO) conducted in this study. Notably, the search space contains not only integer variables but also categorical ones. As discussed already in Section 5.3.5, there are various methods available for HPO like

Table 8.1: Overview of the approaches for automated EMG assessments with Machine Learning.

Approach	**1**	**2**
Description	DISEASE vs. CTRL muscle level	DISEASE vs. CTRL over- and under-sampling muscle level
EMG # cases	380 time-series (EMG)	380 time-series (EMG)
Length	$\leq$ 24000 data points	$\leq$ 24000 data points
Class 0 (CTRL)	122 time-series (EMG)	258 time-series (EMG)
Class 1 (DISEASE)	122 time-series (EMG)	258 time-series (EMG)

grid search, evolutionary algorithms, and Bayesian optimization [78]. In this study, the (single-objective) *Mixed-integer Parallel Efficient Global Optimization* (`MIP-EGO`) [231] is chosen (see also Section 5.3.5). `MIP-EGO` is a state-of-the-art Bayesian optimization algorithm, and is chosen due to its efficiency in optimizing expensive problems. It can efficiently handle mixed-integer categorical variables (such as the ones we have in this work). `MIP-EGO` suggests in each iteration a candidate hyperparameter setting that is evaluated by measuring the model's performance on a test dataset.

To optimize the hyperparameters of the random forest, `MIP-EGO` optimized the F1-macro score of a 10-fold cross-validation (CV). In a CV, the dataset is randomly split into K folds (here $K = 10$), trained on $K - 1$ folds, and tested on the remaining Kth fold. This process is repeated until each fold has served as a test set. The average performance scores from all K folds represents the final score. We executed `MIP-EGO` for 200 iterations, and we used the F1-score macro as our optimization criterion to take into account the class imbalance during training.

Table 8.2: Hyperparameter search space for optimizing the random forest classifier.

Parameter	**Range**
Max depth of each tree	$\{None, 2, 4, 6, \ldots, 100\}$
Number of trees	$\{1, 2, \ldots, 100\}$
Max number of features when splitting a node	{`auto`, `sqrt`, `log2`}
Min number of samples required to split a node	$\{2, 3, \ldots, 20\}$
Min number of samples required in the leaf node	$\{1, 2, \ldots, 10\}$
Use bootstrap training samples?	{`True`, `False`}

8.6 Patient-level Approach

As we already pointed out, the pipeline we have proposed so far operates on *the level of muscles*, meaning it predicts, for each muscle EMG recording (constructed from the same patient and the same side), the probability of this muscle falling into the disease class. In addition, we would like to give a prediction on the *level of patients* to approach the classification in a more holistic view. This approach takes all prediction probabilities on the muscles from the same patient and then aggregates them to make an overall predictive decision for this patient. We will call this approach the *patient-level approach*.

Four different aggregation methods are proposed for the patient-level prediction, which utilize prediction probabilities of the recorded muscles of all the patients:

1. **Majority method**: classify the patient as being in the disease class if more than half of his examined muscles have a score greater than 0.5. Otherwise, classify him as being healthy.
2. **Median method**: classify the patient as being in the disease class if the median of the scores of his examined muscles is greater than 0.5. Otherwise, classify him as being healthy.
3. **Two-muscles method**: classify the patient as being in the disease class if at least two of his examined muscles have a score larger than 0.5. Otherwise, classify him as being healthy. The reason for using more than one muscle in this approach is that by using two muscles we reduce the impact of a potential outlier.
4. **Two-muscles average method**: classify the patient as being in the disease class if the average of two of his examined muscles with the highest score is larger than 0.5. Otherwise, classify him as being healthy.

The difference between methods 3 and 4 above can be made clear with an example. If a patient has 0.80 and 0.49 as the two highest scores, then the two-muscles method would classify him as healthy, whereas the two-muscles-average method would classify him as being in the disease class. Thus, this seems like a necessary and interesting alternative method to examine.

