

Unsupervised Minimum Redundancy Maximum Relevance Feature Selection for Predictive Maintenance: Application to a Rotating Machine

Valentin Hamaide and François Glineur

UCLouvain - ICTEAM and CORE Research Institute, Louvain-la-Neuve, Belgium

valentin.hamaide@uclouvain.be

francois.glineur@uclouvain.be

ABSTRACT

Identifying and selecting optimal prognostic health indicators in the context of predictive maintenance is essential to obtain a good model and make accurate predictions. Several metrics have been proposed in the past decade to quantify the relevance of those prognostic parameters. Other works have used the well-known minimum redundancy maximum relevance (mRMR) algorithm to select features that are both relevant and non-redundant. However, the relevance criterion is based on labelled machine malfunctions which are not always available in real life scenarios. In this paper, we develop a prognostic mRMR feature selection, an adaptation of the conventional mRMR algorithm, to a situation where class labels are a priori unknown, which we call unsupervised feature selection. In addition, this paper proposes new metrics for computing the relevance and compares different methods to estimate redundancy between features. We show that using unsupervised feature selection as well as adapting relevance metrics with the dynamic time warping algorithm help increase the effectiveness of the selection of health indicators for a rotating machine case study.

1. INTRODUCTION

Predictive maintenance, or condition-based maintenance, consists of recommending maintenance decisions based on the information collected through condition monitoring, usually in the form of time series. It can be divided into two kinds of problems: i) detecting that the machine under monitoring has entered a faulty state, and therefore predicting that a failure is coming, or ii) predicting the remaining useful life (RUL) of the machine. Usually the first approach can be a trigger for the second one but this might not always be the case. This paper tackles the predictive maintenance problem as a whole:

it seeks for the best combination of features (or health indicators), which is useful for both predictive maintenance tasks.

In a recent survey (Lei et al., 2018), the authors outline a systematic framework for predictive maintenance based on four steps: (1) Data acquisition: capturing and storing the data coming from the different sensors, (2) Health indicators' construction: finding features that represent the health's evolution of the monitored machine, (3) Health states' division: dividing the machine's lifetime into several health states and (4) RUL prediction: estimating the time remaining before the machine needs to be replaced. Our paper focuses on the selection of the subset of health indicators (HI), most commonly referred to as feature selection, that allows the most accurate separation of health states or the best RUL estimation. It falls between steps (2) and (3) of the aforementioned predictive maintenance framework. It is assumed that a set of HI was previously constructed. There are plenty of scientific articles that tackle this issue. In the case of vibration data, the reader can refer to Wang et al. (2017) for a review of the common health indicators.

Feature selection has been applied in the predictive maintenance context in two main ways. In the first approach, the selection of features is based solely on one or several prognostic metrics quantifying the relevance of a feature with respect to the prognostic task. Coble initiated this in her 2010 doctoral dissertation (Coble, J. B., 2010) where she derives three complementary metrics of what a good prognostic parameter is. Subsequently, other prognostic metrics were proposed by other authors. An overview of the proposed metrics in the literature is presented in section 2.1. A possible drawback with this approach is that the redundancy between features is not taken into account.

The second method for selecting features is based on their relevance to a class label rather than their relevance with respect to a prognostic metric. This falls in the framework of

Valentin Hamaide et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

<https://doi.org/10.36001/IJPHM.2021.v12i2.2955>

supervised feature selection where we assume that labelled machine malfunctions are available in the case of classification, or that the remaining useful life is used as the label for a regression. Supervised feature selection methods (Chandrashekar & Sahin, 2014) can be categorized according to their strategy to select features: filter or wrapper approach. In the filter approach, features are ranked according to a specific criterion, usually a statistical or information theory measure between the feature and the supervised label. As such, the filter approach does not take redundancy between features into account. However, one can use the minimum redundancy maximum relevance (mRMR) algorithm (Peng et al., 2005) that takes into account both the relevance and the redundancy between features via the mutual information criteria. Several authors applied this mRMR approach with known class labels to select features in the predictive maintenance context, namely e.g. Y. Li et al. (2017); X. Zhang et al. (2018); Yan & Jia (2019); Tang et al. (2020); Hu et al. (2020); Shahidi et al. (2016). Liu et al. (2013) also include a redundancy analysis in the selection of features, but they do so via a method they call effectiveness–correlation fusion. They compute both effectiveness scores of features using several machine learning criteria (kernel class separability, margin width, scatter matrix, Pearson correlation with labels, etc.) and redundancy between features via Pearson’s correlation.

In the wrapper approach, feature selection is performed based on the predictor, i.e. the predictor algorithm is wrapped into a search algorithm which seeks a subset of features that yields the highest classifier performance. This approach is however computationally intensive and depends on the classification/regression algorithm used (meaning that a different set of features would be selected for a different classifier algorithm). A drawback of using the classifier performance as the criterion for selecting features is that the classifier is prone to overfitting (Chandrashekar & Sahin, 2014).

While supervised feature selection is the way to go if labelled machine malfunctions are available, it is usually not the case for most real-life applications. Another possibility would be to use the time before failure as class labels for a regression. However, this is not guaranteed to produce good results, since degradation usually appears at a certain point and is rarely a continuous degradation process starting at the beginning of machine life. Moreover, degradation is not necessarily linear, while the regression label based on time before failure is.

The main purpose of this paper is to propose a feature selection approach for predictive maintenance that considers both the relevance and redundancy between features without the need for class labels. The idea is to adapt the mRMR algorithm, and more specifically, the “maximum relevance” part of the algorithm where the features are not compared to a class label but to a prognostic metric. This prognostic mRMR

feature selection could be classified as an unsupervised feature selection with a filter approach.

Several unsupervised feature selection methods have been proposed in the machine learning context. A recent survey (Solorio-Fernández et al., 2020) details the most common algorithms and provides a taxonomy of those methods. These algorithms can also be divided into filter and wrapper approaches where the former ranks features according to information theory or spectral similarity concepts, and the latter does so mainly through clustering techniques. The criteria for feature relevancy are however difficult to assess. Those methods consist of choosing features to best preserve the manifold structure of original data, seeking cluster indicators or else defining a criterion based on the correlation between features. The last approach is used in Fernandes et al. (2019) for a metallurgic application where features are selected according to their lowest pairwise correlation. While those approaches are interesting in the absence of any knowledge about what a good feature should be, we believe there is room for improvement as we can make assumptions about what a good prognostic parameter should be in the context of predictive maintenance. According to Coble & Hines (2009), a good feature should have a monotonic dependence with time, have the same underlying shape across different machines and show high separability between starting and failure values. Knowing this, we can define prognostic metrics for feature relevancy and couple them with a modified version of the mRMR algorithm to select non-redundant features. This association of prognostic metrics with the mRMR algorithm is the central idea proposed in this paper.

The remainder of this paper is structured as follows: section 2 discusses previous research on the topic: the existing relevance metrics in prognostics and the minimum redundancy maximum relevance (mRMR) algorithm. In section 3, we present our approach, which involves improving existing metrics and adapting the mRMR feature selection approach to a situation without class labels. The algorithm is then tested on a rotating machine application and section 5 concludes.

2. BACKGROUND

First, a set of sensor measurements need to be acquired on several machines from the installation to the failure (or at least until a degradation occurs), as opposed to machines being preventively replaced. Indeed, to identify a relevant subset of features, we can expect to observe signs of deterioration in some of the features for the machines that went through a failure and not necessarily for those replaced at an early stage. We refer to the data collected from a particular machine as a run-to-failure time series. Let us define $x_i^{(r)} \in \mathbb{R}^{n_r \times 1}$, the i^{th} feature of the run-to-failure time series r , where n_r is the number of samples in the time series. The dataset

consists of N features and R run-to-failure time series, i.e. $x^{(r)} \in \mathbb{R}^{n_r \times N}$ for $r = 1, \dots, R$. The notation $x_i^{(r)}(t)$ is used to access the t^{th} sample in the array and $x_i^{(r)}(-t)$ to access the t^{th} sample in the array starting from the end. Referring to x_i actually refers to the collection of time series (of possibly different lengths) $x_i = \left(x_i^{(r)} \right)_{r=1, \dots, R}$. After acquiring data from sensor measurements and constructing a set of features, the feature selection can start.

