Recommendations for Data-Driven Degradation Estimation with Case Studies from Manufacturing and Dry-Bulk Shipping

Nils Finke[1,2,*], Marisa Mohr[1,3,⋆], Alexander Lontke[3,4], Marwin Züfle[4], Samuel Kounev[4], and Ralf Möller[1]

[1] University of Lübeck, Institute of Information Systems, Lübeck, Germany
{finke,mohr}@ifis.uni-luebeck.de
[2] Oldendorff Carriers GmbH & Co. KG., Lübeck, Germany
[3] inovex GmbH, Hamburg, Germany
[4] University of Würzburg, Department of Computer Science, Würzburg, Germany

Abstract. Predictive planning of maintenance windows reduces the risk of unwanted production or operational downtimes and helps to keep machines, vessels, or any system in optimal condition. The quality of such a data-driven model for the prediction of remaining useful lifetime is largely determined by the data used to train it. Training data with qualitative information, such as labeled data, is extremely rare, so classical similarity models cannot be applied. Instead, degradation models extrapolate future conditions from historical behaviour by regression. Research offers numerous methods for predicting the remaining useful lifetime by degradation regression. However, the implementation of existing approaches poses significant challenges to users due to a lack of comparability and best practices. This paper provides a general approach for composing existing process steps such as health stage classification, frequency analysis, feature extraction, or regression models for the estimation of degradation. To challenge effectiveness and relations between the steps, we run several experiments in two comprehensive case studies, one from manufacturing and one from dry-bulk shipping. We conclude with recommendations for composing a data-driven degradation estimation process.

Keywords: Remaining Useful Lifetime · Bearing · Vessel Performance

1 Introduction

Data-driven products and machine learning methods offer large benefits for production engineering companies. Predictive maintenance can reduce the risk of unwanted production and operational downtime and help keep machines, vessels, and systems in optimal condition. A key challenge of this is the estimation of remaining useful lifetime (RUL), that is, predicting the time to failure. However, the development of such products requires a large initial investment in the model

* Nils Finke and Marisa Mohr contributed equally to this work.

definition and training data acquisition. The latter is especially important, as the prediction quality of a machine learning model is largely determined by the data used for training. Labeled data or large amounts of observed run-to-failure data are extremely rare. Of course, one could deliberately degrade machines to capture more failure patterns, but that is at least financially irresponsible.

One way to model RUL without having labeled or entire failure data from similar machines is to use degradation models. Degradation models estimate the RUL only indirectly by relating the degradation of parts of the product itself to the failure mechanisms. Degradation analysis allows the analyst to extrapolate to an assumed failure time based on measurement of time series performance or sensor data directly related to the suspected failure of the machine under consideration. An initial evaluation of appropriate data that give an indicator of degradation presents an initial challenge. However, after initial investment, one also benefits from a prediction of intermediate states up to the failure itself.

Research provides numerous methods for modelling degradation and RUL. To decide on an appropriate approach, there are few or insufficient comparisons of existing methods. To help deciding on a solution for real-world challenges, one needs a mechanism to compare existing methods. To demonstrate feasibility, one is interested to setup a basic solution before improving the overall approach.

In this paper, we present a general data-driven approach for predicting RUL that considers comparability of existing approaches in the best possible sense. This approach includes four steps: health stage (HS) classification, frequency analysis, feature extraction, and the prediction itself performed by regression. By means of the approach, we focus on four general research questions that arise in the search for an appropriate modelling method:

1. Can HS classification improve the accuracy of prediction?
2. Does the frequency spectrum of a time series provide more useful information than the raw data, i.e., time spectrum, itself?
3. Which feature sets are appropriate for the estimation of degradation?
4. Which data-driven regression method yields the highest accuracy?

For general validity and comparability, we present two comprehensive case studies in different industries, namely manufacturing and dry-bulk shipping. The aim of this work is not to achieve the best possible predictive accuracy. Instead, we investigate the interaction of the steps and conclude with recommendations for the composition of a data-driven degradation estimation process.

2 Remaining Useful Lifetime Prediction

In this section we place our work in the context of RUL prediction, and present related work.

2.1 Modelling the Remaining Useful Lifetime

Depending on the type of measurement data, three different model families are applied. The different families of data-driven models for predicting RUL are

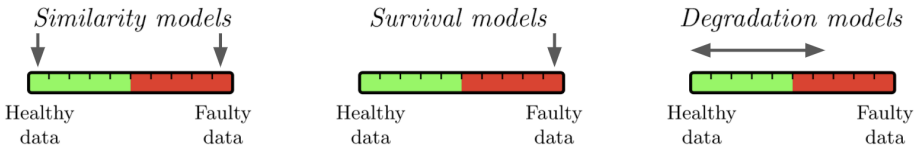

Fig. 1. Three families of models for the prediction of RUL.

visualised in Figure 1, with arrows indicating the types of training data available. *Similarity models* use run-to-failure data from similar machines, starting during healthy operation and ending close to failure or maintenance. RUL is directly estimated from historical labeled training data by applying a pattern matching of trends or conditional indicator values. *Survival models* are used when the user does not have a complete history of run-to-failure data but instead has data about the life span of related components. Probability distributions are determined based on the behaviour of related components and used to estimate RUL. *Degradation models* estimate the degradation process without requiring faulty data. Historical behaviour of a machine condition indicator is used to extrapolate the damage progression to indirectly determine RUL.

In real-world challenges complete run-to-failure data are rarely available, we focus on degradation models. We use data-driven statistical methods being suitable when little domain knowledge is available or generalised models are desired.