8.7 Performance Evaluation

As previously mentioned, the dataset used in this chapter contains data from 40 patients with neuromuscular disorders and 25 healthy patients. In detail, we have 380 intramuscular EMG recordings, of which 258 have a neuromuscular disorder, and the other 122 are healthy. Evidently, this dataset is not balanced, and thus classification accuracy is not an appropriate performance measure, as it will overestimate the performance. We report it for Approach 2, as the dataset is balanced there, and for completeness, we also report it for Approach 1. In this view, we have also included some other commonly employed performance measures, namely,

precision, recall, F_1-score, sensitivity, specificity, ROC (Receiver operating characteristic) curve, and the area under the ROC (AUC). We explain these performance measures briefly as follows:

- **accuracy**: the number of correct classifications divided by the number of data points.
- **positive class**: *DISEASE* (i.e., the disease class).
- **negative class**: *CTRL* (i.e., the healthy class).
- **true positive**: correct classifications to class *DISEASE*.
- **false positive**: incorrect classifications to class *DISEASE*.
- **precision**: the number of true positive classifications divided by the total number of positive classifications.
- **recall/sensitivity**: the number of true positive classifications divided by the total number of true positives (i.e., true positive rate).
- **Specificity**: the number of true negative classifications divided by the total number of true negatives (i.e., true negative rate).
- $F_1 = 2 \times \text{precision} \times \text{recall}/(\text{precision} + \text{recall})$.
- The **ROC curve** describes the trade-off between true positive rate and false positive rate while the **area under the curve** (AUC) quantifies such a trade-off.

We calculate the $F1$-score, the recall, and precision with two schemes, namely, *macro* and *weighted*. The former calculates metrics for each label (DISEASE, CTRL) and finds their unweighted mean. This does not take label imbalance into account. The latter calculates metrics for each label (DISEASE, CTRL) and finds their average weighted by the class's support (the number of true instances for each label). This alters macro to account for label imbalance. Furthermore, confusion matrices or visualization methods such as ROC can provide deeper performance insights. A confusion matrix describes the frequency of cases that are correctly or incorrectly classified [90] and is considered a useful illustration of the classification quality. Depending on the data, the ROC additionally helps understanding the performance of the model [78].

We clarify the two types of results presented in Section 8.8: the ones obtained from the muscle-level approach and the patient-level approach. The former means that the results underline the performance of the pipeline on the EMG recordings classification task (introduced in Section 8.5.3). Approach 1 and Approach 2 (introduced in Section 8.5.3) are the two variants of the muscle-level approach. The latter quantifies the performance of the post-processing task, which aims to classify the patients using the output of the muscle-level pipeline (introduced in Section 8.6).

8.7.1 Muscle-Level Performance Evaluation

Due to the small number of EMG recordings (380 samples), we decided to validate the entire muscle-level pipeline using a 10-fold CV. We should note here that the (nested) CV of the

hyperparameter optimization process (see Section 8.5.4) was executed on the *training-fold* of each split of this, overall, 10-fold CV process. This allows the hyperparameter optimization task to be unbiased, as it *does not* take into account the test set of the overall 10-fold CV process. Moreover, the balancing of modeling Approach 2 (see Section 8.5.3) is applied *only* to the training set in each fold of the 10-fold CV.
We would also like to emphasize here that during the CV in the pipeline, the folds are generated in a *patient level* way (not to be confused with the *patient-level approach* introduced in Section 8.6). This means that the EMG recordings belonging to one patient are **all** included in the training or testing fold and are **never** separated between the training data and test data. This is important in order to prevent data leakage, as two different EMG recordings of one patient carry similar information about the underlying process that generated them (i.e., same pathophysiology). Each resulting performance score represents the average of 5 independent runs of the described pipeline.

8.7.2 Patient-Level Performance Evaluation

The resulting performance scores for the patient-level are based on the post-processing of the scores returned by the automatic machine learning pipeline (muscle-level). For the patient-level approach, we follow the procedure explained in detail in Section 8.6. Each resulting performance score of the patient-level approach represents the average of the post-processing of the 5 independent runs of the automatic machine learning pipeline.

8.8 Results

In this section, the results of the muscle-level and patient-level classification tasks are presented.