2.1. Existing relevance metrics

In her 2010 doctoral dissertation, Coble, J. B. (2010) investigated several prognostic metrics for feature selection. She derived three complementary metrics that define a good prognostic parameter: monotonicity, trendability and prognosability. The first one quantifies the prognostic feature's underlying positive or negative trend, while trendability indicates the degree to which the features of a set of machines have the same underlying shape. The last complementary metric, prognosability, refers to a measure that encourages well-clustered failure values and high separability with starting values.

Monotonicity is defined as the difference between the number of positive and negative slopes computed for each pair of successive times steps (i.e. by computing $\text{sign}(x(t+1) - x(t))$) divided by the number of time steps. Prognosability is computed as the ratio between the standard deviation of the critical failure values of a set of machines and its mean range between starting and failure values. The result is exponentially weighted to obtain a metric with values between zero and one. The metric encourages well-clustered values, i.e. a parameter with small standard deviation before failure and large parameter range across the machine's life. Finally, the trendability of a feature is defined as the minimum correlation between pairs of machines according to that feature. A caveat in this metric is that it requires to compute the correlation with time series of different lengths. Different methods to tackle this issue are discussed in subsection 3.1.3. In Coble, J. B. (2010), the prognostic features are resampled with respect to the fraction of total lifetime into 100 observations, with each observation corresponding to 1% of lifetime.

Other metrics have also been explored since Coble. Camci et al. (2013) provide another formulation for monotonicity by dividing a HI into different stages. Other monotonicity metrics quantify the dependence between the HI and time (Javed et al., 2014; N. Li et al., 2014; Javed et al., 2013). Note that in the aforementioned references, the name trendability is used instead of monotonicity for the metrics that quantify the dependence between the HI and time. This naming convention can be somewhat confusing for the reader. In this paper, we shall also refer to those metrics as monotonicity since a correlation between a HI and time induces that the metric is mono-

tonic (since time is monotonic in a time series). Spearman's rank correlation was used in Lei et al. (2016) and Carino et al. (2015) to account for non-linear relationships between the HI and time instead of linear relationships in the conventional Pearson's correlation.

B. Zhang et al. (2016) propose a robustness metric to quantify the smoothness of the degradation trend. Metrics that quantify the dependence between a HI and different health states via Pearson's correlation (for classification purposes) have been explored in Zhao et al. (2013) and Liu et al. (2016). Liu et al. (2016) also define a metric to quantify the correlation between multiple HI in order to limit the selection of correlated features.

2.2. mRMR algorithm

The minimum redundancy maximum relevance algorithm was developed for pattern recognition by Peng et al. (2005). The idea of the algorithm is to select a subset of features $\{x_i\}$ that is both relevant and non-redundant based on the concept of mutual information. The mutual information between two features x and y is expressed based on the joint probability distribution $p(x, y)$ and marginal probabilities $p(x)$ and $p(y)$:

$$MI(x, y) = \sum_{i,j} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)} \quad (1)$$

It is equal to zero if and only if the two random variables are independent, and higher values mean higher dependency. Mutual information is closely related to the concept of entropy. Indeed, the mutual information between two variables can be expressed as $MI(x, y) = H(x) + H(y) - H(x, y)$ where $H(x)$, $H(y)$ and $H(x, y)$ are respectively the entropy of variables x and y , and the joint entropy between x and y .

From the mutual information point of view, the purpose of feature selection is to select features that jointly have the largest dependency on a target class c . Because it is usually hard to obtain an accurate estimation of multivariate density $p(x_1, \dots, x_m)$ and $p(x_1, \dots, x_m, c)$, as well as computationally challenging, the mRMR method is used. The concept is based on two concurrent optimization problems, the max-relevance defined as

$$\max_S D(S, c) \triangleq \frac{1}{|S|} \sum_{i=1}^{|S|} \text{MI}(x_i; c) \quad (2)$$

and the minimum redundancy defined as

$$\min_S R(S) \triangleq \frac{1}{|S|(|S|-1)} \sum_{i=1}^{|S|} \sum_{j=1, j \neq i}^{|S|} \text{MI}(x_i, x_j) \quad (3)$$

The formulation of the optimization problem in equations (2-3) requires to jointly optimize two different objectives which

is not possible as such. Therefore, the problem is reformulated as a single objective optimization by combining the two into a single expression. Two cases are defined:

$$\max(D - R) \quad (4)$$

$$\max\left(\frac{D}{R}\right) \quad (5)$$

that we refer to as OFD (objective function difference) and OFQ (objective function quotient) respectively.

Exact solution to the mRMR problem requires to enumerate $\binom{|S|}{M}$ possible combinations of features, where M is the total number of features and $|S|$ the number of features we wish to select. Note that the number of possible combinations would increase to 2^M should we allow the selection of any number of features. In practice, a near optimal solution is usually sufficient. Incremental search methods can be used to find a set of features with an $O(|S| \cdot M)$ complexity. Suppose we already have S_{m-1} , the selected set with $m - 1$ features, the aim is then to find the m^{th} feature from the set $X \setminus S_{m-1}$. This is done by selecting the feature that maximizes (4):

$$\max_{x_j \in X \setminus S_{m-1}} \left(\text{MI}(x_j; c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} \text{MI}(x_j, x_i) \right) \quad (6)$$

or the feature that maximize (5):

$$\max_{x_j \in X \setminus S_{m-1}} \left(\frac{\text{MI}(x_j; c)}{\frac{1}{m-1} \sum_{x_i \in S_{m-1}} \text{MI}(x_j, x_i)} \right) \quad (7)$$

In addition to the computational reduction of the mRMR compared to the original joint maximum dependency selection, the authors proved that the mRMR formulation is equivalent to this maximum dependency criterion if one feature is selected (added) at a time (Peng et al., 2005).

3. UNSUPERVISED MRMR FEATURE SELECTION

This section describes our algorithm for unsupervised minimum redundancy maximum relevance feature selection applied to predictive maintenance, which we call prognostic mRMR. Subsection 3.1 characterizes the relevance of a feature via three prognostic metrics that are improved versions of the metrics from Coble & Hines (2009). More specifically, we suggest improvements to increase robustness of the monotonicity metric and propose alternative strategies to handle the different lengths of the run-to-failure time series in the trendability metric. To take into account the redundancy between features, we adapt the mRMR algorithm in the absence of class labels in section 3.2 and propose different strategies to compute the redundancy between features.

3.1. Feature relevance: prognostic metrics

3.1.1. Monotonicity

We define monotonicity using Spearman's rank correlation:

$$M(x_i) = \frac{1}{R} \sum_{r=1}^R \text{corr}(\text{rank}(x_i^{(r)}), \text{rank}([1, \dots, n_r])) \quad (8)$$

where $\text{corr}(x, y)$ is Pearson's correlation coefficient between variable x and y :

$$\text{corr}(x, y) = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}}$$

and $\text{rank}(x)$ is the relative position label of the observations within the variable. Defining monotonicity in this way instead of counting positive and negative slopes ($\text{sign}(x(t+1) - x(t))$) as done in Coble & Hines (2009) has the advantage of being a lot less subject to noise in the data as shown in Figure 1a. In addition, Spearman's rank correlation is used instead of Pearson's correlation for three reasons. First, Spearman's correlation is better suited for non-linear relationships between the HI and time (Figure 1b). Second, it is less sensitive to strong outliers as can be observed in Figure 1d. Finally, for mostly uncorrelated data, the two measures are similar (Figure 1c).

