applied sciences

MDPI

Article

Model-Free Data Mining of Families of Rotating Machinery

Elizabeth Hofer and Martin v. Mohrenschildt *

Faculty of Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada; hofere1@mcmaster.ca
* Correspondence: mohrens@mcmaster.ca

Abstract: Machines designed to perform the same tasks using different technologies can be organized into families based on their similarities or differences. We are interested in identifying common properties and differences of such machines from raw sensor data for analysis and fault diagnostics. The usual first step is a feature extraction process that requires an understanding of the machine's harmonics, bearing frequencies, etc. In this paper, we present a model-free path from the raw sensor data to statistically meaningful feature vectors. This is accomplished by defining a transform independent of the operating frequency and performing statistical reductions to identify the components with the largest variances, resulting in a low dimensional statistically meaningful feature space. To obtain an insight into the family relationships we perform a clustering. As the data set has some labeled characteristics we define an entropy-based measure to evaluate a clustering using the a priori-known labels, resulting in a symmetric measurement uniquely defining the clustering goal. Applying this hierarchically we obtain the family tree. The methods are presented can be applied in general situations. As a case study we apply them to a real data set of vibrating screens.

Keywords: rotating machines; feature extraction; feature reduction; unsupervised learning; hierarchical clustering; entropy

check for updates

Citation: Hofer, E.; v. Mohrenschildt, M. Model-Free Data Mining of Families of Rotating Machinery. *Appl. Sci.* **2022**, *12*, 3178. https://doi.org/10.3390/app12063178

Academic Editors: Radu Godina and Pedro Espadinha da Cruz

Received: 18 January 2022
Accepted: 14 March 2022
Published: 21 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Recent advances in data mining and AI have given machine manufacturers valuable tools to aid in the prediction of faults; this, however, is contingent on the identification of meaningful features in the data. Within the context of rotating machines, any machine driven by a motor or shaft, obvious commonly used features include RPM, rotational phases, and G-force, Fourier transform. However, our research has shown that these features can only aid in uncovering information that was previously known, and that manufacturers are struggling to mine non-trivial information. A use-case for data mining that manufacturers are interested in would be to establish baseline operational conditions for a family of machines—a group of machines that are designed to accomplish the same task using different technologies. With an accurate collection of family baselines, we can detect abnormal behaviour and predict faults, allowing for the optimal direction of resources for investigation and repair (in industry known as preventive maintenance). Thus, the challenge we are presented with is extracting meaningful features and using them to determine the families.

Rotating machines are used in many applications across many industries. For example, the mining and aggregate industry uses vibrating screens to sort gravel by size by exciting the gravel at specific frequencies. The vibrational recordings of the machines, as captured by a series of accelerometers, contain a wealth of information about the machine. We can use this to determine a *machine family* relationship. Family members may vary in size or geometry, drive type, their bearings may contain a different number of roller elements, etc.

In this paper, we propose a model-free approach to data-mining for rotating machines, to extract statistically significant features from raw sensor data and then use a conditioned entropy-based clustering criterium designed to create meaningful *family* groupings.

Background

The mining of sensory data of machinery is studied extensively in the literature. Sensors such as accelerometers, microphones, transducers, or induction coils are used to obtain time-domain data of the machine operating periodically at a steady state. The data are then (1) processed, (2) their features are extracted, and (3) analysis/decisions are performed [1]. Possible features include time-domain features [2,3], but usually frequency domain features are used. Approaches typically use a Fourier transform for feature extraction such as in [4,5], but all these assume a known frequency fingerprint to be extracted. In order to specifically construct a feature-space that best suits the application some works define their own feature extraction process [3,6].

Model-free for us means that we do not have any knowledge of the harmonic structure of the machines or bearing information, we only assume a periodic movement at some operating frequency. We propose a feature space that is independent of the operating frequency of the machine without any assumptions on the harmonic structure and statistically determine the significant features.

Model-free is often associated with neural networks and deep learning [7]. Deep Learning (DL) approaches often operate on the time-domain data directly, combining steps (1–3), e.g., [8] for bearing fault diagnostics, and [9] which presents a general review of time-series classification techniques. DL methods are quite accurate [8] but due to the “black box” nature of ML and DL it does not provide engineers any insight as to which features of the data are particularly important to classifications.