2.2 Related Work

Research provides many approaches for the estimation of degradation processes. In regular operation healthy data outweighs degradation data, so data-driven prediction is often challenged by imbalance. To address imbalance, an additional preprocessing step, such as HS classification can be used. To distinguish between healthy and faulty data, different classification indicators from kurtosis to self-organising maps are applied before model training, e.g., in [8,12,14,15,20,24]. To gain other information further preprocessing by frequency analysis are performed before extracting features for degradation regression. Examples range from classical discrete Fourier transform (DFT), short-time Fourier transform (STFT) to Hilbert-Huang transform [4,6,9,11,14,21]. Since most classical data-driven models cannot directly process time series, the extraction of additional scalar-valued features from time series is necessary before these algorithms can be applied.

Feature extraction performed using feature engineering methods range from classical statistical measurements such as root mean square and kurtosis [2,6,20] to information-theoretic entropies [5,25,21,11]. Other authors provide feature learning methods based on isomap [4], autoencoder [9] or convolutional neural networks (CNN) [14]. Data-driven models for degradation estimation are implemented, e.g., by polynomial regression (PR) [13,23], support vector regression (SVR) [10], or artificial neural networks (ANN) for regression [21].

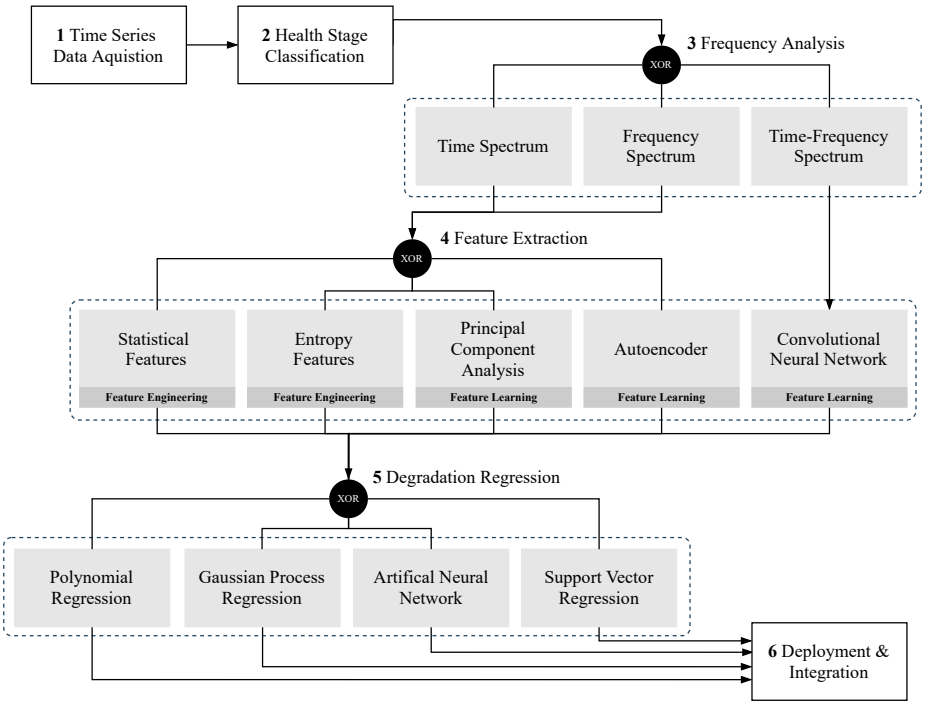

Fig. 2. Six technical steps of RUL prediction (2 and 3 are optional).

Besides on preprocessing steps, prediction accuracy depends on the choice of features and the regression algorithm used. Comparisons are either available for feature sets based on different selection indicators [25,23] or for data-driven methods for RUL regression [7,19]. However, none of the comparisons take into account the interaction of preprocessing steps, feature extraction, and regression algorithms at once. In two comprehensive case studies, we strengthen the understanding and effectiveness of the different steps as well as their interactions.

3 A Data-Driven Approach for Degradation Estimation

We present a general approach for comparison of the different steps for the estimation of degradation. In general, the degradation estimation process consists of six technical steps, i.e., time series data acquisition, HS classification, frequency analysis, feature or indicator extraction, degradation estimation by regression, and deployment and integration as presented in Figure 2. We focus on steps 2 to 5, as data acquisition, and appropriate deployment, and integration of the predictive model depend on both domain and user's system infrastructure.

Next, we follow Figure 2 by addressing the steps before discussing them as part of two case studies to answer the introduced research questions.

3.1 Health Stage Classification

The second step in the overall process visualised in Figure 2 is considered optional. Caused by the fact that healthy data outweighs degradation data in regular operation, data-driven prediction of the degradation process can be impeded or even biased by healthy data. In order to distinguish between healthy and faulty stages in a time series, the point in time when the degradation starts has to be identified. The boundary of the two stages is called the first prediction time (FPT). For simplicity, in this work we include an approach by Li et al. [12], where kurtosis is used as such a classification indicator. The FPT corresponds to the time when the kurtosis of a sliding window over the time series exceeds the interval $\mu \pm 2\sigma$ for the second time, where μ is the mean and σ is the standard deviation at the beginning of the time series. After the identification of FPT, observations in data classified as healthy are omitted from both training and prediction of degradation. The prediction by the regression model is initiated when data is classified as unhealthy. To answer our first research question, we evaluate in Section 4 whether this additional step can improve the accuracy of the estimation of degradation by adding the HS classifier.

3.2 Frequency Analysis

The third step in the overall process visualised in Figure 2 is the analysis of the frequency range of a time series that can provide further insights into the degradation process. In this step we distinguish between time spectrum, frequency spectrum and time-frequency spectrum analysis. By *time spectrum*, we denote the raw time series on which no frequency analysis is performed. By *frequency spectrum*, we denote a time series that is transformed by discrete Fourier transform (DFT). DFT transforms a finite sequence of equally-spaced observed data points $(x_0, ..., x_T)$ into another sequence $(X_0, X_1, \ldots, X_T)$ that is a complex-valued function of frequency The fast Fourier transform (FFT) is an efficient algorithm for computing DFT. Showing a trend, degradation time series are inherently non-stationary, i.e., the mean is not constant over time. To analyse the frequency spectrum of non-stationary time series, short-time Fourier transform (STFT) is used. To assume stationarity, the STFT uses a window function to select short time periods with constant mean. Several frequency spectra are calculated per window by DFT. By *time-frequency spectrum*, we denote a time series on which STFT is performed. Note that in the next step of the overall process, not every feature extraction method can be applied on every frequency analysis method. Implementation details follow in Section 4.