3.1.2. Prognosability

The prognosability metric used here is the same as in Coble & Hines (2009), except that failure values are not defined as the last value of each machine but rather as the mean failure value of a given time-window T to avoid possibly noisy evaluations, i.e.

$$\text{fv}(x_i) = \left(\sum_{t=1}^T \frac{1}{T} x_i^{(r)}(-t) \right)_{r=1, \dots, R}$$

The size of the window is application specific and has to be defined by the user. The same applies for start values, i.e.

$$\text{sv}(x_i) = \left(\sum_{t=1}^T \frac{1}{T} x_i^{(r)}(t) \right)_{r=1, \dots, R}$$

Mathematically, prognosability can be expressed as

$$P(x_i) = \exp\left(\frac{-\text{std}(\text{fv}(x_i))}{\text{mean}(|\text{fv}(x_i) - \text{sv}(x_i)|)}\right) \quad (9)$$

$$= \exp\left(\frac{-\sqrt{\frac{1}{N} \sum_{r=1}^R \left(\sum_{t=1}^T \frac{1}{T} x_i^{(r)}(-t) - \mu_f\right)^2}}{\frac{1}{R} \sum_{r=1}^R \left|\sum_{t=1}^T \frac{1}{T} x_i^{(r)}(-t) - \sum_{t=1}^T \frac{1}{T} x_i^{(r)}(t)\right|}\right) \quad (10)$$

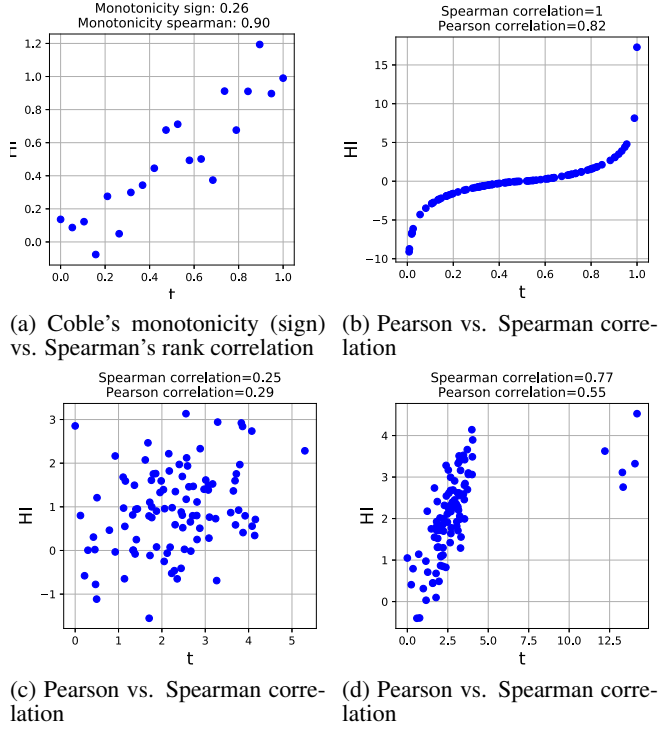


Figure 1. (a) Monotonicity computed with number of positive/negative slopes (Coble & Hines, 2009) is more sensitive to noise than spearman's correlation. (b) Spearman correlation gives perfect correlation even if the relationship is non-linear. (c) For mostly uncorrelated data Spearman correlation and Pearson correlation give similar results. (d) Spearman correlation is less sensitive to strong outliers in the tails.¹

where $\mu_f = \frac{1}{R} \frac{1}{T} \sum_{r=1}^R \sum_{t=1}^T x_i^{(r)}(-t)$.

3.1.3. Trendability

The trendability metric is computed in the same way as in Coble & Hines (2009) i.e. measuring that a feature has the same underlying shape by computing the correlation between pairs of machines. However, we choose to take the mean value instead of the minimum value of the correlations. The reason behind this choice is that taking a minimum value could emphasize potentially odd machines or behaviors. However, taking the minimum could also result in a more conservative choice, which might be wanted in some applications. In the end, the choice should therefore be left to the designer and in this paper, we choose to use a mean value. Mathemat-

ically, the trendability can thus be expressed as

$$T(x_i) = \frac{2}{R(R-1)} \sum_{r=1}^{R-1} \sum_{s=r+1}^R \left| \text{corr}(x_i^{(r)}, x_i^{(s)}) \right| \text{ for } r, s = 1, \dots, R \quad (11)$$

To compute the correlation coefficient $\text{corr}(x_i^{(r)}, x_i^{(s)})$ where $x_i^{(r)}$ and $x_i^{(s)}$ are time series with different lengths, several strategies will be compared:

- Resample the time series to the same length with one of the three following solutions and compute the redundancy via the absolute correlation:
 - *Resample long* strategy: Upsample the shortest time series to match the longest-one. This is done in three steps. First, the time index of both series is mapped to 0-1 in the life percentage space. Then, the shortest time series is upsampled via linear interpolation to the same number of samples as the longest one. Finally, the correlation can then be computed as usual.
 - *Resample 100* strategy: Resample the two time series to 100 samples. This is the strategy that was used by Coble, J. B. (2010). This is also done in three steps. First, the time index of both series is mapped to 0-1 in the life percentage space. Then, both time series are resampled to exactly 100 samples via a moving average window (each sample then represents 1% of lifetime). Finally, the correlation can then be computed as usual.
 - *History removed* strategy: Truncate the longest time series by removing the samples furthest away from the failure. Note that for this strategy to make sense, both series must have the same sampling rate.
- *DTW* strategy: Keep the time series with different lengths and use the Dynamic Time Warping algorithm (DTW) to compute the distance between the two time series. Dynamic time warping is a technique for comparing time series that computes a distance insensitive to local compression and stretches (Giorgino et al., 2009). The algorithm seeks for a warping which optimally deforms one of the two input series onto the other with certain restrictions:
 - Every sample from one sequence must match with one or more samples from the other sequence
 - The first sample from one sequence must match with the first sample from the other sequence
 - The last sample from one sequence must match with the last sample from the other sequence
 - The mapping of the samples from one sequence to the other must be monotonically increasing

¹The code and synthetic data used to generate the plots was inspired from author *Skbkekak* on https://commons.wikimedia.org/wiki/File:Spearman_fig1.svg

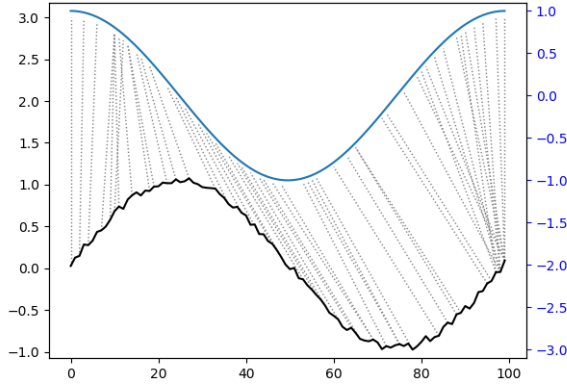


Figure 2. Dynamic Time Warping between two time series²

The distance between the two series is computed, after stretching, by summing the distances of each matched pair of elements (see example in Figure 2). Mathematically, it can be formulated as

$$d_\phi(X, Y) = \sum_{k=1}^T d(\phi_x(k), \phi_y(k))m_\phi(k)/M_\phi \quad (12)$$

where ϕ_x and ϕ_y are the warping functions that remap the time indices of X and Y respectively, $m_\phi(k)$ is a per-step weighting coefficient and M_ϕ is the normalization constant, which ensures that the accumulated distortions are comparable for different series. For a detailed overview of the algorithm, we refer to Giorgino et al. (2009).

