Timeseries Feature Extraction for Dataset Creation in Prognostic Health Management: A Case Study in Steel Manufacturing

Thanos Kontogiannis[1], Wanda Melfo[2], Nick Eleftheroglou[3], and Dimitrios Zarouchas[4]

[1,3] *Intelligent and Sustainable Prognostics Group, Aerospace Structures and Materials Department, Faculty of Aerospace Engineering, TU Delft, Delft, 2629 HS, Netherlands*
a.kontogiannis@tudelft.nl
n.eleftheroglou@tudelft.nl

[1,3,4] *Center of Excellence in AI for Structures, Aerospace Structures and Materials Department, Faculty of Aerospace Engineering, TU Delft, Delft, 2629 HS, Netherlands*
d.zarouchas@tudelft.nl

[2] *Research and Development, Tata Steel Europe, IJmuiden, 1970 CA, Netherlands*
wanda.melfo@tatasteeleurope.com

ABSTRACT

This study focuses on a critical aspect of implementing prognostics and health management (PHM) for assets: the creation of a descriptive dataset. In real-world applications, dealing with sparse and unlabelled big data is common, particularly in industries like production lines where complex subprocesses are monitored by multiple sensors. Moreover, selective application of quality control means that much of the data lacks information about end properties, making datasets provided by manufacturers unsuitable for PHM frameworks. This work aims to bridge the gap between raw production data and PHM frameworks, focusing on steel manufacturing management. In the context of steel manufacturing, compromised surface quality, characterized by thicker oxide layers chipping during milling, has been observed. We propose inferring compromised coils by analyzing temperature profiles directly before the coiling station to address this. Deviations from the goal temperature profile can indicate compromised surface quality, eliminating the need for tedious oxide layer thickness measurements, which are not feasible for continuous hot strip milling processes. The available dataset comprised multiple years of production, with no direct indication of the surface quality. Exploratory clustering analysis was the first step in the lack of labels. Even though indicative of the underlying pattern of the healthy/damaged coils distinction, three shortcomings were identified. Clustering was solely based on the similarity between the temperature profiles of the coils, so no domain knowledge was included regarding the goal temperature profile. Additionally, since different steel grades have different goal profiles, the model needs to be specifically trained for each grade. Also, a soft classification between healthy and damaged can provide more detailed information about the surface quality. Coils with low-confidence classifications can be identified and treated accordingly, thereby improving PHM framework performance by providing a dataset with only high-confidence samples. To tackle these issues, an expert-knowledge-based normalization technique and feature engineering, paired with synthetic labelling, contributed to the creation of a soft neural network classifier. This study presents the reality of handling real-world data for PHM applications and highlights the need for careful and informed feature extraction. This ensures the seamless integration of PHM frameworks into real-world systems, ultimately enhancing production yield by improving end-product quality.

1. INTRODUCTION

The 4^{th} industrial revolution led to a skyrocketing increase in the available data in production and manufacturing lines. Sensors were developed and installed throughout the processes, and computer-operated regulating devices were retrofitted to production equipment. This not only meant that the manufacturing process could be guided by preset rules that were constantly tailored to the real measurements of the system but also that an enormous amount of data became available. Manufacturers, suspecting these data's value, made sure to gather and store them in databases. However, the vast majority of the available data are unstructured and unlabelled, leading to their under-utilization.

The unstructured nature of big data gathered from multiple sources during production is one of the main challenges of applying prognostic health management (PHM) frameworks (Zio, 2022). Data quality greatly affects performance, especially for fault detection (FD), which is usually the first task during a PHM framework. The need to identify anomalies and deviations from the normal operating condition of the asset arises, but data are usually high-dimensional with non-smooth distribution densities. This makes their reconstruction and, in turn, the distinction between healthy and abnormal, challenging.

In order to tackle the high-dimensionality of the data, the challenge of informative feature extraction (FE) arises (Jardine, Lin, & Banjevic, 2006), in the hope of projecting the data into manifolds of lower dimensionalities, where the underlying classes become distinguishable. Traditional pre-processing techniques include statistical feature extraction in the time domain (Caesarendra & Tjahjowidodo, 2017), fast fourier transform (Z. Wang, McConnell, Balog, & Johnson, 2014), discrete wavelet transform (Z. Wang et al., 2014), continuous wavelet transform (Kankar, Sharma, & Harsha, 2011), morphology operators (Gush et al., 2018) and principal component analysis (Choi, Lee, & Lee, 2005). Adding to that, in the latest years, due to the ever-increasing applications of machine learning (ML) and deep learning (DL) in computer science, numerous successful applications for FE for PHM frameworks have been demonstrated. Categorical adversarial autoencoders (Liu et al., 2018), stacked autoencoders (Y. Wang, Yang, et al., 2020), generative adversarial networks (Jiang, Hong, Zhou, He, & Cheng, 2019; Xia et al., 2022), deep convolutional networks (Wu & Zhao, 2018) and deep belief networks (Y. Wang, Pan, Yuan, Yang, & Gui, 2020), are examples of ML and DL frameworks to extract lower dimensionality representations of big data.

However, it becomes apparent that DL does not provide a universal solution to FE (Zhao et al., 2019). Great effort and resources are associated with designing and training a successful DL model, and usually, the impressively performing but complex architectures make the DL networks task- and domain-specific. In this work, presented with a big real-world dataset from steel manufacturing, a data science and fundamental approach for feature extraction is followed. The aim is to showcase that even highly complex datasets with high variability, can be handled with expert-driven analysis, proving the discriminating power of informative features. The following sections will describe the issue under consideration and the available dataset (Section 2), followed by an overview of the applied methods (Section 3), the findings (Section 4), and a concluding discussion (Section 5).

2. Problem statement and dataset description

Steel strips are being widely used for numerous applications across multiple domains, such as the automotive industry, the aerospace industry, chemical equipment and light manufacturing, all of which, among others, have increasing demands considering surface quality. However, surface defects can appear during manufacturing, which significantly diminishes the end surface quality of the manufactured steel strips. Known root causes of surface quality defects are material defects, process defects and corrosion defects (Z. Wang, Wang, & Chen, 2020). Material and process defects can be more easily mitigated by tailoring the material's composition and manufacturing process (i.e. rolling forces, timely inspection and replacement of rollers). Unfortunately, corrosion defects are, by nature, more challenging. The low stability of the typical three-layer oxide composition of steel (hematite Fe_2O_3, magnetite Fe_3O_4 and wustite $Fe_{1-y}O$) at the low coiling temperatures, the presence of other elements in low-carbon steel, the presence of inclusions, the continuous cooling conditions, the temperature gradient across the width of the strip, the absence or lack of oxygen in the centre regions, all affect the oxide evolution (Chen & Yuen, 2001; Deng et al., 2017). The extensive study of Min K. et. al. (Min, Kim, Kim, & Lee, 2012) revealed a correlation between the thickness of the oxide layer and the surface quality. This is attributed to the fact that a thicker oxide scale is more brittle and, thus, more prone to chip off.