8.8.1 Muscle-level results

The muscle-level approach aims at classifying intramuscular EMG recordings as either being in the DISEASE class (ALS/IBM) or the CTRL class (healthy). In Table 8.3, we present the results for Approach 1 and Approach 2 of the muscle-level. For clarity, Approach 1 refers to the unbalanced muscle-level pipeline and Approach 2 refers to the balanced muscle-level pipeline (see also Table 8.1 and Section 8.5.3). Furthermore, Figures 8.1 and 8.2 show the confusion matrices of both modeling approaches 1 and 2 for the training and the test set, respectively.
First of all, the achieved results indicate that machine learning techniques can carry out a task like this. Comparing between Approaches 1 and 2, Table 8.3 shows that Approach 1 (AUC = 0.817) is generally better suited for this task than Approach 2 (AUC = 0.795), although the difference between the two is minimal. Here, we take the AUC as the major

performance value since it quantifies the best potential performance for both approaches, while the other scores only compare them with a fixed decision threshold (0.5 in this chapter). From Figures 8.1 and 8.2 we can see that the sensitivity of Approach 1 is greater than that of Approach 2, however the specificity of Approach 2 is greater than that of Approach 1. This can also be backed-up from Table 8.3 where the sensitivity of Approach 1 and 2 is 0.896 and 0.816, respectively, whereas the specificities are 0.546 for Approach 1 and 0.604 for Approach 2. A reason for this behavior could be partially due to the fact that for Approach 2, we reduce in every fold the training data of our positive class and increase the training data of our negative class in order to balance the data points between the two labels.
Finally, in Table 8.4 we can see the common features[5] selected in every fold of the 10-fold CV and in every single of the 5 independent runs. We show their aggregated impurity-based importance values (averaged over a 10-fold CV and then averaged over all 5 repeated runs of the 10-fold CV) and the standard deviation of the means over the 5 runs. The standard deviation shows that the average importance of these features has been consistent throughout the runs, and that their ranking is quite reliable. These features should be further investigated for their predictive power, clinical relevance, and interpretability.

Table 8.3: Performance scores for the muscle-level **Approach 1** and **Approach 2**. The scores are calculated on the test set and averaged over a 10-fold cross validation. The mean and standard deviation are calculated from 5 repeated runs of the 10-fold CV.

Score	**Approach 1**	**Approach 2**
Accuracy	0.778±0.021	0.747±0.009
F1 (macro)	0.708±0.027	0.692±0.012
F1 (weighted)	0.759±0.021	0.740±0.008
Precision (macro)	0.767±0.032	0.723±0.013
Recall (macro)	0.721±0.025	0.710±0.011
Precision (weighted)	0.792±0.029	0.773±0.005
Recall (weighted)	0.778±0.021	0.747±0.009
Sensitivity	0.896±0.015	0.816±0.006
Specificity	0.546±0.037	0.604±0.025
AUC	0.817±0.023	0.795±0.031

8.8.2 Patient-level results

The patient-level approach aims at classifying patients as either being in the DISEASE class (ALS/IBM) or the CTRL class (healthy), based on the post-processing of the prediction scores

[5]Please see `https://tsfresh.readthedocs.io/en/latest/text/list_of_features.html`
[6]Counting starts from 0.

		Predicted	
		CTRL	DIS
Actual	CTRL	85.15	24.65
	DIS	2.19	230.01

		Predicted	
		CTRL	DIS
Actual	CTRL	6.52	5.69
	DIS	2.79	23.01

Figure 8.1: Confusion matrix of modeling **Approach 1** for the training data (left) and test data (right). *CTRL* is the CTRL class, referring to healthy recordings and *DIS* is the DISEASE class, referring to the disease recordings. The scores are calculated and averaged over all folds of the 10-fold cross validation. The values are averaged over 5 repetitions of the 10-fold CV.

		Predicted	
		CTRL	DIS
Actual	CTRL	169.41	1.84
	DIS	0.2	171.062

		Predicted	
		CTRL	DIS
Actual	CTRL	7.38	4.82
	DIS	5.1	20.75

Figure 8.2: Confusion matrix of modeling **Approach 2** for the training data (left) and test data (right). *CTRL* is the CTRL class, referring to healthy recordings and *DIS* is the DISEASE class, referring to the disease recordings. The scores are calculated and averaged over all folds of the 10-fold cross validation. The values are averaged over 5 repetitions of the 10-fold CV.

of their intramuscular EMG recordings, from Approach 1 and Approach 2 of the muscle-level. In Table 8.6 we show the performance scores of all the methods of the patient-level post-processing on Approach 1 and Approach 2.