3.3 Feature Extraction

A model cannot represent information that it does not have. The extraction of features in step four visualised in Figure 2 refers to the creation of new information that was previously not available. Techniques for feature extraction can be classified into two groups, namely feature engineering and feature learning.

Table 1. List of statistical features.

Mean	$\bar{x} = \frac{1}{T}\sum_{i=1}^{T} x_i$	Skewness	$\frac{\frac{1}{T}\sum_{i=1}^{T}(x_i-\bar{x})^3}{(\frac{1}{T}\sum_{i=1}^{T}(x_i-\bar{x})^2)^{\frac{3}{2}}}$
Max	$\max\{x_1, ..., x_T\}$	Kurtosis	$\frac{\frac{1}{T}\sum_{i=1}^{T}(x_i-\bar{x})^4}{(\frac{1}{T}\sum_{i=1}^{T}(x_i-\bar{x})^2)^2}$
Min	$\min\{x_1, ..., x_T\}$	Peak factor	$\frac{max(x)}{\sqrt{\frac{1}{T}\sum_{i=1}^{T} x_i^2}}$
Root mean square	$\sqrt{\frac{1}{n}\sum_{i=1}^{T} x_i^2}$	Change coefficient	$\frac{\bar{x}}{\sqrt{\frac{1}{T}\sum_{i=1}^{T} x_i^2}}$
Peak to peak value	$\max(x) - \min(x)$	Clearence factor	$\frac{max(x)}{\frac{1}{T}\sum_{i=1}^{T}(x_i)^2}$
Variance	$\frac{1}{T}\sum_{i=1}^{T}(x_i - \bar{x})^2$	Absolute Energy	$\sum_{i=1}^{T} x_i^2$

Feature Engineering is the older discipline of the two. New features are created by processing domain-specific knowledge or by transforming data. Techniques for feature engineering origin from at least two research areas. The first way to extract features is by means of statistical analysis. A list of the *statistical features* for a univariate time series $x \in \mathbb{R}^T$ used in this work is given in Table 1. Another way of extracting features is by using information-theoretic measurements, called *entropies*. The concept of entropy was first introduced by Claude Elwood Shannon in 1948 and has since been used to quantify the complexity of data in numerous other fields. (Shannon) entropy is defined as the expected number of bits needed to encode a message that is $H = -\sum_{z \in Z} p_z \log_2(p_z)$, where Z is the set of possible symbols used in a message, p_z is the probability of $z \in Z$ appearing in a message. The number of bits required is in direct relation to the complexity (and entropy) of the message, meaning few or many bits reflect a low or high entropy, respectively. To use entropies as features for time series, observations are encoded as sequences of symbolic abstractions. As far as current research is concerned, there are two general approaches of symbolisation [17]. Classical symbolisation approaches use data range partitioning and thresholds for symbol assignment such as the well-known Symbolic Aggregate approXimation (SAX). The ordinal pattern symbolisation approach, describing the up and downs in a time series, is based on an approach by Bandt and Pompe [3]. Combining the ordinal pattern symbolisation approach with Shannon entropy leads to a special case called permutation entropy. All listed features can be applied directly to time and frequency spectrum.

Feature Learning compared to feature engineering, solve optimisation problems to learn features from a set of time series. Learned features can reveal task-specific patterns that are not obvious to humans, including non-linear patterns. There are numerous ways to learn features as principal component analysis, autoencoders, and convolutional neural networks.

The *principal component analysis (PCA)* is a well-known method converting a set of observations of possibly correlated variables $X \in \mathbb{R}^{n \times p}$ into a set of values of linearly uncorrelated variables $X' \in \mathbb{R}^{n \times p}$. Using eigenvalue analysis,

an orthogonal transformation that preserves greatest variance in data yields in new p basis vectors, also called principal components. Keeping only the first r principal components gives the truncated transformation $X'_r = XW_r$, where $W \in \mathbb{R}^{p \times r}$ is a matrix whose columns are the eigenvectors of $X^T X$ sorted in descending order of the r highest corresponding eigenvalues, and a new lower-dimensional representation of the data.

A relatively new method for reducing dimensionality are *autoencoders*, a branch of ANNs. The architecture consists of two connected ANNs compressing the input variable into a reduced dimensional space, also called encoder, and re-creating the input data, also called decoder. Each node of the hidden "bottleneck" layer of compressed information can be treated as a feature in subsequent learning tasks, just as the selected principle components. The autoencoder as well as PCA can be applied directly to the time and frequency spectrum.

A *convolutional neural network (CNN)* is another type of ANNs typically used for image recognition, but also for signal processing. The architecture of a classical CNN consists of one or more convolutional layers followed by a pooling layer. In a convolutional layer, a matrix, also called filter kernel, is moved stepwise over the input data calculating the inner product of both. The result is called feature map. Accordingly, neighbouring neurons in the convolutional layer correspond to overlapping regions such as similar frequencies in signals. In a pooling layer, superfluous information is discarded and a more abstract lower-dimensional representation of the relevant information is obtained by combining neighbouring elements of the map, e.g., by calculating the maximum. To feed the matrix output of the convolution layer and the pooling layer into a final fully connected layer, it must first be unrolled (flattened). The flatten layer is then treated as a feature. The CNN has to be applied to the time-frequency spectrum. Implementation details for all feature extraction methods are listed in Section 4.

3.4 Degradation Regression

Regression models, as one of the most popular data-driven techniques for RUL prediction, fit available degradation data by regression functions and extrapolate the future progression. We consider the following regression models: multiple linear regression, Gaussian process regression, artificial neural network regression, and support vector regression.