To obtain a score that reflects redundancy with a value between 0 and 1, we take the exponential of the negative distance $\exp(-d_\phi(x_i^{(r)}, x_i^{(s)}))$. The negative term in the exponential is due to the inversely proportional relation between distance and redundancy (a low distance means a high redundancy). The trendability computation thus becomes:

$$T(x_i) = \frac{2}{R(R-1)} \sum_{r=1}^{R-1} \sum_{s=r+1}^R \left(\exp(-d_\phi(x_i^{(r)}, x_i^{(s)})) \right) \quad \text{for } r, s = 1, \dots, R \quad (13)$$

From an intuitive point of view and aside from being able to handle varying length time series, the DTW distance is an interesting measure that allows mapping degradations that occur at different times in different machines. If the degradations show the same underlying trend, the

measure will still result in a low distance, and thus a high trendability, even if those degradations did not occur simultaneously.

3.1.4. Single metric for feature relevance: fitness score

To obtain a unique score that quantifies the relevance of a prognostic feature, a fitness score is defined, which is a weighted average of the three metrics mentioned above. It is defined as

$$F(x_i) = w_1 \cdot M(x_i) + w_2 \cdot T(x_i) + w_3 \cdot P(x_i) \quad (14)$$

where w_1, w_2, w_3 are the weights associated to each metric. In this paper we choose an equal contribution for each metric, i.e. $w_1 = w_2 = w_3 = 1/3$. Note that for each metric to contribute roughly equally to the fitness score, we must further normalize them to spread equally among the range (0-1) with a min-max scaling :

$$m_{\text{scaled}}(i) = \frac{m(i) - \min(m(i))}{\max(m(i)) - \min(m(i))}$$

where $m(i)$ is the metric value associated to feature i .

3.2. Taking redundancy into account: prognostic mRMR

Since we assume we are faced with a predictive maintenance application with unknown class labels, a modification to the conventional mRMR algorithm presented in section 2.2 is needed.

We adapt the mRMR formulation in eq. (2-3) where the relevance criterion (mutual information between the feature and class label) is replaced by the fitness score and the redundancy criterion (mutual information between pairs of features) can be interchanged with different measures such as correlation or dynamic time warping. Let \mathcal{HI} be the set of all possible features and let $S \subseteq \mathcal{HI}$ denote the subset of features we are trying to identify, then equations (2-3) now become:

$$\max_S D(S) \triangleq \frac{1}{|S|} \sum_{i=1}^{|S|} \text{rel}(x_i) \quad (15)$$

$$\min_S R(S) \triangleq \frac{1}{|S|(|S|-1)} \sum_{i=1}^{|S|} \sum_{j=1, j \neq i}^{|S|} \text{red}(x_i, x_j) \quad (16)$$

where the relevance is defined by the fitness score, i.e. $\text{rel}(x_i) = F(x_i)$ and $\text{red}(x_i, x_j)$ is a measure of redundancy between features i and j . The reformulation of the objective problem as a sum or quotient remains the same as in eq. (4) and (5) as well as the incremental search methods defined by eq. (6,7).

One can also define weights associated with each objective D and R and seek for optimal ones. We leave this issue for future work and keep unit weights for both D and R . For the contributions in D and R to be similar, we also scale the

²The figure was taken from <https://dynamicimewarping.github.io/python/>

fitness score and redundancy score via a min-max scaling, i.e.

$$D_{scaled}(S) = \frac{D(S) - \min_{i \in S} D(S)}{\max_{i \in S} D(S) - \min_{i \in S} D(S)} \quad (17)$$

$$R_{scaled}(S) = \frac{R(S) - \min_{i \in R} R(S)}{\max_{i \in S} R(S) - \min_{i \in S} R(S)} \quad (18)$$

for a fair selection process. However, note that the shift (subtraction on the numerator) in the OFD case, and the scaling (division in the denominator) in the OFQ case do not impact the outcome of the optimization.

For the redundancy criterion, the mutual information is in general used to compare features with each other. However, as mentioned in Ding & Peng (2005), the absolute value of Pearson's correlation can also be used for continuous variables. In this article, we will compare both measures as well as the dynamic time warping.

To compute the mutual information and estimate the probability distributions, we rely on a non-parametric method based on entropy estimation from k -nearest neighbors' distance. We use the implementation from *scikit-learn* which is based on the algorithms presented in Kraskov et al. (2004) and Ross (2014). To obtain a value between zero and one and thus be able to compare it directly to the usual correlation coefficient, a transformation is performed:

$$\text{corr}_g(x, y) = \sqrt{1 - e^{-2MI(x, y)}} \quad (19)$$

We can show that if x, y are normally distributed with correlation ρ , then $MI(x, y) = \frac{1}{2} \log(1 - \rho^2)$ so that $\text{corr}_g(x, y) = \rho$ (Gel'Fand & Yaglom, 1959).

The third redundancy measure is based on dynamic time warping, which is also used for the computation of the trendability metric in section 4.2.1. In Radovic et al. (2017), the authors use the inverse of the dynamic time warping distance as a measure of redundancy for temporal gene data. However, this does not ensure the measure to be between 0 and 1. Instead, we reuse the same approach as for the DTW based trendability, i.e. by computing the redundancy measure as $\exp(-d_\phi(x_i, x_j))$ with d_ϕ defined in eq. (12). The redundancy criteria are then averaged across all run-to-failure time series. Mathematically, this is

$$\text{red}_{corr}(x_i, x_j) = \frac{1}{R} \sum_{r=1}^R \left| \text{corr} \left(x_i^{(r)}, x_j^{(r)} \right) \right| \quad (20)$$

$$\text{red}_{MI}(x_i, x_j) = \frac{1}{R} \sum_{r=1}^R \sqrt{1 - e^{-2MI(x_i^{(r)}, x_j^{(r)})}} \quad (21)$$

$$\text{red}_{dtw}(x_i, x_j) = \frac{1}{R} \sum_{r=1}^R \exp \left(-d_\phi \left(x_i^{(r)}, x_j^{(r)} \right) \right) \quad (22)$$

| Type | Choices |
|------------|---|
| Relevance | $F(x_i) = w_1 \cdot M(x_i) + w_2 \cdot T(x_i) + w_3 \cdot P(x_i)$ with $T(x_i)$ computed with <ul style="list-style-type: none"> • Correlation (eq. 11) and resample long strategy • Correlation (eq. 11) and resample 100 strategy • Correlation (eq. 11) and history removed strategy • Dynamic time warping (eq. 13) |
| Redundancy | <ul style="list-style-type: none"> • Correlation (eq. 20) • Mutual information (eq. 21) • Dynamic time warping (eq. 22) • Relevance only (redundancy not taken into account) |
| Objective | <ul style="list-style-type: none"> • OFD (eq. 4) • OFQ (eq. 5) |

Table 1. Design choices for selecting features with the prognostic mRMR algorithm.

As mentioned in section 2.2, exact solutions to the feature selection quickly become intractable. Instead, a heuristic is performed where one feature that maximizes one of the two formulations above (OFD or OFQ) is added at a time. The first feature chosen is the feature with the highest fitness score, and the second feature is the one that maximizes one of the two formulations above. We continue adding features until a predefined number of features is obtained. Algorithm 1 fully describes the proposed heuristic and Table 1 summarizes the different design choices of the algorithm.

4. APPLICATION TO A ROTATING MACHINE

In this section, the prognostic mRMR feature selection is applied to predict incoming failures in a high-speed rotating condenser. Section 4.1 describes the case study and the evaluation of the performance of the feature selection. Section 4.2 compares the different relevance and redundancy measures of the algorithm and section 4.3 compares our method with existing techniques proposed in the literature.