The results indicate again that machine learning techniques can carry out a task like this. Comparing the methods and approaches within Table 8.6, we see that the patient-level post-processing of Approach 1 has a higher diagnostic yield than the patient-level post-processing of Approach 2. This is also backed up when comparing the AUC between the two approaches. In more details, we see that the AUC of the median and two-muscles average of the patient-level post-processing of Approach 1 is 0.815 and 0.798, respectively, compared to 0.786 and 0.777 of patient-level post-processing of Approach 2. A closer look at Table 8.6 suggests that generally, for the patient-level post-processing of Approach 1, the majority method allows for the best results in terms of the F1 score (for both "macro" and "weighted" averages), with the two-muscles coming in the second rank, then the median method for the "macro" average and the two-muscles average for the "weighted" average. The two-muscles average method comes last in the "macro" average, and the median method for the "weighted" average. For the patient-level post-processing of Approach 2, the two-muscles come first, then the two-muscles average, then the majority method, and last the median method. Note that the AUC score is not used to compare all methods since it is not defined for the majority and two-muscles methods. Figure 8.4 show the ROCs curves from all 5 repetitions of the median and two-muscles average methods of the patient-level post-processing of Approach 1. Figure 8.3 shows the confusion matrices of all the methods of the patient-level post-processing of Approach 1. From Figure 8.3

Table 8.4: Impurity-based importance scores for **Approach 1** of the muscle-level. These are the common features selected by `Boruta` in every fold of the 10-fold CV and in every repetition of the 10-fold CV. The importance scores are calculated and averaged over all folds of the 10-fold CV. The mean and standard deviation are calculated from 5 repeated runs of the 10-fold CV.

Feature	Importance Score
Percentage of values that are present in the time-series more than once. (percentage_of_reoccurring_values_to_all_values)	4.6 ± 0.12
Absolute value of the 35^{th} fourier coefficient[6] of the 1D discrete FFT of a real input. (fft_coefficient__coeff_34__attr_"abs")	4.43 ± 0.1
Absolute value of the 32^{nd} fourier coefficient of the 1D discrete FFT of a real input. (fft_coefficient__coeff_31__attr_"abs")	3.53 ± 0.13
Factor which is 1 if all values in the time-series occur only once, and below one if this is not the case. (ratio_value_number_to_time_series_length')	3.48 ± 0.06
Absolute value of the 41^{st} fourier coefficient of the 1D discrete FFT of a real input. (fft_coefficient__coeff_40__attr_"abs")	3.46 ± 0.12
Percentage of non-unique data points. (percentage_of_reoccurring_datapoints_to_all_datapoints)	2.91 ± 0.05

and Table 8.6 we see that the method with the highest sensitivity in the patient-level post-processing of Approach 1 is the two-muscles average method. The reason for this might lie in the fact that ALS and IBM are "patchy" diseases, meaning that only a proportion of muscles may be affected at the time of the EMG recording. Therefore, the two-muscles average method is more sensitive, as we also explain in Section 8.6.

Finally, in order to see whether using hyperparameter optimization (HPO) on the model's hyperparameters can indeed improve the performance of the developed methodology, in Table 8.5 we calculated the average percentage of improvement for each patient-level method when using HPO versus not using HPO in both approaches 1 and 2. We averaged the percentages of improvement overall the performance metrics of each method. The last row shows the overall average improvement of these methods. From Table 8.5 we can see an average improvement of 2.94% on the patient-level post-processing of Approach 1 when using HPO on Approach 1, compared to using the default values of the random forest algorithm[7] (no HPO) and 0.75%

[7]See here for the default values `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier`

Table 8.5: Percentage of Improvement of the patient-level post-processing of Approach 1 and Approach 2, using hyperparameter optimization vs no hyperparameter optimization. Each row shows the average improvement for that patient-level's method performance metrics. The last row shows the average improvement overall these methods.