Multiple linear regression (MLR) is a statistical technique that fits an observed dependent variable by several independent variables using the method of least squares. More precisely, the coefficients of a linear function $y_t = x_{t1}w_1 + x_{t2}w_2 + \cdots + x_{tK}w_K + \varepsilon_t = \mathbf{x}_t^\top \boldsymbol{w} + \varepsilon_t, t = 1, 2, \ldots, T$, are estimated, where y is the response variable, x_K are the predictors, and w the coefficients of the model.

In a traditional regression model, we infer a single function, $Y = f(X)$. In *Gaussian process regression (GPR)*, we place a Gaussian process over $f(X)$. A Gaussian process (GP) is a collection of random variables, of which any finite subset of random variables is Gaussian distributed. It is completely specified by its mean $\mu = m(x) = \mathrm{E}[f(x)]$ and its covariance or kernel function

$k(x, x') = \mathrm{E}[(f(x) - m(x))(f(x') - m(x'))]$. As such, GP describes a distribution over possible Gaussian density functions. The chosen kernel k (e.g. periodic, linear, radial basis function) that describes the general shapes of the functions, defines a prior distribution of $f(X)$. This similarly equals selecting the degree of a polynomial function for regression. Placing the Gaussian prior over $f(X)$ yields a posterior joint distribution being used to determine the future process.

An *artificial neuronal network (ANN)* can pretend to be any type of regression model. The output of an ANN is based on the activation function between input and output layer. As an ANN is mainly used for classification, sigmoid function is used as a popular activation function, whereas when using ANN to solve a linear regression problem, the activation function is chosen as linear equation $y = w_0 + w_1 x_1 + \cdots w_n x_n$.

Support vector regression (SVR) is based on similar principles as support vector machine (SVM) for classification, identifying the optimal support vectors of a hyperplane that separates the data into their respective classes. Instead of separating classes, SVR fits a hyperplane describing the training data best. To solve the optimisation problem of finding the best hyperplane, the coefficient vector of the hyperplane is minimised – in contrast to ordinary least squares fitting where the squared error is minimised. Instead the squared error term is handled in the constraints allowing a certain error range ϵ, i.e., $\min \frac{1}{2}\|w\|^2$ s.t. $|y_i - w_i x_i| < \epsilon$.

4 Case Studies

We present two case studies from two different branches of industry. We introduce the case studies and follow with general experimental settings before evaluating our proposed approach in each case study.

4.1 Introduction and Data

In the first case study, we address degradation of mechanical bearings in manufacturing. In the second case study, we consider performance degradation of vessels in dry-bulk shipping. In the first case study we focus on one specific machine part, whilst in the second case study we address not only one specific part, but an entire system.

Bearing Degradation in Manufacturing The research project Collaborative Smart Contracting Platform for digital value-creation Networks (KOSMoS) provides a cross-company platform for a secure and semi-transparent exchange of production data[1]. The system establishes the optimal conditions for transparent documentation of the maintenance processes of a machine and thus supports, for example, the planning of service deployments. In addition, machine downtimes can be avoided by combining transparent documentation of maintenance history

[1] https://www.kosmos-bmbf.de/

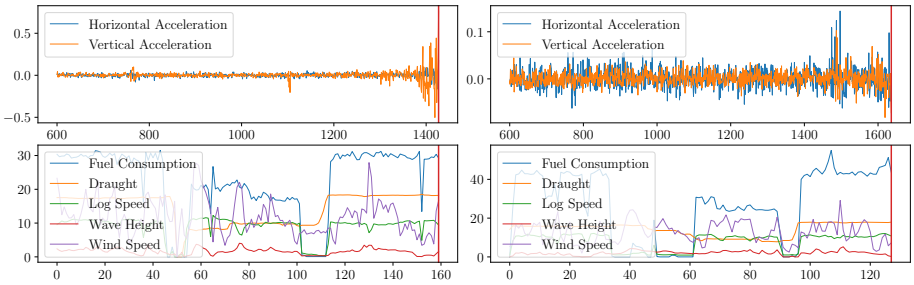

Fig. 3. Horizontal and vertical acceleration (vibration) of bearings b_{14} (top left) and b_{32} (top right). Fuel consumption, draught, speed, wave height and wind speed for two vessels (bottom left and right). The red line indicates end of useful lifetime.

and production data in predictive maintenance models [16]. A common challenge for the KOSMoS consortium partners from industry is the RUL prediction of mechanical *bearings*, a degrading machine part, which is installed in almost every machine, and thus has significant relevance for maintenance.

The dataset used for the case study is the well-known bearing dataset provided by FEMTO-ST institute within PRONOSTIA, an experimental platform dedicated to the testing and validation of bearing failure detection, diagnostic, and prognostic approaches [18]. The FEMTO bearing dataset contains run-to-failure tests of 17 bearings each with time series data of vibration acceleration along the horizontal and vertical dimension as well as temperature. Temperature is not present in every run, thus, we exclude it in our experiments. Details can be found in [18]. The observed data are divided into a training and a test set with six and eleven bearings, respectively. Figure 3 visualises the horizontal and vertical vibration acceleration over time for two bearings in the training set. Degradation itself corresponds directly to increasing vibrations.

Vessel Performance Degradation in Dry-Bulk Shipping Seaborne transportation is considered to be the most energy-efficient type of transportation due the amount of cargo carried on one single *vessel*. Nonetheless, the CO_2 emission made up form shipping is substantial when considering the overall global emission. The amount of fuel burned for vessel propulsion stands in direct relation to the emission and is one major cost driver of the vessels operational costs. Thus, from an environmental and commercial perspective it is key to reduce the amount of fuel burned. An increase in the fuel consumption can be interpreted as a decrease of a vessel performance and thus a decrease of its RUL. One of the main reasons for increasing fuel consumption is hull fouling, requiring vessel owners to periodically perform hull cleaning and propeller polishing [1].