4.1. Problem description

4.1.1. Test case and features

The predictive maintenance case study is a high-speed rotating condenser (RotCo) modulating the RF frequency inside a cyclotron (Kleven et al., 2013). The RotCo is composed of a stator and a rotor with eight blades and rotates at a constant speed of 7500 RPM with the help of ball bearings. A picture of the system is shown in Figure 3. Several sensors are placed inside the machine to gather data. An accelerom-

input : Dataset: $x = \{x^{(1)}, \dots, x^{(R)}\}$ where $x^{(r)} \in \mathbb{R}^{n_r \times N}$, features (list of name of the features), $w_1 = w_2 = w_3 = \frac{1}{3}$, trendability_method, redundancy_method (red_{corr}, red_{MI} or red_{dtw} via eq. (20-22)), n_f (number of features to keep), objective (OFD via eq. (4) or OFQ via eq. (5))

```

rel = [] // relevance: empty array
red_mat = 0 // redundancy matrix of size
            N x N initialized with zeros
ranked_features = [] // empty array
for i in 1,...,size(features) do
    /* Relevance computation */
    m = M(xi) (via eq. 8) // monotonicity
    p = P(xi) (via eq. 9) // prognosability
    t = T(xi) (via eq. 11 or 13 depending on
                trendability_method) // trendability
    rel[i] = w1 ·  $\frac{m - \min(m)}{\max(m) - \min(m)}$  + w2 ·
              $\frac{p - \min(p)}{\max(p) - \min(p)}$  + w3 ·  $\frac{t - \min(t)}{\max(t) - \min(t)}$  (eq. 14)
    /* Redundancy computation */
    for j in i+1,...,size(features) do
        | red_mat[i, j] = redundancymethod(xi, xj)
    end
end
/* Heuristic search */
ranked_features.append(arg maxi rel) // append
the feature with maximum relevance to
ranked_features
features.pop(arg maxi rel) // Remove that
feature from the feature set
while size(features) < nf do
    score = [] (empty array)
    for f in features do
        | Ω = ranked_features ∪ f
        | D = rel[i]
        |  $R = \sum_{i=1}^{|\Omega|-1} \sum_{j=i+1}^{|\Omega|} \frac{2}{|\Omega|(|\Omega|-1)} \text{red\_mat}[i, j]$ 
        | score[f] = D - R if objective is OFD else
        | D/R if objective is OFQ
    end
    best_feature = arg maxf score
    ranked_features.append(best_feature)
    features.pop(best_feature)
end
return ranked_features

```

Algorithm 1: prognostic mRMR algorithm

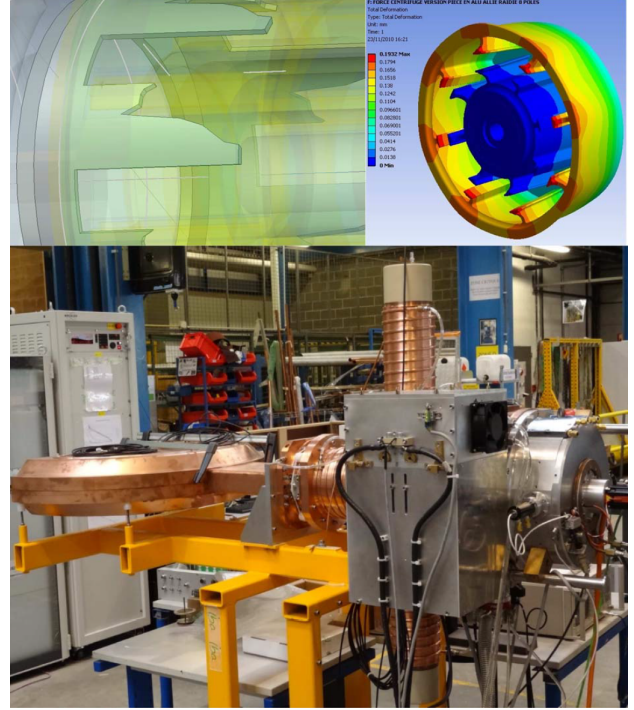


Figure 3. (bottom) RF system of the cyclotron with the rotating condenser on the right. (Top) Detailed view on the rotating condenser. (Kleeven et al., 2013) CC-BY-3.0

eter sensor is placed on the condenser to measure vibrations and performs 10-second acquisitions at a rate of 10kHz every hour. Other sensors are placed on the machine to gather data every second. Those include temperature sensors, vacuum pressure and a torque sensor. In total, we have gathered $R = 11$ run-to-failure time series.

After this data acquisition step, several health indicators (features) are constructed. For the vibration data, time-domain and frequency-domain features on each of the 10-second acquisition files are constructed. For the time-domain, those include Root mean square (RMS), Median absolute deviation (MAD) which is a robust measure of variability based on the deviations from the median, peak to peak values, skewness (third statistical moment) and kurtosis (fourth statistical moment). Those also include metrics based on the peaks of the signals: crest factor, clearance factor, shape factor, margin factor and max amplitude (for a detailed explanation on those features, the reader can refer to MathWorks (2021)). For the frequency-domain features, we construct the amplitudes at the fundamental frequency and its first 3 harmonics, the spectral power of all 20 Hz non-overlapping bands from 0-5kHz and finally the amplitudes at characteristic bearing frequencies (Schoen et al., 1995), i.e.

- Ball Pass Frequency Outer Race: $\frac{n_f}{2} \left(1 - \frac{D_b}{D_p} \cos \phi\right)$
- Ball Pass Frequency Inner Race: $\frac{n_f}{2} \left(1 + \frac{D_b}{D_p} \cos \phi\right)$

- Ball spin frequency: $\frac{D_p f}{2D_b} \left(1 - \left(\frac{D_b}{D_p} \cos \phi\right)^2\right)$
- Fundamental train frequency: $\frac{f}{2} \left(1 - \frac{D_b}{D_p} \cos \phi\right)$

where D_p is the pitch diameter, D_b the ball diameter, ϕ the contact angle and n the amount of balls. The first 3 harmonics of those characteristic frequencies are also included. For the non-vibration data, we perform aggregations of the signals over a 1-hour time-window to match with the vibration acquisition sampling. Those aggregations include the mean, max, min, standard deviation, skewness and kurtosis values. This finally results in $N = 317$ features (297 from vibration data and 24 from non-vibration data) computed every hour.

4.1.2. Evaluation of the feature selection's performance

The next step is the feature selection. While we can evaluate the best set of features with the algorithm developed (Algorithm 1) for a given method, we cannot conclude which one works better in practice nor how many features should be selected from the obtained ranked features.

Hence, the prediction problem is formulated as a binary classification where the machine is either in a healthy state or a faulty state and we compare the approaches and the number of features to be selected based on the classification score. However, in practice we do not know when the machine enters a faulty state. In this case, based on engineering expertise, we assume that the machine is likely to be in a faulty state about 5 days before the failure. Moreover, we assume that the machine is in a healthy state from the beginning of its life until 15 days prior to failure. 10 days of data for which we are the most unsure are thus excluded. This results in two artificial classes on which a classification can be performed. Note that in section 4.3.2, other labelling strategies, i.e. different than the 5-day time window, are tested.

The classification task is performed with a Support Vector Machine (SVM) algorithm using a RBF kernel (see e.g. Hearst et al. (1998)) which is a suitable classifier for this task. Furthermore, since there are only a few instances of failures in our dataset, a leave-one-out cross-validation is performed where each fold is defined as a run-to-failure time series. The classification score is averaged on the 11 folds of the dataset. No hyperparameter tuning is performed on the SVM as the goal of this article is not to obtain the best prediction capabilities but to compare different feature selection scenarios. The classification score chosen is the F_1 score which is the harmonic mean between the precision and recall, as it is a robust measure against imbalanced datasets (few failure data compared to healthy data).

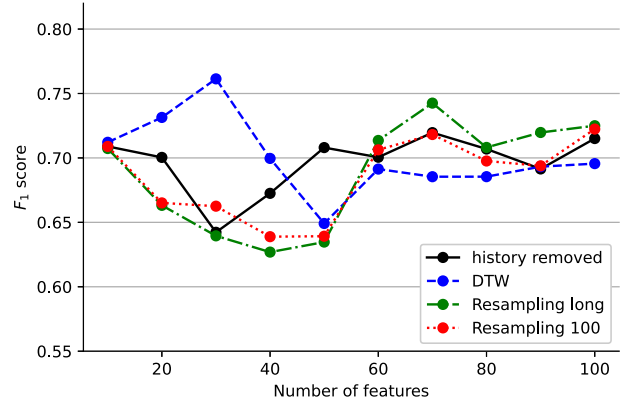


Figure 4. Comparison of the four variants to compute the trendability metric and its consequence on the relevance.