Patient-level method	**Approach** 1	**Approach** 2
Majority	4.73%	0.14%
Median	1.87%	−1.13%
Two Muscles	2.55%	2.59%
Two Muscles Average	2.61%	1.41%
Average Improvement	2.94%	0.75%

for the patient-level post-processing of Approach 2 when using HPO on Approach 2. From Table 8.5 we see that HPO can have a positive or negative impact based on the experimental setup (e.g., median method for patient-level post-processing of Approach 1 vs median method for patient-level post-processing of Approach 2). However, we can see that the usage of HPO can, in general, lead to improved performance, even though the improvement can be marginal in some cases (e.g., in patient-level post-processing of Approach 2).

(a)

Actual \ Predicted	CTRL	DIS
CTRL	12.6	12.4
DIS	3.4	36.6

(b)

Actual \ Predicted	CTRL	DIS
CTRL	14.4	10.6
DIS	3.6	36.4

(c)

Actual \ Predicted	CTRL	DIS
CTRL	12.6	12.4
DIS	2.6	37.4

(d)

Actual \ Predicted	CTRL	DIS
CTRL	11.2	13.8
DIS	1.4	38.6

Figure 8.3: Confusion matrices of all the methods of the **patient-level** post-processing of modeling **Approach 1**. *CTRL* is the CTRL class, referring to the healthy controls and *DIS* is the DISEASE class, referring to the disease patients. (a) Median method, (b) Majority method, (c) Two-muscles method, (d) Two-muscles-average method. The entries are averaged over all 5 repetitions.

8.9 Discussions and Conclusions

This chapter presents an automated method for classifying electromyography (EMG) data on a muscle-level and a patient-level method for classifying patients. Both tasks aim at classifying between healthy and not healthy. Our dataset contains 65 patients and 65 muscles. As multiple

muscles were examined per patient, we have the EMG of 122 muscles of healthy subjects and 258 muscles of ALS/IBM patients. The data were collected from routine clinical practice rather than in an artificial research setting.

Our method extracts and selects the most significant features from the time-series, trains a random forest model, and optimizes its hyperparameters in an automated approach for the muscle-level classification task. We develop two approaches for this classification task: one where the data labels are kept imbalanced (Approach 1) and one where we balance the labels (Approach 2). The results indicate that machine learning techniques can carry out the task of distinguishing between normal and abnormal EMGs. Comparing Approach 1 and Approach 2 shows that Approach 1 (AUC = 0.817) is generally better suited for this task than Approach 2 (AUC = 0.795), although the difference between the two is minimal. Taking into consideration Figure 8.1 for Approach 1, we see that the test error is slightly higher than the training error. The reason for this can be attributed to the small sample size used in this study. For Approach 2 (see Figure 8.2), we argue that the testing result can not be compared directly to that on the train set since the class-balancing procedure is only applied on the training set. We also see that in both approaches, sensitivity outweighs specificity. As a screening algorithm, high sensitivity is preferable to limit the number of false-negatives. From a clinical point of view, sensitivity is the more important metric in this algorithm. We should also emphasize that the automatically computed features allow for a high diagnostic yield. Since EMG classification is routinely performed qualitatively, this method allows for identifying new EMG biomarkers.

For the patient-level classification task, the achieved results indicate again that we can automate this process using machine learning techniques. We see that the patient-level post-processing of Approach 1 has a higher diagnostic yield than the patient-level post-processing of Approach 2. This is also backed up when comparing the AUC between the two approaches. In more detail, we see that the AUC of the median and two-muscles average of the patient-level post-processing of Approach 1 is 0.815 and 0.798, respectively, compared to 0.786 and 0.777 of the patient-level post-processing of Approach 2. The results further show that the majority method yields the best results in terms of the F1 score (for both “macro” and “weighted” averages), with the two-muscles coming in the second rank, then the median method for the “macro” average and the two-muscles average for the “weighted” average. The two-muscles average method comes last in the “macro” average and the median method for the “weighted” average. Similarly, for the patient-level post-processing of Approach 2, the two-muscles come first, then the two-muscles average, then the majority method, and last the median method. Finally, we saw an average improvement of 2.94% on the patient-level post-processing of Approach 1 when using hyperparameter optimization (HPO) on Approach 1, compared to using the default values of the random forest algorithm (no HPO) and 0.75% for the patient-level post-processing of Approach 2 when using HPO on Approach 2. These results validate the application of HPO to both approaches. The results further indicate that HPO can have a positive or negative impact based

on the experimental setup (e.g., median method for patient-level post-processing of Approach 1 vs median method for patient-level post-processing of Approach 2). This, however, is still to be investigated further.