As demonstrated by Min K. et al. (Min et al., 2012), measuring the oxide layer thickness during production is not feasible. Production must be halted, and the oxide layer formation must be frozen (i.e., by spraying molten glass on the surface). This process can quickly become costly and counterproductive for a real-world application. This fact, combined with the fact that practical scale differs from lab-grown (Deng et al., 2017), led our team to try to develop a way to infer it indirectly from production measurements.

The first step towards achieving this goal is creating a labelled dataset from historical data containing coils with deteriorated and pristine surface quality. We theorize that, by observing the steel strip's coiling temperature (CT) profile, major deviations from the goal temperature and, more importantly, rapid fluctuations, can indicate a chipped-off oxide layer. The reasoning behind this is that when the oxide layer chips off, some parts of exposed steel appear on the surface that have drastically different emissivity than the oxides, throwing off the pyrometer temperature measurements. Thus, the need to distinguish faulty cases from normal ones from sequential data arises.

The dataset in hand consists of the process parameters, the CT profiles and the material properties of the manufactured steel strips from the hot strip milling (HSM) process of Tata Steel Europe ©. Due to the great variability in the CT profiles as well as the goal temperatures, a single steel grade was chosen, considering its observed troublesome behaviour during milling (the details of which will not be disclosed due to confidentiality). After data cleaning, the remaining dataset consists of 3768 CT profiles, that will in turn, be used for the development of the classification algorithm.

3. METHODS

Given the dataset's unlabelled nature, an exploratory clustering analysis was the first step towards processing the dataset to discover the expected underlying pattern of healthy and damaged coils. Afterwards, a domain-specific normalization was introduced to the sequential data to assist towards creating a universal framework independent of the steel grade. This is of high importance since a great number of different steel grades are produced. Therefore, if the developed framework is grade-specific, it will need to be trained for each grade specifically, making it counter-productive. Two different FE techniques were realized and contrasted: a domain-agnostic one and an expert-knowledge-based one. Finally, synthetic labels were created to facilitate the training of a neural network (NN) soft classifier to discriminate the produced coils into healthy and damaged ones (meaning with chipped-off oxides and pristine surface quality, respectively).

3.1. K-means with Dynamic Time Wrapping

Clustering analysis is one of the first steps in processing unlabelled data, due to its ability to uncover underlying patterns and connections in the dataset without requiring any prior knowledge. A well-known and established clustering algorithm is the k-means algorithm (Lloyd, 1982). The k-means algorithm strives to partition the n available observations into k clusters, where each observation belongs to the cluster with the nearest mean, referred to as the centroids. The original algorithm works by minimizing the squared Euclidean distances between the centroids and the observations. An immediate issue can be observed when the algorithm is tasked with clustering sequential data. The Euclidean distance between two points A and B can be calculated by Eq. (1), with δ being the distance between the elements.

$$D(A,B) = \sqrt{\delta(a_1,b_1)^2 + \cdots + \delta(a_T,b_T)^2} \qquad (1)$$

If A and B are sequences with $A = \langle x,y,x,x\rangle$ and $B = \langle x,x,y,x\rangle$, their Euclidean distance according to Eq. (1), will be great, even though intuitively, they sequences are similar. This is attributed to the inability of the Euclidean distance to capture similarities that are shifted in time. For that reason, the dynamic time wrapping (DTW) metric is introduced. Its main attribute is that it can capture similarities between sequences independently of the velocity (Sakoe, 1971). The way to achieve this is by aligning the coordinates inside the sequences by minimizing Eq. (2), where A_i is the subsequence $\langle a_1,\ldots,a_i\rangle$.

$$D(A,B) = \delta(a_i,b_i) + \min\left\{\begin{matrix} D(A_{i-1},B_{j-1}) \\ D(A_i,B_{j-1}) \\ D(A_{i-1},B_j)\end{matrix}\right\} \qquad (2)$$

Even though DTW can effectively find the optimal alignment between sequences and provide a single score for similarity, k-means requires the calculation of a cluster prototype (the centroid), which is the average of the assigned observations. Petitjean et al. (Petitjean, Ketterlin, & Gançarski, 2011), proposed the DTW barycenter averaging (DBA) algorithm, which iteratively calculates the barycenter of a set of sequences for the k-means algorithm. Later on, a differentiable function for computing the soft minimum of all of the alignment costs increases performance and reduces arithmetic complexity, referred to as soft-dtw algorithm (Cuturi & Blondel, 2018). For the aforementioned reasons and considering the large size of the dataset, the soft-dtw algorithm is chosen.

3.2. Domain-specific Normalization

One of the main issues with the given application for the distinction between good and bad coils is the great difference between the goal CT profiles for different steel grades. The difference lies not only in the temperature but also in the shape of the wanted goal CT profile. Some steel grades require a coffin-shaped CT profile where the head and the tail of the coil are hotter than the middle section. Adding to that, steel strips that belong to each coil are not manufactured equally. The manufacturer provides a range of properties for each grade, and thus, the final goal CT profile depends on the exact needs of the specific order. This inevitably leads to the inability to generalize any realized framework since it would need to be redesigned and explicitly trained for each available steel grade. The authors try to alleviate this dependency by normalizing the CT profiles with the goal CT. Let $traj_1 = \langle t_1,t_2,\ldots,t_F\rangle$ and its respective goal CT $goal_{ct} = \langle g_1,g_2,\ldots,g_F\rangle$. The normalized CT profile is calculated with the following:

$$traj_1{}^j_{norm} = \frac{t_j - g_j}{g_j}, j = [1,F] \qquad (3)$$

For the remaining of the analysis, the normalized trajectories $traj_{norm}$ will be used.