4.2. Comparing methods to compute the prognostic mRMR

In this section, the different methods for computing the prognostic mRMR algorithm are compared, as summarized by Table 1. Subsection 4.2.1 compares the relevance criteria associated to the different choices in the trendability metric. Subsection 4.2.2 compares the redundancy strategies and the final subsection compares the two objective function formulations.

4.2.1. Relevance measures

The different strategies to compute the trendability metric and hence to compute the relevance of the features are compared. We assume that we want to keep at most 100 features for computational, interpretability and stability reasons. For each number of selected features k between 10 and 100, we report on Figure 4 the cross-validated F_1 score associated to the selection of k best features in terms of their relevance score (see eq. 15). The process is repeated for the four strategies to compute the trendability as presented in section 3.

We observe no significant difference between the four approaches proposed except from 20 to 40 features selected, where the DTW approach is outperforming the others. Although a test for consistency on other data should be performed to confirm the trend, DTW seems to be a good candidate for comparing the features of different machines and hence for evaluating the relevance of a feature. In addition to the observed positive trend, DTW uses the time structure of the features and can map any instant in the time series of a particular machine to any other instant in another machine. Indeed, intuitively, the start of a degradation phase in a specific machine will most likely never occur at the exact same time in another machine. This is where the correlation measure (on which the other three methods are based) fails by only being able to compare pairs of points at equivalent indexes between two time series.

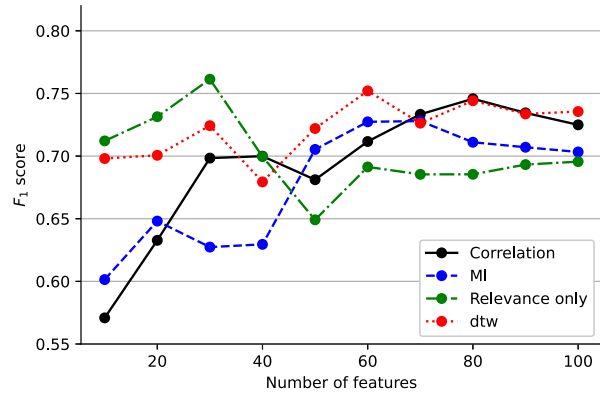
4.2.2. Redundancy measures

This section aims at improving the selection of features by considering the redundancy between features and comparing the proposed redundancy measures. In a first analysis we choose to use the DTW approach for the trendability and relevance computation as it resulted in the best score. We compare the different relevance-redundancy scenarios as well as the no redundancy scenario (based only on the relevance criterion) and choose the OFD as the objective function. The results are reported in Figure 5a. We observe that taking into account redundancy does not improve and actually decreases the effectiveness of our model for 10 to 40 features. For more features considered however, the relevance-only approach performs slightly worse than the others. This can be explained by the fact that the other approaches tend to select features that are sometimes not relevant only because they are highly independent from each other. An improved approach taking the best of both worlds is to preselect features that are at least above a certain relevance threshold or to define a threshold on the maximum number of features to preselect.

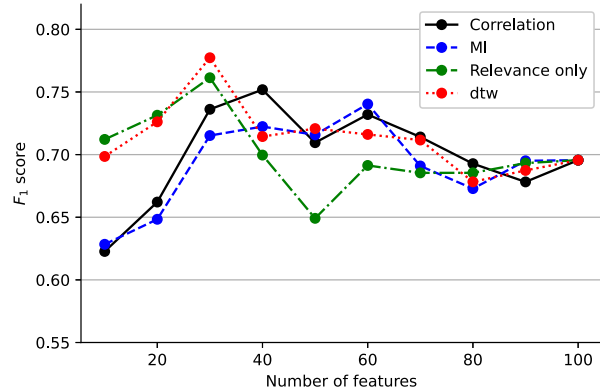
In the next analysis the 100 most relevant features are preselected according to their fitness score. The same comparison as in Figure 5a is performed with those 100 preselected features. The results are shown in Figure 5b. We can observe that the DTW approach now outperforms the relevance only approach by a small margin and reaches an overall maximum score of 0.78 for 30 selected features. For the correlation and mutual information approaches however, they still are not able to achieve better performance than the relevance only approach but they show better performance when 40 to 70 features are selected and similar performance when more than 70 features are selected. The conclusion from Figure 5 is that taking into account redundancy between features can definitely help, but a careful preselection should be done first to exclude highly irrelevant features. Moreover, as a recommendation, we propose to compute the redundancy between features via the DTW measure as it yields good and stable results.

4.2.3. Objective function formulations

Based on the best approach obtained so far (trendability and redundancy computed with DTW approach), we compare the two objective function formulations: OFD (eq. 4) and OFQ (eq. 5). The results are reported in Figure 6. We observe no impact on the outcome for the choice of the objective function formulation except for a slightly better performance with OFD when 30 features are selected, which is likely negligible. Since the conventional way to formulate the mRMR is via OFD, we propose to keep this formulation.



(a) Using all features.



(b) Using only the 100 most relevant features

Figure 5. Comparison of relevance-only approach (None in green) and the 3 mRMR approaches with trendability metric computed via DTW.

4.3. Comparison of our method with existing methods

This section compares our prognostic mRMR algorithm with existing feature selection methods proposed in the literature. The prognostic mRMR is computed with the measures that give the best results on the case study, i.e. with the trendability metric and redundancy measure computed via the dynamic time warping measure, OFD chosen as the objective function and a preselection of the best 100 features according to their fitness score. In section 4.3.1, we compare our approach with the feature selection based on the prognostic metrics of Coble, J. B. (2010). In section 4.3.2, we compare our approach with the conventional mRMR feature selection.

4.3.1. Comparison with feature selection based on the prognostic metrics from Coble

A comparison between our prognostic mRMR algorithm and the feature selection based on the three original metrics defined by Coble, J. B. (2010) is shown in Figure 7. We can

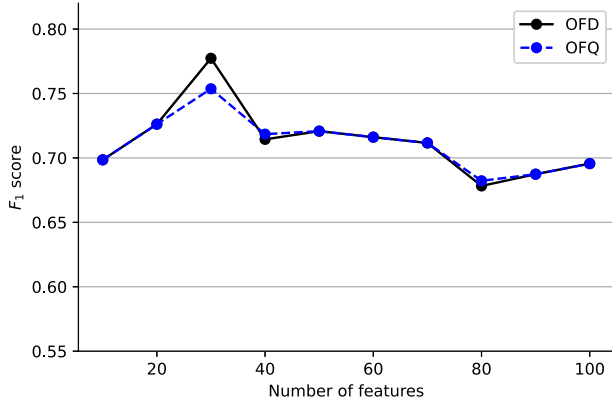


Figure 6. Comparison of the two objective function formulations: OFD (see eq. 4) and OFQ (see eq. 5)

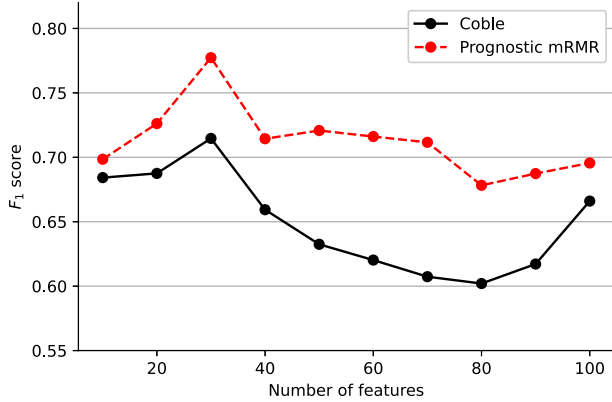


Figure 7. Comparison of the prognostic mRMR with the feature selection from Coble & Hines (2009).

clearly observe that our method selects a subset of features that is better able to discriminate between a faulty and healthy state (F_1 score is higher), regardless of the chosen number of features. This may be explained by two reasons. First, redundancy is taken into account in the prognostic mRMR approach while it is not in Coble's approach. Second, the prognostic metrics used in the prognostic mRMR are more robust than the ones in Coble. Indeed, this can be observed when comparing Coble's approach (the solid black line from Figure 7) with the relevance only version of our prognostic mRMR (green dotted line from Figure 5).