We perform a model-free approach, identifying frequency patterns by performing statistical analysis in the frequency domain. More specifically, we introduce the notion of the *Harmonic Feature Space* (HFS), an operating frequency independent domain. In this Harmonic Feature Space we perform statistical reductions [10], including principal component analysis (PCA) [11], resulting in significant dimensionality reduction. We are driven to identify components that maximize the variance. We point out that feature vectors are computed from raw sensor data without any prior understanding of the harmonic structures, fractional harmonics, resonances, or other fault frequencies.

Support Vector Machines (SVM) are a go-to tool in classification and work very well for many applications [12] including fault diagnostics [13]. Multi-class SVM suffer from the problem of being sensitive to/requiring class-specific scaling [14]. Additionally, SVM are a supervised learning process, they do not directly provide new insights similarities between machines. From a classification standpoint, there is significant literature comparing neural network approaches to SVM, e.g., [13]. We take and agree that with a well-designed feature extraction SVM performs very well and often outperforms neural networks as a classification tool. While we demonstrate that the proposed harmonic feature space allows for high accuracy training/recall of SVM classifications, we focus on clustering methods.

The goal of clustering is to compute subsets of the data, the clusters, grouping data that is similar as defined by some measurement function, usually a Euclidean distance [15–17]. Several questions are studied, including algorithmic aspects, what is the “best” distance measure and what is the “best” number of clusters. We are particularly interested in the latter; given a distance measure how many clusters should we group the data into, and how many will represent the natural clusterings of the data? This is not a trivial question. As mentioned previously, we use a hierarchical approach to obtain insights into the machine’s family, but the “correct” family groupings are open to interpretation. Known methods of determining the optimal number of clusters include the Elbow method (which we use for the components later in this paper), information criterion approaches, and information-theoretic approaches.

We find our answer by exploring the conditioned or Bayesian entropy. Entropy-type measures for the heterogeneity of clusters are well known [18]. As we have labels for the data set we can derive clustering goals: recordings of the same machine should be in the same cluster, and recordings corresponding to different labels should be in different clusters. Note that we do necessarily want to have clusterings identical to the labels, it is acceptable, if not ideal, to have a certain class of machine split into more than one cluster as that gives us insight into the nature of the data. The conditions are hence derived based on prior information about the clustering we want. These goals are usually competing, so we define the notion of *symmetric conditioned entropy* that allows us to compute a unique number of clusters satisfying set criteria by minimizing that measurement. The recordings naturally, without any scaling/unsupervised learning, group into clusters. Repeatedly clustering the clusters again, hierarchically [19], results in the desired family tree.

We apply the presented methods to a real data set, accelerometer recordings of large vibrating screens. We have about 1800 recordings of 150 different machines with several different operating principles, different geometry, and varying sizes.

2. The Harmonic Feature Space

In contrast to model-based approaches we assume we have no information about the frequency patterns of the machines except that they operate at a fixed frequency. In order to obtain meaningful features of the data set, we need to construct a feature space that is independent of the fundamental (operating) frequency. Simply computing the Fourier transform of the recordings, as shown in Figure 1 does not communicate any relationships or patterns between the data. Instead we choose to transform the data such that the frequency bins are placed proportionally to the operating frequency f_{op} as demonstrated in Figure 2, clear patterns appear in the harmonics and fractional harmonics of the recording.