3.3. Feature Extraction

For the good/bad coil distinction, a NN classifier will be utilized (as explained in Sec. 3.5). NNs are generally unable to handle and interpret sequential data as inputs, excluding recurrent NNs (RNN) (Amari, 1972). RNNs come with their own set of limitations with lengthy sequential data, namely the high computational complexity, the vanishing gradient problem and the often-required tedious hyper-parameter tuning. To partially tackle said limitations, one can choose to split the sequential data into overlapping windows, but the choice of the length and overlap of the windows adds to the complexity of choosing optimal hyperparameters. For the aforementioned reasons, traditional fully connected layers (FC) will be used, and thus, the CT profiles need to be represented with fea-

tures. Two different techniques for FE are being contrasted: A domain-agnostic approach where a plethora of features is being extracted (statistical, temporal and spectral) and filtered automatically, and an expert-knowledge-based one.

3.3.1. Domain-Agnostic FE

Given sequential data, a domain-agnostic FE refers to the process where a variety of features are extracted without considering the nature of the data in the hope of capturing as many characteristics as possible. In this study, the features extracted were:

- Statistical: max, absolute max, min, kurtosis, standard deviation, variation, mean, median, min, quantile, sum of values, length, variance, variation coefficient, count of values above/below mean value, first location of min and max, length of longest strike above/below mean, root mean square, sum of reoccurring values,
- Autocorrelation values (Yentes et al., 2013) for $lag = (1, 2, \ldots, 10)$ and descriptive statistics on the aggregation function (mean, variance, median, standard deviation) over the autocorrelation,
- Approximate entropy (Yentes et al., 2013) with $(m = 2, r = 0.1), (m = 2, r = 0.3), (m = 2, r = 0.5), (m = 2, r = 0.7), (m = 2, r = 0.9)$ with m the length of the compared run of data and r the filtering level,
- Non-linearity measure with c3 statistics (Schreiber & Schmitz, 1997) with $lag = (1, 2, 3)$,
- Complexity-invariant distance (CID) with and without normalization (Batista, Keogh, Tataw, & De Souza, 2014),
- Coefficients $(0, 1, \ldots, 14)$ of continuous wavelet transform with Ricker wavelet for $widths = (2, 5, 10, 20)$ (Mallat, 1999),
- All the coefficients (real and imaginary part, angle and absolute) of the fast Fourier transformation (FFT),
- Statistics of the absolute FFT (mean, variance, skew and kurtosis),
- Binned entropy of the power spectral density with the Welch method (Welch, 1967) ,
- Friedrich polynomial coefficients (Friedrich et al., 2000) for order of 3,
- Value of the partial autocorrelation function (Box, Jenkins, Reinsel, & Ljung, 2015) for $lag = (1, 2, \ldots, 10)$,
- Permutation entropy (Bandt & Pompe, 2002) with $dimension = (3, 4, \ldots, 7)$,
- Sample entropy (Richman & Moorman, 2000),
- Time reversal asymmetry statistic (Fulcher & Jones, 2014) with $lag = (1, 2, 3)$.

(The values chosen for the parameters of the aforementioned features are the commonly used values since tuning their values would require domain knowledge, defeating the purpose of a domain-agnostic framework).
After all of the features are extracted, to limit the number of irrelevant features, the FRESH algorithm (Christ, Kempa-Liehr, & Feindt, 2016) is deployed. It first performs the Kolmogorov-Smirnov test (Massey Jr, 1951) independently for every feature and calculates the p-value. Then, the FRESH algorithm utilizes the Benjamini-Yekutieli (Benjamini & Yekutieli, 2001) procedure under correction for dependent hypotheses to decide which null hypothesis H_0 to reject. Only the features for which the H_0 is rejected are kept. Finally, a Pearson correlation analysis is performed to remove features that are correlated with a value greater than 0.6, as this would indicate that they are (weakly) linearly correlated. Correlated features will get overweighted during the training, thus creating biased models whose results and generalizability can be compromised.

3.3.2. Expert-knowledge-based FE

Contrary to the first FE method, where a plethora of well-known features for sequential data are automatically extracted and filtered, for the expert-knowledge-based FE, as the name would suggest, a closer look at the data is required. After examining a normalized sequence for both a known good and a known bad coil (Figure 1), it becomes evident that the discrepancy between the two different classes is apparent in the time domain. Thus, the features that will be extracted are going to be limited to the time domain, meaning that no transformations to the data will be performed. The prominent characteristics of coils with a compromised surface quality are that they overshoot the upper and/or lower bounds of the accepted temperature range, that they present drifts from the goal temperature and, more importantly, they present abrupt peaks of high amplitude.
Based on the above observations, we choose to extract the features presented on Table 1. On the left column, the name of the feature is presented, while on the right are the values of the parameters that are used for their calculation. It is worth noting that the values of 0.05 and -0.05 were chosen for the threshold of the $count\_above$, $count\_below$ and $number\_crossing\_m$ features since the acceptable temperature range for the chosen steel grade is $\pm 5\%$.
The $number\_high\_peaks$ feature was engineered by the authors for this specific use case. The appearance of high-amplitude peaks is deemed detrimental to the classification of the coils, so a new feature is introduced to identify the peaks that have a standard deviation larger than 2 and return their count. The pseudo-code for the implemented feature can be found in Appendix.

3.4. Synthetic Labelling

Classification tasks are, by nature, handled by supervised algorithms. Supervised algorithms depend on labelled data to learn the decision boundary of the multidimensional manifold upon which the data points lie. To that end, the Tata Steel experts provided a set of 14 sequences of the steel grade under

Table 1. Features extracted for expert-knowledge-based FE.