To determine the effect of hull fouling on the fuel consumption, the relation of other variables impacting consumption such as weather, speed and vessel load need to be considered. Figure 3 gives an intuition of the relation of some of the variables considered to determine performance degradation due to hull fouling.

Table 2. All combinations of preprocessing steps used in the case studies.

	Statistical Features	Entropy Features	PCA	Autoencoder	Statistical+ PCA	Statistical+ Autoencoder	CNN
Time Spectrum	A	B	C	D	E	F	–
Frequency Spectrum	G	H	I	J	K	L	–
Time-Frequency Spectrum	–	–	–	–	–	–	M

Fuel consumption (blue) decreases/increases with changing speed (green) and changing draught (orange) due to different load of the vessel. Further, to retain vessel speed resistance effects like wind and waves need to be overcome, in turn as well impacting consumption. Waves and wind might positively impact propulsion (and thus consumption) depending on their direction. Please note, that for simplicity we here do not present all variables used. For this case study, we mainly follow the suggestions made by Adland et al. [1] and would like to emphasise to read on for better understanding of the variables. For the sake of completeness, we just name all variables used: air temperature, mean draught, draught forward, draught aft, fuel consumption, log speed, trim, speed over ground, wave height, wave direction, water salinity, water temperature, wind speed, wind direction. Our dataset consists of sensor data of 15 vessels splitted into sets of 12 vessels for training and 3 vessels for testing. Data are ranging from beginning of 2016 to end of 2020 with a time interval of five minutes between each observation of the variables. The point in time of the hull cleaning and propeller polishing operation is used as target variable.

4.2 Experimental Settings

We perform experiments for each case study, whereas each experiment results from the combinations of the components introduced in Section 3. Note that technically not all combinations of components from the frequency analysis and feature extraction step are possible, thus, we denote them explicitly as follows. We choose $Z_{i,j}$ to be an experiment, where $Z \in \{A, \dots, M\}$ denotes a combination of preprocessing steps listed in Table 2, $i \in \{\text{true}, \text{false}\}$ denotes if the HS classifier is used, and $j \in \{\text{MLR}, \text{GPR}, \text{ANN}, \text{SVR}\}$ denotes the selected regression model for prediction. In total, we conduct 104 experiments.

All approaches are compared based on the overall performance accuracy of each individual approach. To determine performance accuracy, we use root mean square error (RMSE) and Pearson correlation coefficient (PCC) between the observed and the estimated process of degradation. RMSE is defined by $\text{RMSE}(x, y) = (\frac{1}{n}\sum_{i=1}^{T}(y_i - x_i)^2)^{1/2}$, where $x = (x_1, ..., x_T)$ and $y = (y_1, ..., y_T)$ are time series and T is the length of both time series. PCC measures the linear correlation of two time series x and y, and is defined by $\text{PCC}(x, y) = (\sum_{i=1}^{T} x_i y_i - n\bar{x}\bar{y})/((T-1)s_x s_y)$ where $\bar{x}$, $\bar{y}$ and s_x, s_y are the mean and the

sample standard deviation of each respective time series. PCC describes the similarity of the behaviour of two time series, i.e., PCC indicates whether a learned model is able to correctly identify the degradation pattern (in case, PCC is close to 1). Note, PCC should be considered together with RMSE.

For the purpose of reproducibility, we list the implementation details as follows. Outliers are removed based on Z-Score before data is normalised with Min-Max-Scaler by scikit-learn. In case of different parameters or results, we write $(x_{\text{case 1}}|x_{\text{case 2}})$. FFT and STFT are implemented with SciPy. For STFT, the Hann window function is used with a window length of (256|30) and an overlap of (128|15). Statistical and entropy features are provided by tsfresh. For the calculation of Shannon entropy we use the classical symbolisation of the time series by SAX from pyts. For the calculation of permutation entropy we use the ordinal symbolisation by tsfresh with delay $\tau = 10$ and order $d = 5$. PCA is implemented using scikit-learn with encoding size 25. The autoencoder, CNN and ANN are implemented using Keras. The autoencoder architecture for feature learning consists of two encoding layers of size 160 and 80, followed by the coding layer of size 25 and two decoding layers of size 80 and 160. The CNN architecture for feature learning consists of 2 convolutional layers of dimension 6×6, each followed by a pooling layer of dimension 2×2 and a batch normalisation before the flattening layer is used for feature representation. The ANN architecture for the regression task consists of two hidden layers and an output layer, each of them with 512 hidden units. The activation function is chosen as rectified linear unit, i.e., $\text{ReLu}(x) = \max(0, x)$. To avoid overfitting, the dropout rate is set to 0.5. The autoencoder, CNN, and ANN are trained using Adam optimizer with learning rate 0.001 and loss function as mean squared error. MLR, GPR and SVR are implemented by scikit-learn with default settings. For health stage classification only one of the available variables is used, namely horizontal vibration for bearing and log speed for the vessel dataset. Observations in each dataset are recorded until end of useful lifetime. Thus, the difference between the observation time and the end of the recording denotes its RUL (see red line in Figure 3). RUL for the bearing dataset is in seconds, whilst RUL for the vessel dataset is in days.