To conclude, we see that the algorithms presented can assist clinicians in diagnosing if a patient has a neuropathy/myopathy or is healthy. In fact, the EMG in ALS patients is likely to show neurogenic changes (e.g., increased MUP amplitudes compared to healthy subjects), whereas the EMG of IBM patients is more likely to show myopathic changes (e.g., decreased MUP amplitudes). The fact that our proposed method reaches a relatively high performance despite the heterogeneity of the "diseased" group shows its potential. Indeed, performance may be higher when a similar approach is used to distinguish healthy controls from ALS- or IBM-patients as separate groups. In addition, both ALS and IBM can be "patchy" diseases, meaning that only a proportion of muscles may be affected at the time of the EMG recording. As the EMG signal of non-affected other muscles is expected to be similar to that of healthy controls, at least when using the current qualitative assessment, it is remarkable that the performance of the muscle-level approach was relatively high. This suggests that the EMG signal of these, apparently normal, muscles may contain information that is used by the ML-based approach but not during routine clinical assessment.

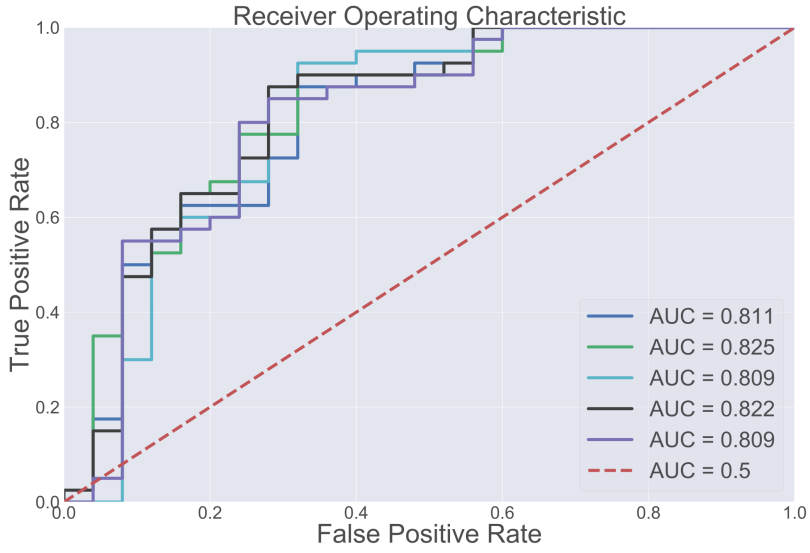
A major limitation of this study lies in the relatively small dataset. This is unavoidable given the rarity of IBM in particular, which has a current population of less than 100 patients in the Netherlands [17]. We specifically investigated IBM and ALS patients because of the well-known clinical difficulties in interpreting the EMG of these diseases. Whether our approach works equally well for other myopathies/neuropathies remains to be established. However, as these are usually easier to classify using current clinical assessment, we would expect the performance of our ML approach to be higher, rather than lower, as well. An additional limitation of the current approach is the random selection of each muscle's final 5-second EMG segment. This selection was based on the absence of artifacts without using any information on the level of muscle activation. However, we aimed to use the last 5 seconds available, assuming that this segment was more likely to contain information of the muscle at (near-) maximal contraction. Longer recordings, in which the clinical level of muscle activation is clearly marked, may lead to further improvements in performance, as muscle activity at rest is different in both IBM and ALS patients compared to healthy subjects.

Future research should emphasize a more detailed analysis of the nature of the selected features that could point towards useful biomarkers for disease progression. Furthermore, future work should investigate a patient-level pipeline to classify patients into a class directly.