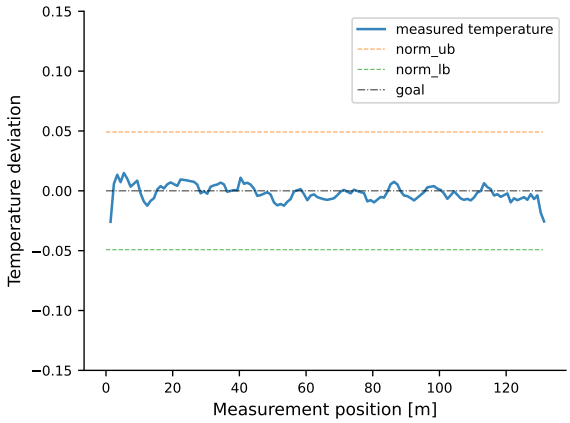
Feature	Parameters
abs_energy	-
absolute_maximum	-
absolute_sum_of_changes	-
cid_ce	normalize = False
count_above	t = 0.05
count_below	t = -0.05
skewness	-
longest_strike_above_mean	-
longest_strike_belowe_mean	-
maximum	-
mean	-
mean_abs_change	-
mean_change	-
minimum	-
number_crossing_m	m = 0.05 m = -0.05
standard_deviation	-
number_high_peaks*	n = 2 n = 5 n = 10

consideration that were identified to have low surface quality (one of which is shown in Figure 1b). Since the amount of labelled data is deemed inadequate to train a classification algorithm, the need to populate them arises. Upon inspection, and due to its use in the clustering analysis (Section 3.1), the DTW similarity metric is utilised. An ideal coil's CT profile would be identical to the goal CT profile. Leaning on that idea, the DTW similarity of each coil to the goal CT is calculated using Eq. (2). 76 coils with the highest score (indicating the **highest** dissimilarity to the goal CT) combined with the 14 expert-annotated ones comprise the bad coils labelled dataset. The 90 coils with the lowest DTW score form the good coils dataset.

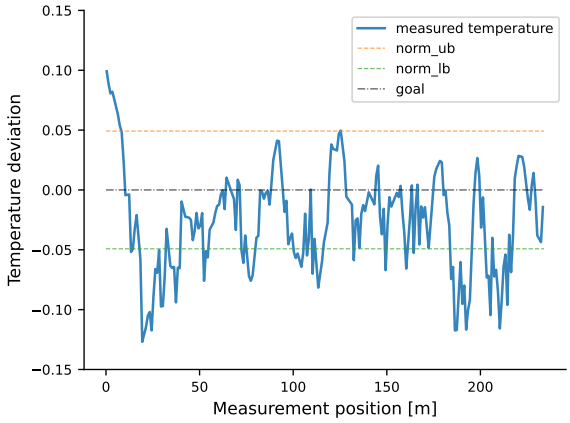
To guide the learning of the decision boundary, aside from providing examples of the extreme cases of both classes, we devised a labelled dataset of an extra 20 coils with intermediate DTW scores, half of which are used for training and half for testing. (this dataset will be referred to as manually annotated). This aims to not only provide information about the more ambiguous cases during training but also to provide a challenging test dataset that will assist in the evaluation of the performance of the classification algorithm. Figure 2 shows two of these coils. In conclusion, the final training dataset is constructed by performing an 80/20 % random split on the initial 180 coils and then adding half of the manually annotated dataset. The test dataset consists of the remaining data.

3.5. Neural Network classifier

A simple multilayer perceptron (MLP) is employed for the classification task. MLPs are fully connected feedforward NNs with non-linear activation functions. For the architecture of the model, typical design guidelines were followed. It consists of:



(a)



(b)

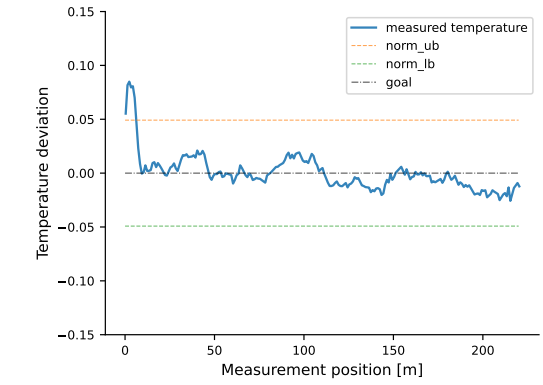
Figure 1. Normalized CT profile examples of a (a) good and a (b) bad coil

- **Input Layer:** where each feature is used as input for one input node,
- **Hidden Layer:** with $size = 64$, $relu$ activation function and to avoid overfitting, a dropout layer (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) with a dropout probability equal to 0.5,
- **Output Layer:** where, according to standard binary classification practice, it has $size = 2$ and a $softmax$ activation function, which will output the membership probability of each sample to each class.

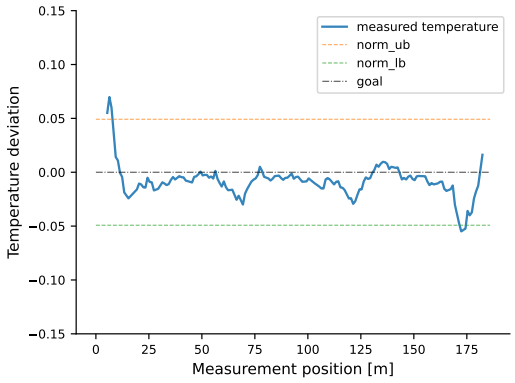
The simple and shallow architecture of the NN was chosen not only due to its decreased computational cost but also to avoid the tedious tuning and training of deep architectures.

4. Results

As previously discussed, the clustering analysis is performed on the raw data, while the classification is performed on the features extracted from the normalized sequences as described



(a)



(b)

Figure 2. Normalized CT profile examples of ambiguous coils. The spike in the beginning is observed on most coils, so it's not an indication of bad surface quality. Thus, (a) is labelled as good and (b) as bad (due to the peak at the end)

in Section 3.2. All of the code is written in Python, and the NN model was developed using Pytorch. Prior to the training of the NN, the training data are z-normalized. To avoid data leakage, the test data are z-normalized separately from the training data, utilizing the calculated scaling parameters of the training data. Models are trained for 200 epochs or until there is no improvement in the test accuracy. After the models have converged, they are tasked with classifying the entire dataset with all of the coils produced for the steel grade under consideration. The entire dataset follows the same FE procedure as the training set and is z-normalized with the pre-trained scaling parameters. Coils with a membership probability of less than 0.6 to either class are manually incorporated into the bad coils class to enhance our confidence in the models' predictions. Since the good/bad coil classification is the first step towards applying a PHM framework, we can tolerate false negatives, but we would like to avoid false positives. Since the good coils are of no interest to the analysis, a more inclusive bad coil class is preferred. First, the results of the clustering analysis on the raw data will be showcased, followed by the classification results with the introduced FE techniques.