4.3 Results

Each experiment is trained on a training dataset so that the RUL of an unseen sequence from the test dataset can be predicted before the results are then evaluated using RMSE and PCC. The experimental code and results can be found on GitHub[2]. Figure 4 shows violin plots for each experiment. We remind again that our aim is not to achieve the best prediction accuracy, but to evaluate the influence of each step in the prediction process. To answer the first research question, whether HS classification can improve the accuracy of the prediction, we compare RMSE and PCC of experiments $\{A, \ldots, M\}_{\text{true},j}$ vs. $\{A, \ldots, M\}_{\text{false},j}$ for every regression model $j = \{\text{MLR}, \text{GPR}, \text{ANN}, \text{SVR}\}$. Experiments show that RMSE

[2] https://github.com/inovex/RCIS2021-degradation-estimation-bearing-vessels

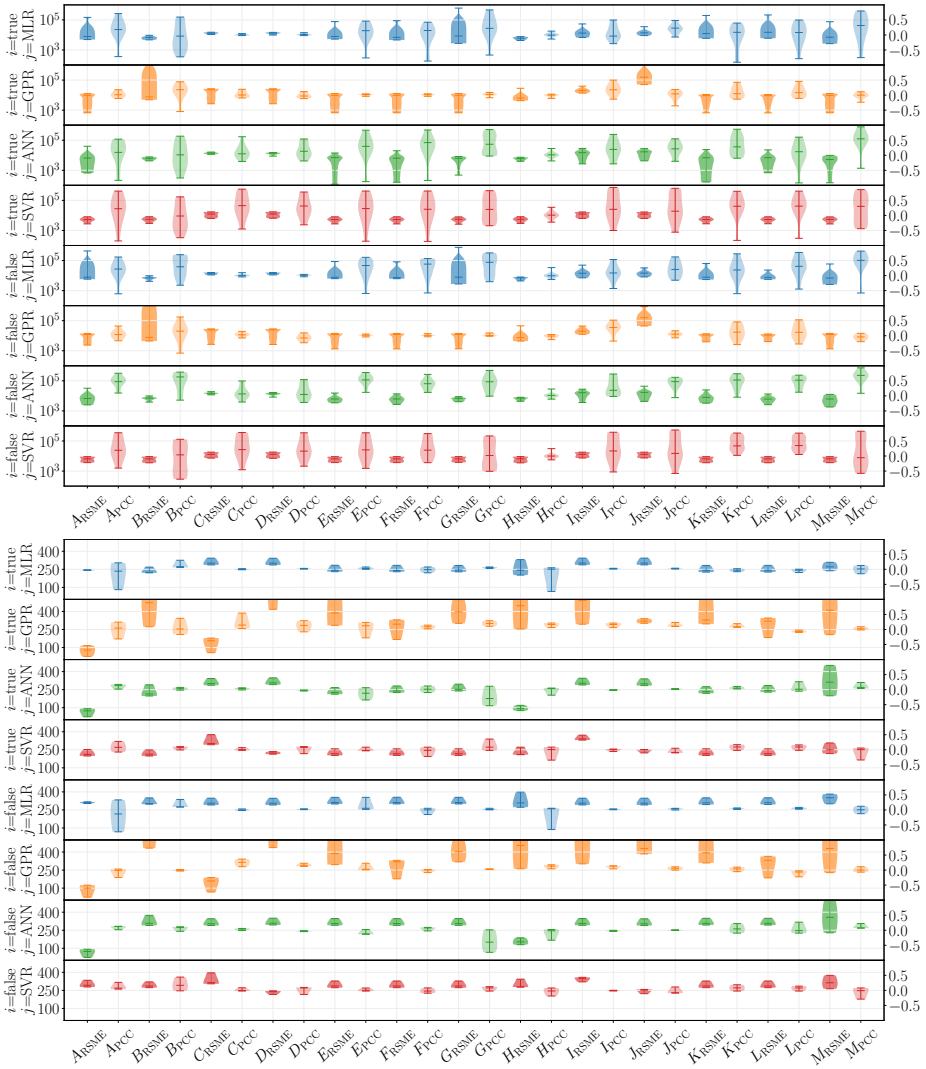

Fig. 4. Violin plots for RMSE (dark) and PCC (light) for bearing (top) and vessel (bottom) data for each experiment. Note, bearing results are log-scaled for readability.

decreases if the HS classifier is applied in (59|100)%, (78|97)%, (64|92)%, and (74|92)% of the predictions, respectively. Thus, in general, an improvement is observed. This does not imply that the total RMSE over all bearings or vessels must also decrease. Indeed, for *bearing* data it even increases for $\{K, L, M\}_{\text{true,MLR}}$ and $\{B, J\}_{\text{true,GPR}}$, which can be taken from Figure 4 (top, blue and orange). In case of *vessel* data, it increases for $\{A\}_{\text{true,ANN}}$ and $\{I\}_{\text{true,SVR}}$, which can be taken from Figure 4 (bottom, green and red). Compared to RMSE, PCC in-

creases in (41|59)%, (63|28)%, (34|41)%, and (69|38)% of the predictions, which indicates an overall improvement. Nevertheless, there is a deterioration of the average PCC in both case studies when using MLR. Hence, it should be checked individually whether there is an improvement in the functional relationship.

To answer the second research question, whether frequency analysis can provide additional information, we compare RMSE and PCC of experiments $A_{i,j}$ vs. $G_{i,j}$, $B_{i,j}$ vs. $H_{i,j}$, $C_{i,j}$ vs. $I_{i,j}$, $D_{i,j}$ vs. $J_{i,j}$, $E_{i,j}$ vs. $K_{i,j}$ and $F_{i,j}$ vs. $L_{i,j}$ for all i, j. The average RMSE and average PCC shows that only in the case of $B_{i,j}$ vs. $H_{i,j}$ an improvement is achieved, i.e., a reduction of the RMSE and an increase of the PCC. More specifically, we find that the feature calculation on the frequency spectra leads to a reduction of RMSE only in $(42|21)\%, (50|58)\%, (47|8)\%, (38|50)\%, (52|50)\%$ and $(55|42)\%$ of the predictions, which is close to random guessing. It is further to point out that $D_{\text{true,GPR}}$ and $E_{\text{true,MLR}}$ lead to an increase in RMSE in 100% of the predictions for bearing data, while in the case of vessel data they decrease in 100% of predictions. Therefore, we do not recommend blind use of frequency analysis, but rather use it wisely. Note, that we did not investigate whether combining features on the raw time spectrum in combination with features on the frequency spectrum gives better results. We leave this for future work in the context of feature selection.