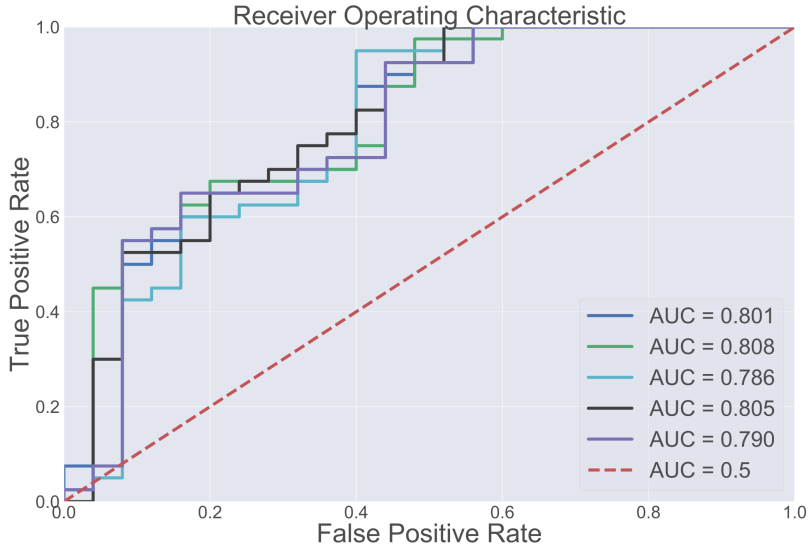
Table 8.6: Performance scores of all the methods of the patient-level post-processing on modeling approaches 1 and 2, tested in this chapter. The scores are calculated on the test set and averaged in a 10-fold cross validation. The mean and standard deviation are calculated from 5 repeated runs of the 10-fold CV. Note that for the majority and two-muscle methods the AUC scores are not applicable. The reason behind that is that we decided to use a fixed score threshold.

Approach	**Method**	**Accuracy**	**F1 macro**	**F1 weighted**	**Precision macro**	**Recall macro**
Approach 1	Majority	0.782±0.028	0.753±0.032	0.772±0.029	0.789±0.035	0.743±0.03
	Median	0.757±0.033	0.718±0.04	0.742±0.036	0.768±0.041	0.710±0.037
	Two-Muscles	0.769±0.022	0.73±0.024	0.753±0.022	0.794±0.038	0.72±0.022
	Two-Muscles Average	0.766±0.02	0.716±0.021	0.743±0.019	0.815±0.044	0.707±0.018
Approach 2	Majority	0.72±0.02	0.701±0.024	0.718±0.021	0.705±0.021	0.701±0.025
	Median	0.717±0.018	0.696±0.023	0.714±0.02	0.7±0.019	0.694±0.025
	Two-Muscles	0.738±0.022	0.707±0.021	0.729±0.021	0.732±0.03	0.701±0.019
	Two-Muscles Average	0.742±0.013	0.704±0.018	0.728±0.015	0.742±0.015	0.697±0.016

Approach	**Method**	**Precision weighted**	**Recall weighted**	**Sensitivity**	**Specificity**	**AUC**
Approach 1	Majority	0.786±0.031	0.782±0.028	0.91±0.034	0.576±0.054	—
	Median	0.763±0.037	0.757±0.033	0.915±0.03	0.504±0.061	0.815±0.008
	Two-Muscles	0.784±0.031	0.769±0.022	0.935±0.038	0.504±0.046	—
	Two-Muscles Average	0.797±0.036	0.766±0.02	0.965±0.029	0.448±0.018	0.798±0.01
Approach 2	Majority	0.719±0.021	0.72±0.02	0.785±0.034	0.616±0.061	—
	Median	0.714±0.021	0.717±0.018	0.795±0.011	0.592±0.059	0.786±0.021
	Two-Muscles	0.736±0.025	0.738±0.022	0.865±0.034	0.536±0.022	—
	Two-Muscles Average	0.742±0.013	0.742±0.013	0.890±0.014	0.504±0.036	0.777±0.02



(a)



(b)

Figure 8.4: (a): ROC curves of all 5 repetitions of the **median** method on the **patient-level** post-processing of modeling **Approach 1**. (b): ROC curves of all 5 repetitions of the **two-muscles average** method on the **patient-level** post-processing of modeling **Approach 1**.