4.3.2. Comparison with the classical mRMR approach

The classical use of the mRMR algorithm requires to compute the relevance of the features based on the class labels, usually labelled machine malfunctions such as inner and outer ring defects in the case of bearings. In our application, such rich

labelling is not available but we can use the class label based on the window labelling described in section 4.1.2.

The labels defining a faulty state were defined somewhat arbitrarily based on engineering expertise. Hence, to fully compare the classical mRMR with our prognostic mRMR algorithm, we compare them for different labelling strategies: 10 different labelling strategies where the machine is considered in a faulty state starting n days before the failure where $n = 1, 2, \dots, 10$ are compared. The healthy state spans from machine installation time to 15 days prior to failure as detailed in section 4.1.2. For each labelling strategy, the cross-validated F_1 score is compared for a number of features ranging from 10 to 100. The results are outlined in Figure 8. To assess the two approaches, we either consider the global results by comparing the two curves, or refer to the maximal score attained for any number of features for each of the labelling strategies. For the latter, we simply need to compare the maximum point on each curve while for the former option, we can either compare the sum of differences (SD) between scores corresponding to the same number of features, or the number of times one curve is above the other (ND), based on the 10 scores computed for the increasing feature set size. Mathematically, this is:

$$SD = \sum_{i \in \{10, 20, \dots, 100\}} F_1^{\text{prognostic}}(i) - F_1^{\text{classical}}(i)$$

$$ND = \#\{i \in \{10, 20, \dots, 100\}: F_1^{\text{prognostic}}(i) > F_1^{\text{classical}}(i)\}$$

where $F_1^{\text{prognostic}}(i)$ and $F_1^{\text{classical}}(i)$ are the cross-validated F_1 score associated to the feature set of size i computed via the prognostic mRMR algorithm and the classical mRMR algorithm respectively. The prognostic mRMR should be superior to the classical mRMR if $SD > 0$ or $ND > 5$ for a particular labelling strategy. Table 2 summarizes the results and individual scores are outlined in Figure 8. From a global standpoint, the prognostic approach outperforms the classical approach in 9 out of 10 cases with respect to the SD metric and 7 out of 9 cases + 1 ex aequo with respect to the ND metric. If we only look at the maximum values, the best result is split among the two approaches (5 cases for both). Moreover, prognostic mRMR is consistently better with fewer features and thus is able to select the best compact set of features for the prognostic task.

5. CONCLUSION

We developed an unsupervised minimum redundancy maximum relevance feature selection method for predictive maintenance applications by adapting the conventional mRMR algorithm where the relevance of a feature is computed with respect to prognostic metrics instead of class labels. We also compared different measures to compute the redundancy between features and adapted existing metrics quantifying the

| Label [days] | SD | ND | max score |
|--------------|-------------|-----------|-------------------|
| 1 | 1.02 | 8 | prognostic |
| 2 | 0.69 | 5 | prognostic |
| 3 | 1.16 | 7 | prognostic |
| 4 | 0.71 | 7 | classical |
| 5 | 0.04 | 4 | classical |
| 6 | -0.08 | 3 | classical |
| 7 | 0.47 | 7 | classical |
| 8 | 0.46 | 6 | classical |
| 9 | 0.88 | 10 | prognostic |
| 10 | 0.74 | 8 | prognostic |

Table 2. Results of classical vs. prognostic mRMR. Results are highlighted in bold when the prognostic approach outperforms the classical mRMR approach. Prognostic mRMR is better if $SD > 0$ and $ND > 5$.

relevance of features. We performed a case study for a rotating machine that highlighted the superiority of our feature selection method compared to previous prognostic metrics and the conventional mRMR algorithm, especially for selecting a compact set of features. We also showed that dynamic time warping is a well-suited distance measure for predictive maintenance applications that can help to select a good set of features.

The approaches presented in this paper may still be improved by seeking the best parameters in the fitness metric characterizing the relevance of a feature as well as the weights assigned to the relevance and redundancy in the objective function, which we leave to future work.

ACKNOWLEDGEMENT

This work was supported by the Biowin BiDMed project funded by the Walloon Region. The authors are very thankful to the three reviewers whose insightful and constructive comments greatly helped improve the paper.

REFERENCES

- Camci, F., Medjaher, K., Zerhouni, N., & Nectoux, P. (2013). Feature evaluation for effective bearing prognostics. *Quality and reliability engineering international*, 29(4), 477–486.
- Carino, J. A., Zurita, D., Delgado, M., Ortega, J., & Romero-Troncoso, R. (2015). Remaining useful life estimation of ball bearings by means of monotonic score calibration. In *2015 IEEE International Conference on Industrial Technology (icit)* (pp. 1752–1758).
- Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1), 16–28.
- Coble, J., & Hines, J. W. (2009). Identifying optimal prognostic parameters from data: a genetic algorithms approach. In *Annual conference of the prognostics and health management society* (Vol. 27).
- Coble, J. B. (2010). *Merging data sources to predict remaining useful life—an automated method to identify prognostic parameters*. (PhD thesis)
- Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02), 185–205.
- Fernandes, M., Canito, A., Bolón-Canedo, V., Conceição, L., Praça, I., & Marreiros, G. (2019). Data analysis and feature selection for predictive maintenance: A case-study in the metallurgic industry. *International journal of information management*, 46, 252–262.
- Gel’Fand, I., & Yaglom, A. (1959). Calculation of the amount of information about a random function contained in another such function. *Eleven Papers on Analysis, Probability and Topology*, 12, 199.
- Giorgino, T., et al. (2009). Computing and visualizing dynamic time warping alignments in r: the dtw package. *Journal of statistical Software*, 31(7), 1–24.
- Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4), 18–28.
- Hu, Q., Si, X.-S., Qin, A.-S., Lv, Y.-R., & Zhang, Q.-H. (2020). Machinery fault diagnosis scheme using redefined dimensionless indicators and mrmr feature selection. *IEEE Access*, 8, 40313–40326.
- Javed, K., Gouriveau, R., Zerhouni, N., & Nectoux, P. (2013). A feature extraction procedure based on trigonometric functions and cumulative descriptors to enhance prognostics modeling. In *2013 IEEE Conference on Prognostics and Health Management (PHM)* (pp. 1–7).
- Javed, K., Gouriveau, R., Zerhouni, N., & Nectoux, P. (2014). Enabling health monitoring approach based on vibration data for accurate prognostics. *IEEE Transactions on Industrial Electronics*, 62(1), 647–656.
- Kleeven, W., Abs, M., Forton, E., Henrotin, S., Jongen, Y., Nuttens, V., ... others (2013). The IBA superconducting synchrocyclotron project S2C2. In *Proc. cyclotrons* (pp. 115–119).
- Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical review E*, 69(6), 066138.
- Lei, Y., Li, N., Gontarz, S., Lin, J., Radkowski, S., & Dybala, J. (2016). A model-based method for remaining useful life prediction of machinery. *IEEE Transactions on Reliability*, 65(3), 1314–1326.
- Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to rul prediction. *Mechanical Systems and Signal Processing*, 104, 799–834.