4.1. K-means wth DTW

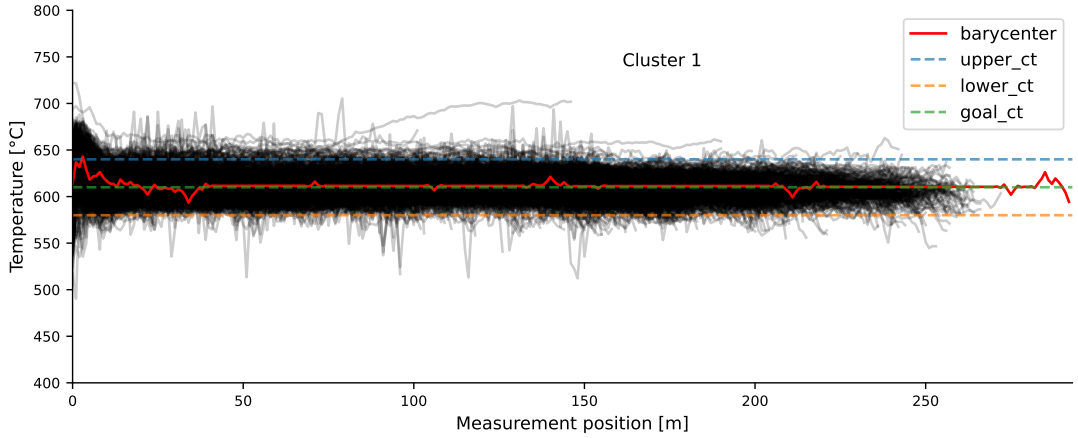
With the K-means algorithm, the number of cluster centres needs to be chosen a priori. To ensure the best fit, the elbow method is applied utilizing the silhouette coefficient (Rousseeuw, 1987). The results can be found in Table 2. As expected, the optimal number of clusters is two, confirming the prior assumption that the coils are split into good and bad ones. Figure 3 shows the results from the clustering and the calculated barycenters from the DBA algorithm. It becomes apparent that the majority of the coils in cluster 1 stay inside or close to the temperature boundaries, while bigger deviations are observed in cluster 2. This leads to the conclusion that the first cluster represents the good coils while the second cluster, the bad ones. However, the clustering is far from perfect since coils with high deviations and rapid fluctuations can be observed in the first (good) cluster. Given that the clustering analysis is the first exploratory step towards separating the data in hand, the results are satisfactory in that the expected underlying pattern of the data is actually observed. The high number of miss-clustered coils and the lack of soft-assignment capabilities means that it cannot be used as an end-to-end way to separate the data.

Table 2. Results of elbow method for DTW k-means.

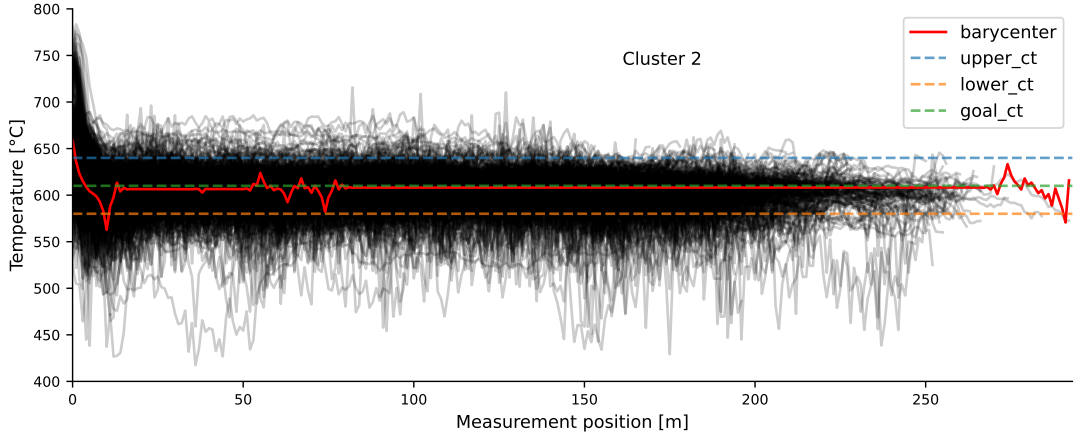
Clusters	Silhouette Score
2	**0.2556**
3	0.1938
4	0.1844
5	0.1947
6	0.1454

4.2. Classification with domain-agnostic features

After following the procedure of the domain-agnostic FE and filtering explained in Section 3.3.1, 111 features are left. The mean achieved accuracy of the NN is 0.8274 over the test data with 0.0180 standard deviation for 10 runs. The results can be seen in Figure 4. It can be observed that while the coils assigned in the bad class show a greater overall deviation from the goal CT, a lot of misclassified coils can be observed in the good class with highly fluctuating temperatures. This performance was to be expected, considering the rather low classification accuracy. To comprehend the low performance of the classification model, a principal component analysis (PCA) was performed on the extracted features with the goal of projecting the samples in a two-dimensional space. The calculated decision boundary is also drawn to enhance this visualisation's information. In order to achieve acceptable classification performance, the different classes need to present minimal overlap on the PCA space so that the classifier can find a way to separate them. This visualization can be seen in Figure 4c. A high overlap between the good and bad coils can be seen, meaning that no possible decision boundary can correctly separate the two classes, regardless of the choice of the model.