To answer the third research question, which feature set is most appropriate, we compare RMSE and PCC of experiments $\{A, G\}_{i,j}$ vs. $\{B, H\}_{i,j}$ vs. $\{C, I\}_{i,j}$ vs. $\{D, J\}_{i,j}$ vs. $\{E, K\}_{i,j}$ vs. $\{F, L\}_{i,j}$ vs. $\mathrm{M}_{i,j}$ for $i = \{\text{true, false}\}$ and every j. For *bearing* data, the average RMSE per feature extraction method across all 8 experiments (with and without HS classification and 4 regression methods) are 20.299, 93.429, 15.797, 50.555, 10.952, 8.449, and 15.986, respectively, suggesting that CNN as particularly effective or entropy feature particularly ineffective. However, when considering the effectiveness of the features in the context of different regression models, experiments $\{A, G\}_{i,j}$, $\{E, K\}_{i,j}$, $\{F, L\}_{i,j}$, and $\mathrm{M}_{i,j}$ perform worst with MLR, and $\{B, H\}_{i,j}$, $\{C, I\}_{i,j}$, and $\{D, J\}_{i,j}$ perform worst with GPR. Disregarding these two regression methods, average RMSEs are 6.962, 6.265, 13.603, 13.513, 7.082, 6.721, and 6.238. Learned features perform on average more than twice as bad as engineered features. Feature Learning on engineered features, such as performing PCA or autoencoder on statistical features, is more efficient. In general, there is no free lunch, i.e., not every feature set is suitable for every regression model [22]. Across all methods, CNN performs best, followed by entropy features, which only fail in the context of GPR. Comparing the model complexities of the two feature extraction methods, it is even more remarkable that the relatively simple entropy features perform so well. For further evaluation, a time and space comparison is necessary, which we leave for future work. For *vessel* data, the average RMSEs per feature extraction method across all 8 experiments is 240, 323, 324, 430, 338, 277, and 335, suggesting that statistical features as particularly effective or autoencoder particularly ineffective. In contrast to bearing data, no outliers are evident across the feature extraction method, except for $\{B, D, J\}_{i,\text{GPR}}$, which is related to the regression model.

To answer the fourth research question, which regression method yields the highest accuracy, we compare RMSE and PCC of experiments $\{A, \ldots, M\}_{i,\mathrm{MLR}}$ vs. $\{A, \ldots, M\}_{i,\mathrm{GPR}}$ vs. $\{A, \ldots, M\}_{i,\mathrm{ANN}}$ vs. $\{A, \ldots, M\}_{i,\mathrm{SVR}}$ for all i. Regarding all experiments, the average RMSEs for each different regression model $j = \{\mathrm{MLR}, \mathrm{GPR}, \mathrm{ANN}, \mathrm{SVR}\}$ are $(22.677|295), (87.091|459)$, $(9.578|267)$, and $(8.041|272)$, respectively, with a standard deviation of $(19.166|33), (202.102|304)$, $(3.663|83)$, and $(3.042|48)$, respectively. In case of *bearing* data, if the two worst preprocessing steps for each regression model are removed from the analysis, i.e., by omitting $\{A, G\}_{i,\mathrm{MLR}}, \{B, J\}_{i,\mathrm{GPR}}, \{I, J\}_{i,\mathrm{ANN}}$, and $\{C, D\}_{i,\mathrm{SVR}}$ for all i, the average RMSEs can be reduced by 29%, 86%, 11%, and 10%, respectively. As a result, GPR has a higher average RMSE than MLR. Also in the case of *vessel* data, GPR has some remarkably poor predictions, in particular on learned features by the autoencoder. The average PCCs are $(0.13|0.01)$, $(0.05|0.07)$, $(0.28|0.01)$ and $(0.21|0.03)$, respectively, which is not close to 1 but still implies a positive relationship. In the case of the vessel data, there is more or less no functional relationship identifiable, which should definitely be improved. GPR in particular turns out to be unsuitable in both cases at first glance, which must be examined with regard to the outlier predictions. All in all, ANN and SVR prove to be particularly stable, which, together with the results of the third research question, indicates good ability to generalise.

5 Open Challenges, Limits and Recommendations

Since with this paper we provide recommendations for composing several methods and not a deployment-ready out-of-the-box framework, open challenges exist. There are still numerous other methods for HS classification, frequency analysis, feature extraction and regression. We have limited ourselves here to the most popular ones. As the focus of this work was not to achieve the best possible performance, but to investigate the relation of different components, the application of regularisation, feature selection methods, a corresponding hyperparameter tuning, as well as the optimisation of network architectures are left for future work. Learning non-linear relationships, as by locally linear embeddings, isometric mappings or kernel PCA can also further improve the results.

We conclude this paper with recommendations for composing data-driven prediction processes for degradation estimation based on the conducted experiments. Limits in the application depend on the individual use case that is to be implemented. Help can be found on GitHub[3]. Note that finding suitable degrading data directly related to the RUL of a machine part or complex system is not trivial. It requires initial analyses of the data and its correlations. The functional relationship have to be investigated or, if necessary, transformed by appropriate preprocessing such as creation of indicators. Along the research questions we recommend as follows.

1. *HS classifier*: We advise integrating a HS classifier within the degradation estimation process, as in the vast majority of cases both RMSE and PCC

[3] https://github.com/inovex/RCIS2021-degradation-estimation-bearing-vessels

are improved. Note that there are other HS classifiers that may be more appropriate for your individual problem.
2. *Frequency analysis*: We do not recommend predicting the degradation solely by features calculated on frequency spectra. This does not mean that such features cannot add value in combinations with others.
3. *Feature set*: While CNN and entropy features are most suited for bearing data, classical statistical features are for vessel data. For getting started, we recommend using feature engineering before putting a lot of effort into feature learning and tuning its hyperparameters. The feature extraction method can be easily replaced in the process later. A good prediction depends on both, the choice of features, as well as the choice of a model.
4. *Regression model*: GPR may be used with caution and only be applied to appropriate data. Furthermore, we recommend more complex models than MLR. Not surprisingly, ANN and SVR perform best, with ANN being able to better represent the functional relationship. SVR is known for good generalisation ability, which is also shown here.