- Li, N., Lei, Y., Liu, Z., & Lin, J. (2014). A particle filtering-based approach for remaining useful life prediction of rolling element bearings. In *2014 international conference on prognostics and health management* (pp. 1–8).
- Li, Y., Yang, Y., Li, G., Xu, M., & Huang, W. (2017). A fault diagnosis scheme for planetary gearboxes using modified multi-scale symbolic dynamic entropy and mrmr feature selection. *Mechanical Systems and Signal Processing*, *91*, 295–312.
- Liu, Z., Zuo, M. J., & Qin, Y. (2016). Remaining useful life prediction of rolling element bearings based on health state assessment. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *230*(2), 314–330.
- Liu, Z., Zuo, M. J., & Xu, H. (2013). Fault diagnosis for planetary gearboxes using multi-criterion fusion feature selection framework. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *227*(9), 2064–2076.
- MathWorks. (2021). *Signal features - matlab & simulink*. Retrieved 2021-03-22, from <https://mathworks.com/help/predmaint/ug/signal-features.html>
- Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238. doi: 10.1109/TPAMI.2005.159
- Radovic, M., Ghalwash, M., Filipovic, N., & Obradovic, Z. (2017). Minimum redundancy maximum relevance feature selection approach for temporal gene expression data. *BMC bioinformatics*, *18*(1), 1–14.
- Ross, B. C. (2014). Mutual information between discrete and continuous data sets. *PloS one*, *9*(2), e87357.
- Schoen, R. R., Habetler, T. G., Kamran, F., & Bartfield, R. (1995). Motor bearing damage detection using stator current monitoring. *IEEE transactions on industry applications*, *31*(6), 1274–1279.
- Shahidi, P., Maraini, D., & Hopkins, B. (2016). Railcar diagnostics using minimal-redundancy maximumrelevance feature selection and support vector machine classification. *International Journal of Prognostics and Health Management*, *7*, 2153–2648.
- Solorio-Fernández, S., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2020). A review of unsupervised feature selection methods. *Artificial Intelligence Review*, *53*(2), 907–948.
- Tang, X., He, Q., Gu, X., Li, C., Zhang, H., & Lu, J. (2020). A novel bearing fault diagnosis method based on gl-mrmr-svm. *Processes*, *8*(7), 784.
- Wang, D., Tsui, K.-L., & Miao, Q. (2017). Prognostics and health management: A review of vibration based bearing and gear health indicators. *Ieee Access*, *6*, 665–676.
- Yan, X., & Jia, M. (2019). Intelligent fault diagnosis of rotating machinery using improved multiscale dispersion entropy and mrmr feature selection. *Knowledge-Based Systems*, *163*, 450–471.
- Zhang, B., Zhang, L., & Xu, J. (2016). Degradation feature selection for remaining useful life prediction of rolling element bearings. *Quality and Reliability Engineering International*, *32*(2), 547–554.
- Zhang, X., Song, Z., Li, D., Zhang, W., Zhao, Z., & Chen, Y. (2018). Fault diagnosis for reducer via improved lmd and svm-rfe-mrmr. *Shock and Vibration*, *2018*.
- Zhao, X., Zuo, M. J., Liu, Z., & Hoseini, M. R. (2013). Diagnosis of artificially created surface damage levels of planet gear teeth using ordinal ranking. *Measurement*, *46*(1), 132–144.

BIOGRAPHIES

Valentin Hamaide is a PhD student in the ICTTEAM institute of the Université Catholique de Louvain. He received his master's degree in mathematical engineering in 2017 in the same university. His research interests are machine learning and optimization with applications in the medical domain.

François Glineur received dual engineering degrees from Université de Mons and CentraleSuplec in 1997, and a PhD in Applied Sciences from Université de Mons in 2001. He visited Delft University of Technology and McMaster University as a post-doctoral researcher, then joined Université catholique de Louvain where he is currently a professor of applied mathematics at the Engineering School, member of the Center for Operations Research and Econometrics and the Institute of Information and Communication Technologies, Electronics and Applied Mathematics. His research interests focus on optimization models and methods (mainly convex optimization and algorithmic efficiency) and their engineering applications, as well as on nonnegative matrix factorization and applications to data analysis.

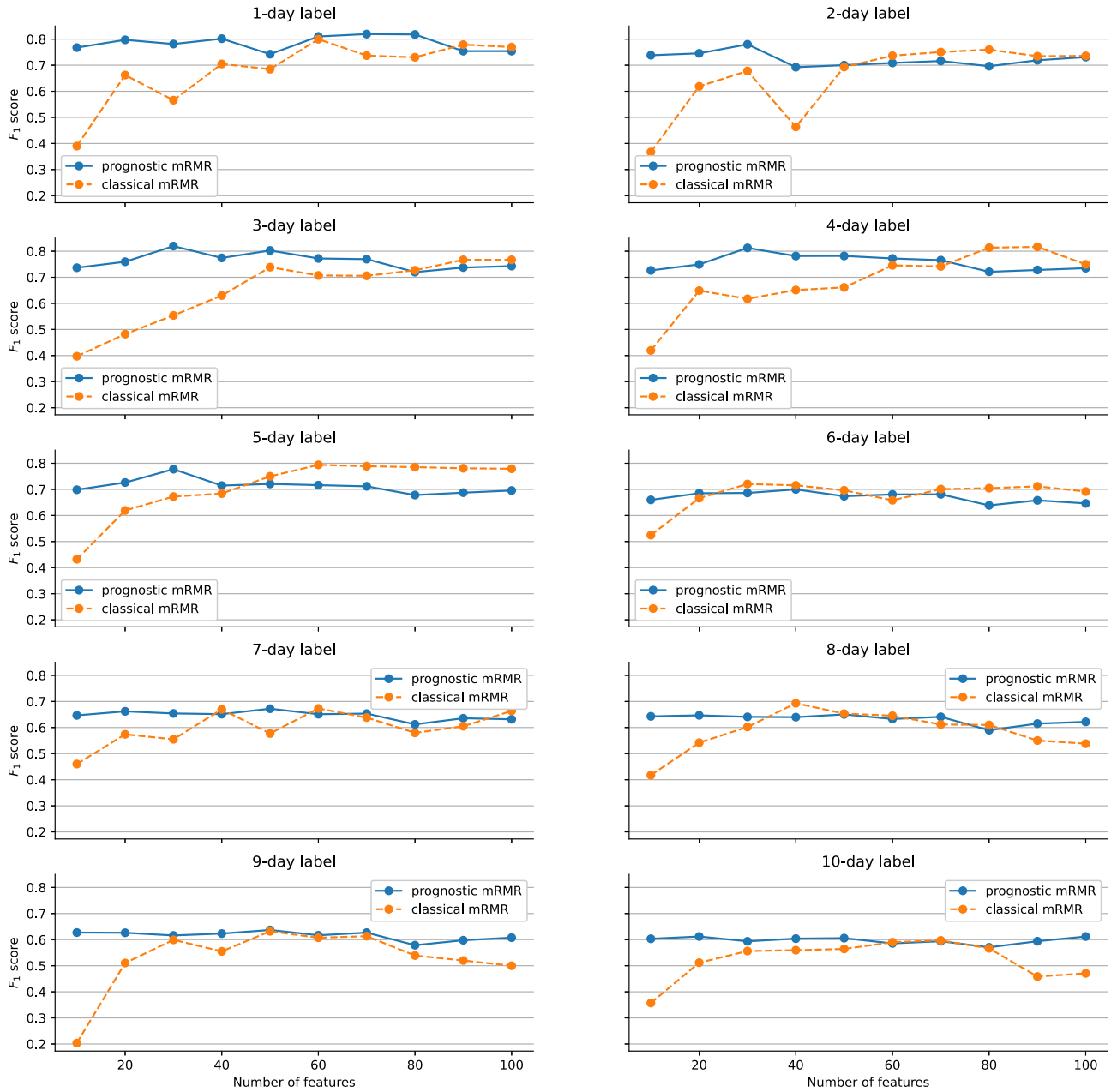


Figure 8. Classical vs. prognostic mRMR feature selection for various labelling strategies.