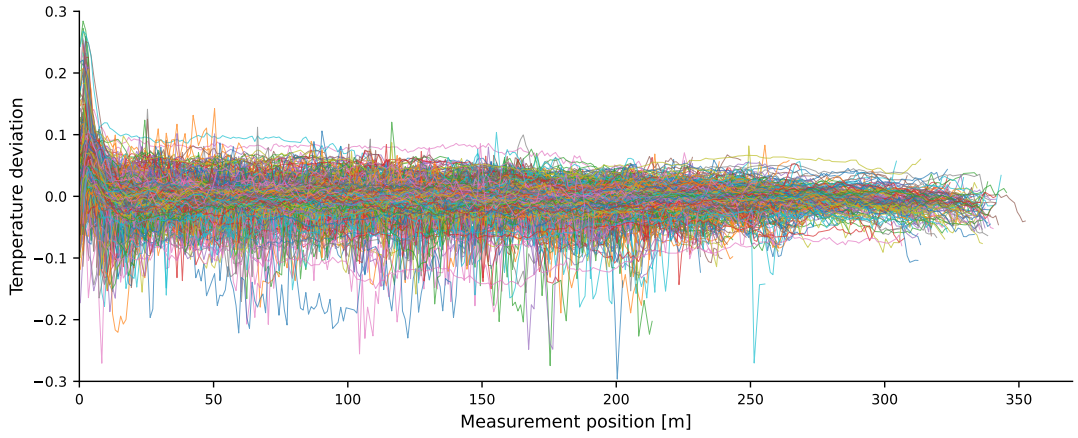


(a) Good coils cluster

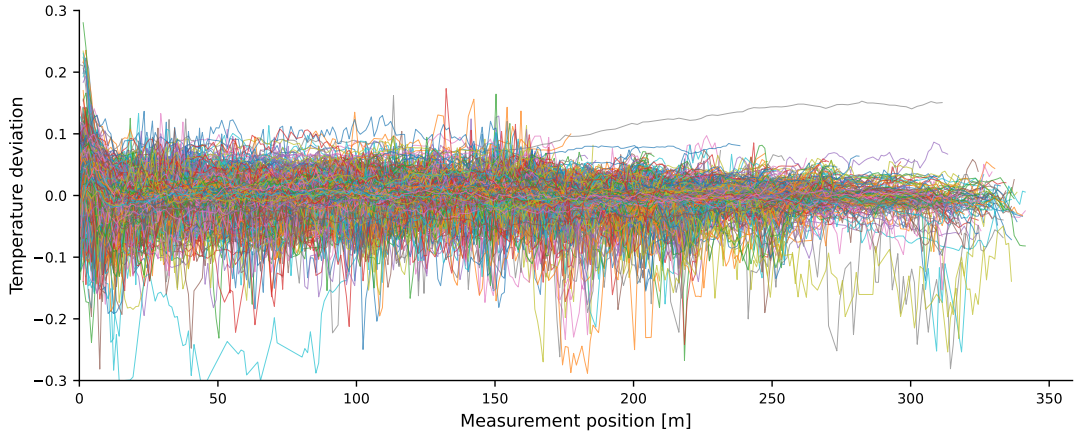


(b) Bad coils cluster

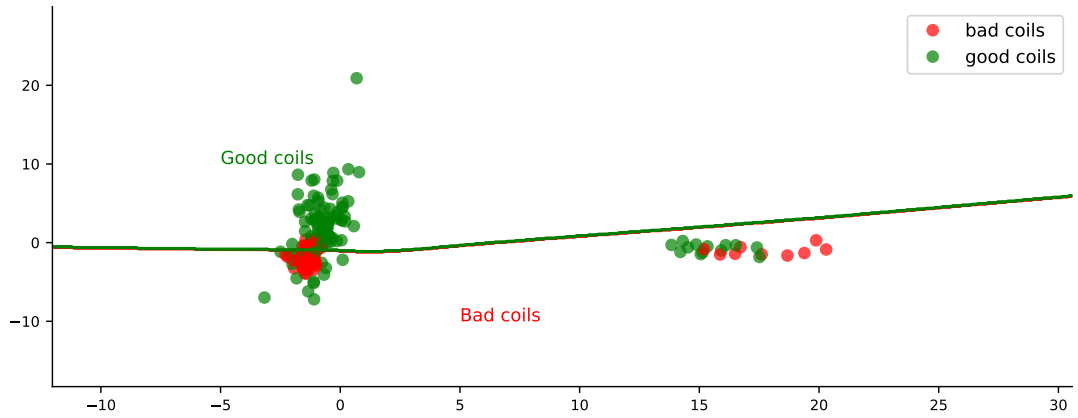
Figure 3. Clustering results with DTW K-means