Acknowledgement

Parts of the content of this paper are taken from the research project KOSMoS. This research and development project is funded by the Federal Ministry of Education and Research (BMBF) in the programme "Innovations for the production, services and work of tomorrow" (funding code 02P17D026) and is supervised by the Projektträger Karlsruhe (PTKA). We also thank Oldendorff Carriers GmbH & Co. KG., Lübeck, Germany for providing data for the case study. The responsibility for the content of this publication is with the authors.

References

1. Adland, R., Cariou, P., Jia, H., Wolff, F.C.: The energy efficiency effects of periodic ship hull cleaning. Journal of Cleaner Production **178** (2018)
2. Ahmad, W., Khan, S.A., Islam, M.M.M., Kim, J.M.: A reliable technique for remaining useful life estimation of rolling element bearings using dynamic regression models. Reliability Engineering & System Safety **184**, 67–76 (2019)
3. Bandt, C., Pompe, B.: Permutation Entropy: A Natural Complexity Measure for Time Series. Physical Review Letters **88**(17) (2002)
4. Benkedjouh, T., Medjaher, K., Zerhouni, N., Rechak, S.: Remaining useful life estimation based on nonlinear feature reduction and support vector regression. Engineering Applications of Artificial Intelligence **26**(7), 1751–1760 (2013)
5. Boskoski, P., Gasperin, M., Petelin, D., Juricic, D.: Bearing fault prognostics using Rényi entropy based features and Gaussian process models. Mechanical Systems and Signal Processing **52-53** (2015)
6. Du, S., Lv, J., Xi, L.: Degradation process prediction for rotational machinery based on hybrid intelligent model. Robotics and Computer-Integrated Manufacturing **28**(2), 190–207 (2012)
7. Goebel, K., Saha, B., Saxena, A.: A comparison of three data-driven techniques for prognostics. 62nd Meeting of the Society For MFPT (2008)

8. Hong, S., Zhou, Z., Zio, E., Wang, W.: An adaptive method for health trend prediction of rotating bearings. Digital Signal Processing **35** (2014)
9. Jia, F., Lei, Y., Lin, J., Zhou, X., Lu, N.: Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. Mechanical Systems and Signal Processing **72-73**, 303–315 (2016)
10. Kim, H.E., Tan, A.C., Mathew, J., Kim, E.Y.H., Choi, B.K.: Machine prognostics based on health state estimation using SVM. In: Asset condition, information systems and decision models (2012)
11. Kim, J.H.C., Nam H., D.A.: Remaining useful life prediction of rolling element bearings using degradation feature based on amplitude decrease at specific frequencies. Structural Health Monitoring (2017)
12. Li, X., Zhang, W., Ding, Q.: Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. Reliability Engineering & System Safety **182**, 208–218 (2019)
13. Loukopoulos, P., Zolkiewski, G., Bennett, I., Sampath, S., Pilidis, P., Li, X., Mba, D.: Abrupt fault remaining useful life estimation using measurements from a reciprocating compressor valve failure. MSSP **121** (2019)
14. Mao, W., He, J., Tang, J., Li, Y.: Predicting remaining useful life of rolling bearings based on deep feature representation and long short-term memory neural network. Advances in Mechanical Engineering **10**(12) (2018)
15. Mao, W., He, J., Zuo, M.J.: Predicting Remaining Useful Life of Rolling Bearings Based on Deep Feature Representation and Transfer Learning. IEEE Transactions on Instrumentation and Measurement **69**(4), 1594–1608 (04 2020)
16. Mohr, M., Becker, C., Möller, R., Richter, M.: Towards collaborative predictive maintenance leveraging private cross-company data. In: INFORMATIK 2020. Gesellschaft für Informatik, Bonn (2021)
17. Mohr, M., Wilhelm, F., Hartwig, M., Möller, R., Keller, K.: New approaches in ordinal pattern representations for multivariate time series. In: Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference (2020)
18. Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-M., B., Zerhouni, N., Varnier, C.: Pronostia: An experimental platform for bearings accelerated degradation tests. In: Conference on Prognostics and Health Management. (2012)
19. Ozkat, E.: The comparison of machine learning algorithms in estimation of remaining useful lifetime. In: Proceedings of 9th International BTKS (2019)
20. Pan, Z., Meng, Z., Chen, Z., Gao, W., Shi, Y.: A two-stage method based on extreme learning machine for predicting the remaining useful life of rolling-element bearings. Mechanical Systems and Signal Processing **144**, 106899 (2020)
21. Wang, F., Liu, X., Deng, G., Yu, X., Li, H., Han, Q.: Remaining Life Prediction Method for Rolling Bearing Based on the Long Short-Term Memory Network. Neural Processing Letters **50**(3) (2019)
22. Wolpert, D., Macready, W.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation **1**(1) (1997)
23. Wu, J., Wu, C., Cao, S., Or, S.W., Deng, C., Shao, X.: Degradation Data-Driven Time-To-Failure Prognostics Approach for Rolling Element Bearings in Electrical Machines. IEEE Transactions on Industrial Electronics **66**(1) (2019)
24. Xue, X., Li, C., Cao, S., Sun, J., Liu, L.: Fault Diagnosis of Rolling Element Bearings with a Two-Step Scheme Based on Permutation Entropy and Random Forests. Entropy **21**(1) (2019)
25. Zhang, B., Zhang, L., Xu, J.: Degradation Feature Selection for Remaining Useful Life Prediction of Rolling Element Bearings. Quality and Reliability Engineering International **32**(2) (2016)