(a) Good coils class

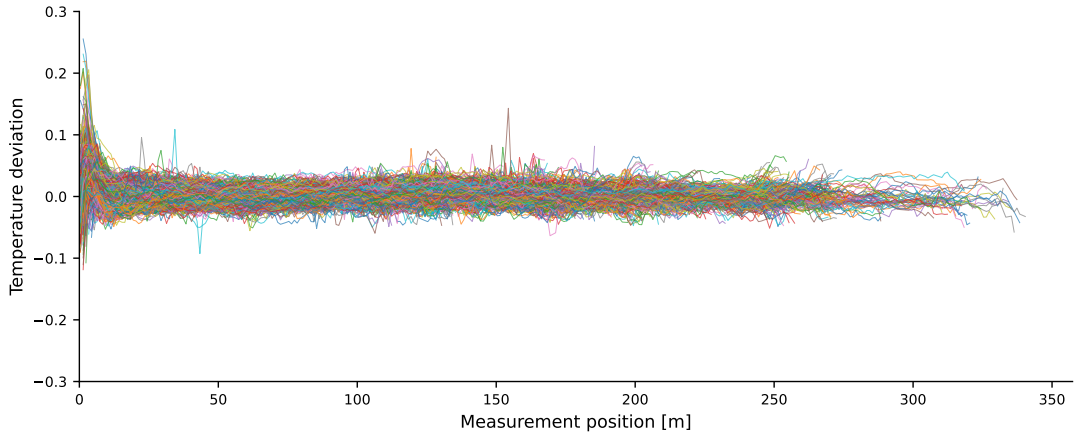


(b) Bad coils class

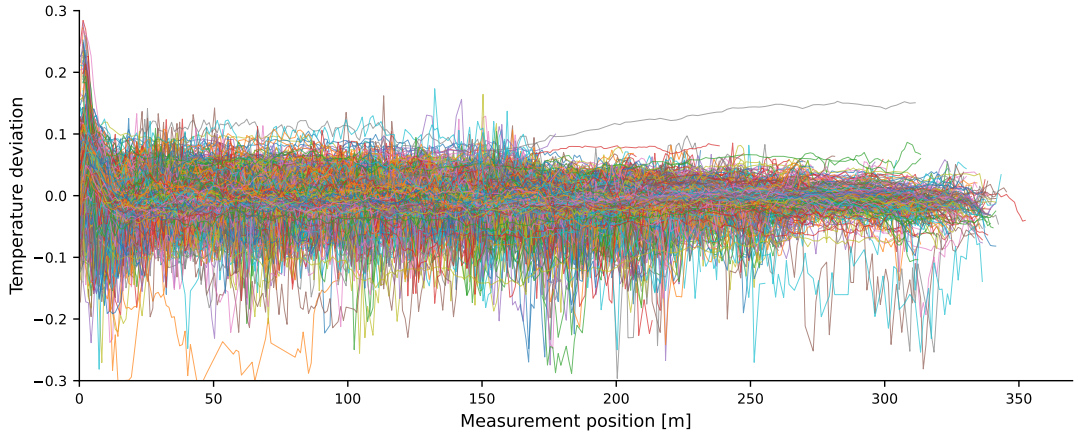


(c) 2-D PCA projection of the train and test samples

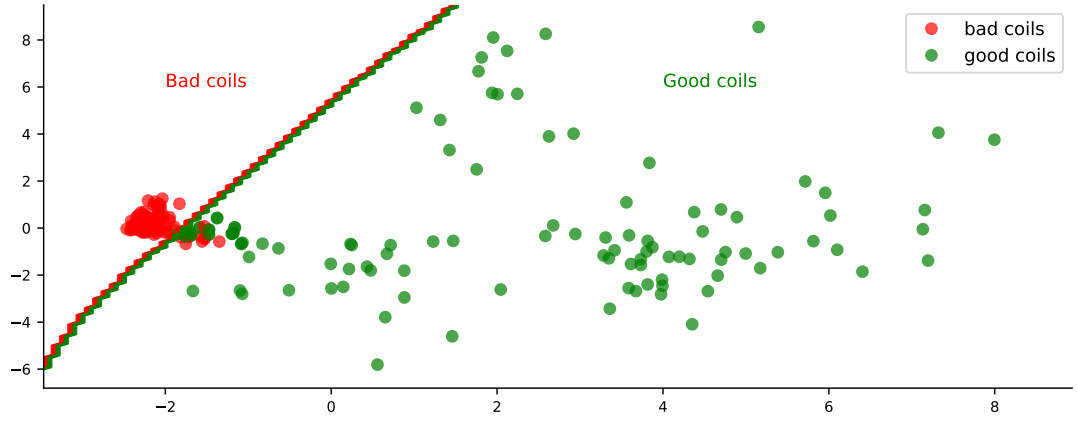
Figure 4. Classification results with domain-agnostic FE



(a) Good coils class



(b) Bad coils class



(c) 2-D PCA projection of the train and test samples

Figure 5. Classification results with expert-knowledge-based FE

4.3. Classification with expert-knowledge-based features

Following the FE method described in Section 3.3.2, 20 features are extracted. The same NN classifier architecture is used with the only change in the number of input nodes, which is altered to 20 to match the number of extracted features. The mean achieved accuracy is 0.9636 over the test data with 0.0075 standard deviation for 10 runs. The results can be seen in Figure 5. It becomes pretty apparent that the classification of the coils is superior to all of the previously presented methods. The healthy coils present minimal deviation from the goal temperature, with only a few coils that have a single abrupt temperature fluctuation in their CT, attesting to the high accuracy of the classifier. This means that there is a very limited number of false positive coils, which is highly important, as discussed at the beginning of the current section. The same visualization procedure is followed as before and presented in Figure 5c. It can be seen that there is a clear separation between the two classes and that the decision boundary lies optimally between them. This clear separation of the two classes explains the high performance of the rather simple classification model.

5. Discussion

The presented results pave the way for an important discussion when it comes to handling real-world complex and big data. After the clustering analysis was performed, the two expected different classes of coils could be identified, that is, the good and the bad class (referring to the CT profile and, in turn, the surface quality). However informative the clustering was in providing insight into the dataset, its performance was far from acceptable, with a lot of misclassified (or rather miss-clustered) coils. This is attributed to the fact that the K-means with DTW distance metric is clustering coils strictly by comparing their shape to each other. No information regarding the acceptable temperature range, the goal CT or what good and bad coils are, is included. Adding to that, the k-means algorithm does not provide a way to soft-assign clusters to data points. Naturally, the next step would be to increase the classification's performance will achieving soft-classification capabilities. The most obvious idea is to create a representation of the data to train a soft ML classifier. Due to the increasing popularity of DL FE methods, a researcher would most probably invest their time in developing complex and computationally heavy models. These models' task would be to try and learn on their own latent representations of the data that would effectively separate the different classes. With this study, we would like to emphasize that traditional FE can be as (if not more) effective for some datasets while reducing the complexity, the computational load, and the overall time invested in developing the FE method.

This is not to say that traditional FE can be applied universally, without effort. This is the main takeaway from comparing an automated traditional FE method that is domain-agnostic with features that are specifically picked or engineered for the application. For the domain-agnostic FE, a plethora of famous and commonly used features for sequential data were automatically extracted and filtered utilizing hypothesis tests and correlation analysis. However, the resulting features fail to capture the distinctive features of the data. This becomes evident by the high overlap of the two classes presented in Figure 4c, and is the culprit of the wrong classification of the data. Spending the effort of manually labelling a small fraction of coils and choosing the right features to represent the data, successfully separates them and achieves the required classification performance.

The next step for this framework is to verify that it works universally for multiple steel grades, with minor or even no modifications at all. After generating the healthy/damaged coils dataset, the process parameters that lead to the damaged state are intended to be identified. The end goal is to apply a PHM framework that will be able to predict quality deterioration and provide alternative parameter settings to mitigate the damage to the surface quality of the produced steel strips.

6. Conclusions

In this study, a real-world data set of manufactured steel strips raises the importance and effectiveness of traditional FE, but only if done appropriately, as described in Section 3.3.2 and paired with manually annotated samples. Automated FE techniques are deemed ineffective; thus, the extracted features must be chosen carefully. This is achieved by keeping in mind that they should capture the characteristics that associate them with their corresponding class. The authors are by no means undermining the importance of deep learning FE methods. Their increasing popularity mainly stems from their successful application in extracting latent representations of big data. They would instead highlight that for some datasets, the effort needed to develop them is unjustified; that is when a correctly defined traditional FE method can solve the task.

References

Amari, S.-I. (1972). Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on computers*, *100*(11), 1197–1206.

Bandt, C., & Pompe, B. (2002). Permutation entropy: a natural complexity measure for time series. *Physical review letters*, *88*(17), 174102.

Batista, G. E., Keogh, E. J., Tataw, O. M., & De Souza, V. M. (2014). Cid: an efficient complexity-invariant distance for time series. *Data Mining and Knowledge Discovery*, *28*, 634–669.

Benjamini, Y., & Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, 1165–1188.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*.

John Wiley & Sons.

Caesarendra, W., & Tjahjowidodo, T. (2017). A review of feature extraction methods in vibration-based condition monitoring and its application for degradation trend estimation of low-speed slew bearing. *Machines*, *5*(4).

Chen, R., & Yuen, W. (2001). Oxide-scale structures formed on commercial hot-rolled steel strip and their formation mechanisms. *Oxidation of metals*, *56*(1), 89–118.

Choi, S. W., Lee, C., & Lee, e. a. (2005). Fault detection and identification of nonlinear processes based on kernel pca. *Chemometrics and intelligent laboratory systems*, *75*(1), 55–67.

Christ, M., Kempa-Liehr, A. W., & Feindt, M. (2016). Distributed and parallel time series feature extraction for industrial big data applications. *arXiv preprint arXiv:1610.07717*.

Cuturi, M., & Blondel, M. (2018). *Soft-dtw: a differentiable loss function for time-series.*

Deng, G., Zhu, H., Tieu, A. K., Su, L., Reid, M., Zhang, L., ... others (2017). Theoretical and experimental investigation of thermal and oxidation behaviours of a high speed steel work roll during hot rolling. *International Journal of Mechanical Sciences*, *131*, 811–826.

Friedrich, R., Siegert, S., Peinke, J., Siefert, M., Lindemann, M., Raethjen, J., ... others (2000). Extracting model equations from experimental data. *Physics Letters A*, *271*(3), 217–222.

Fulcher, B. D., & Jones, N. S. (2014). Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, *26*(12), 3026–3037.

Gush, T., Bukhari, S. B. A., Haider, R., Admasie, S., Oh, Y.-S., Cho, G.-J., & Kim, C.-H. (2018). Fault detection and location in a microgrid using mathematical morphology and recursive least square methods. *International Journal of Electrical Power & Energy Systems*, *102*, 324–331.

Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, *20*(7), 1483–1510.

Jiang, W., Hong, Y., Zhou, B., He, X., & Cheng, C. (2019). A gan-based anomaly detection approach for imbalanced industrial time series. *IEEE Access*, *7*, 143608–143619.

Kankar, P. K., Sharma, S. C., & Harsha, S. P. (2011). Fault diagnosis of ball bearings using continuous wavelet transform. *Applied Soft Computing*, *11*(2), 2300–2312.

Liu, H., Zhou, J., Xu, Y., Zheng, Y., Peng, X., & Jiang, W. (2018). Unsupervised fault diagnosis of rolling bearings using a deep neural network based on generative adversarial networks. *Neurocomputing*, *315*, 412–424.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, *28*(2), 129–137.

Mallat, S. (1999). *A wavelet tour of signal processing*. Elsevier.

Massey Jr, F. J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, *46*(253), 68–78.

Min, K., Kim, K., Kim, S. K., & Lee, D.-J. (2012). Effects of oxide layers on surface defects during hot rolling processes. *Metals and Materials International*, *18*, 341–348.

Petitjean, F., Ketterlin, A., & Gançarski, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. *Pattern recognition*, *44*(3), 678–693.

Richman, J. S., & Moorman, J. R. (2000). Physiological time-series analysis using approximate entropy and sample entropy. *American journal of physiology-heart and circulatory physiology*, *278*(6), H2039–H2049.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65.

Sakoe, H. (1971). Dynamic-programming approach to continuous speech recognition. In *1971 proc. the international congress of acoustics, budapest.*

Schreiber, T., & Schmitz, A. (1997). Discrimination power of measures for nonlinearity in a time series. *Physical Review E*, *55*(5), 5443.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929–1958.

Wang, Y., Pan, Z., Yuan, X., Yang, C., & Gui, W. (2020). A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network. *ISA transactions*, *96*, 457–467.

Wang, Y., Yang, H., Yuan, X., Shardt, Y. A., Yang, C., & Gui, W. (2020). Deep learning for fault-relevant feature extraction and fault classification with stacked supervised auto-encoder. *Journal of Process Control*, *92*, 79–89.

Wang, Z., McConnell, S., Balog, R. S., & Johnson, J. (2014). Arc fault signal detection - fourier transformation vs. wavelet decomposition techniques using synthesized data. In *2014 ieee 40th photovoltaic specialist conference (pvsc)* (p. 3239-3244).

Wang, Z., Wang, J., & Chen, S. (2020). Fault location of strip steel surface quality defects on hot-rolling production line based on information fusion of historical cases and process data. *IEEE Access*, *8*, 171240-171251.

Welch, P. (1967). The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, *15*(2), 70–73.

Wu, H., & Zhao, J. (2018). Deep convolutional neural network model based chemical process fault diagnosis. *Comput-*

ers & chemical engineering, *115*, 185–197.

Xia, X., Pan, X., Li, N., He, X., Ma, L., Zhang, X., & Ding, N. (2022). Gan-based anomaly detection: A review. *Neurocomputing*, *493*, 497–535.

Yentes, J. M., Hunt, N., Schmid, K. K., Kaipust, J. P., McGrath, D., & Stergiou, N. (2013). The appropriate use of approximate entropy and sample entropy with short data sets. *Annals of biomedical engineering*, *41*, 349–365.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, *115*, 213–237.

Zio, E. (2022). Prognostics and health management (phm): Where are we and where do we (need to) go in theory and practice. *Reliability Engineering System Safety*, *218*, 108-119.

APPENDIX

Algorithm 1 Pseudocode of number_high_peaks feature

Inputs:
x (list): the input sequence
n (int): the support of the peak (a peak of support n is defined as a subsequence of x where a value occurs, which is bigger than its n neighbours to the left and to the right)
std_t (int): the number of standard deviations that the peak's value needs to surpass
Procedure:
$x\_reduced = x[n : -n]$
$res = None$
for $(c = 0; c < n + 1; c + +)$ **do**
  $result\_first = x\_reduced > numpy.roll(x, c)[n : -n]$
  if $res = None$ **then**
    $res = result\_first$
  else
    $res \mathrel{+}= result\_first$
  end if
  $res \mathrel{+}= x\_reduced > numpy.roll(x, c)[n : -n]$
end for
$idx\_peaks = np.where(res)[0] + n$
$h_peaks = 0$
for $idx : idx\_peaks$ **do**
  if $|x[idx]| > mean(x) + std\_t * std(x)$ **then**
    $h\_peaks \pm 1$
  end if
end for
Output:
h_peaks (int): the amount of peaks of support n with maximum value higher than std_t times the standard deviation of x