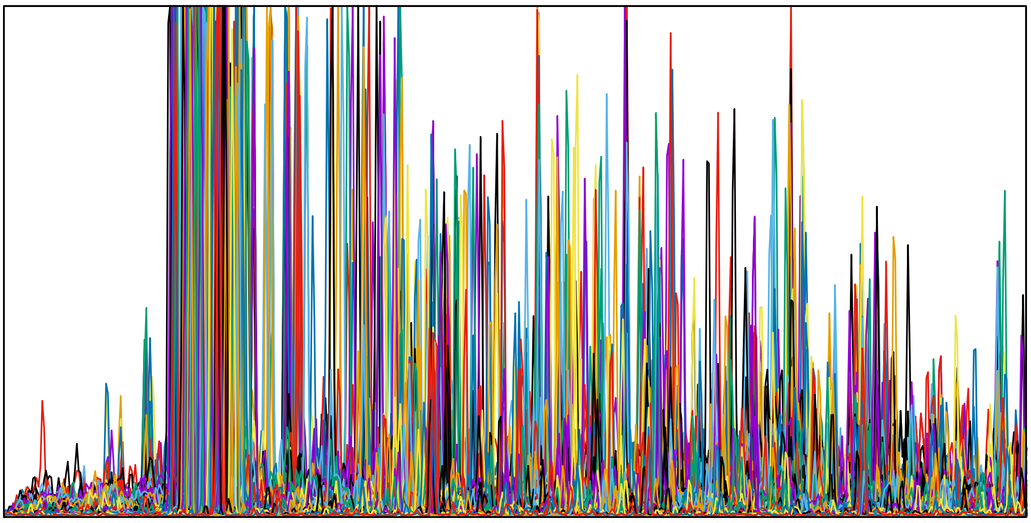

Figure 1. The Fourier transform bins of the *z*-axis of 1800 recordings all computed using the same FFT transform parameters.

To obtain the operating frequency we initially compute a Fourier transform and use a quadratic interpolation of the complex values around the maximal bin [20], other approaches are possible.

We align the patterns in the feature space by recomputing the DFT computing a specific N for each recording such that the first harmonic lies in some fixed bin d where d is the same across all recordings:

$$X_k = \sum_{n=0}^{N-1} x(n) e^{-i\omega_0 nk}, \, X_k \in \mathbb{C}$$

$$f_{op} \simeq f_s \frac{d}{N} \rightarrow N = round(\frac{f_s d}{f_o p}).$$

Hence, by construction, all second harmonics lie in bin $2d$ and so on. This creates a *relative machine fingerprint* of the harmonic resonances for each machine that can easily be compared with other machines, as illustrated in Figure 2. Figure 3 shows the mean of the same data after component-wise reduction as described in the next section.

We must recompute the DFT for each sample because aligning the harmonics is not a linear shift. This has the additional benefit of virtually eliminating spectral leakage since the new ω_0 is an integer multiple of the operating frequency.

We do not consider the magnitude of energy of the first harmonics to be relevant to the machine condition, rather only the energy of the harmonics in proportion to the first harmonic. Therefore, for each sample, we divide all bins by the complex value X_d of the first harmonic bin $X_k \leftarrow X_k X_d^{-1}$ to normalize all recordings. Since X_d is a complex number the normalization will also shift the phases of all the bin to be relative to the first harmonic.

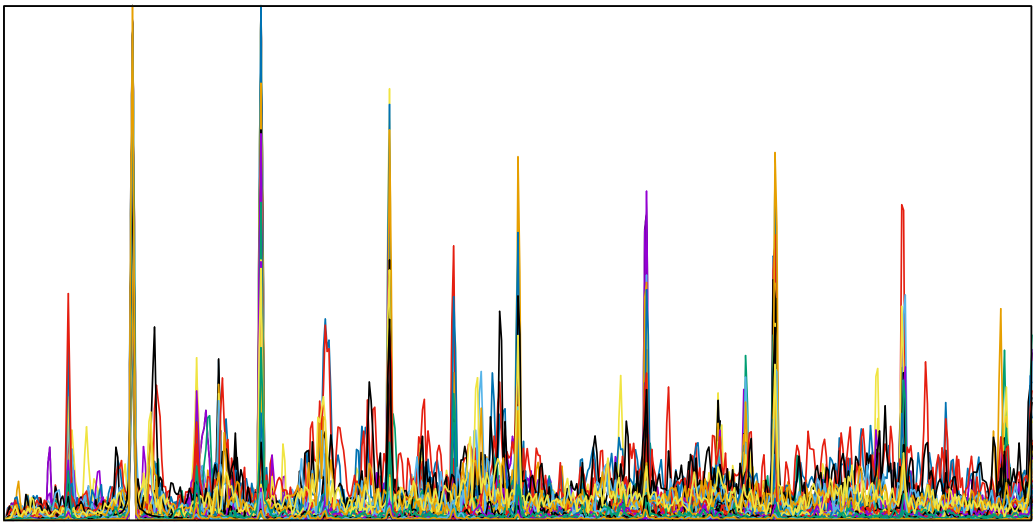

Figure 2. The Fourier transform bins of the z-axis of the same 1800 recordings as Figure 1 now aligned in the harmonic feature space.

Finally, X_k, being a complex number containing magnitude and phase information, we compute magnitude and phase. The feature vector now contains for each axis the relative magnitudes $|X_0|, \ldots, |X_K|$ or the relative phase $\sphericalangle(X_0), \ldots, \sphericalangle(X_K)$.

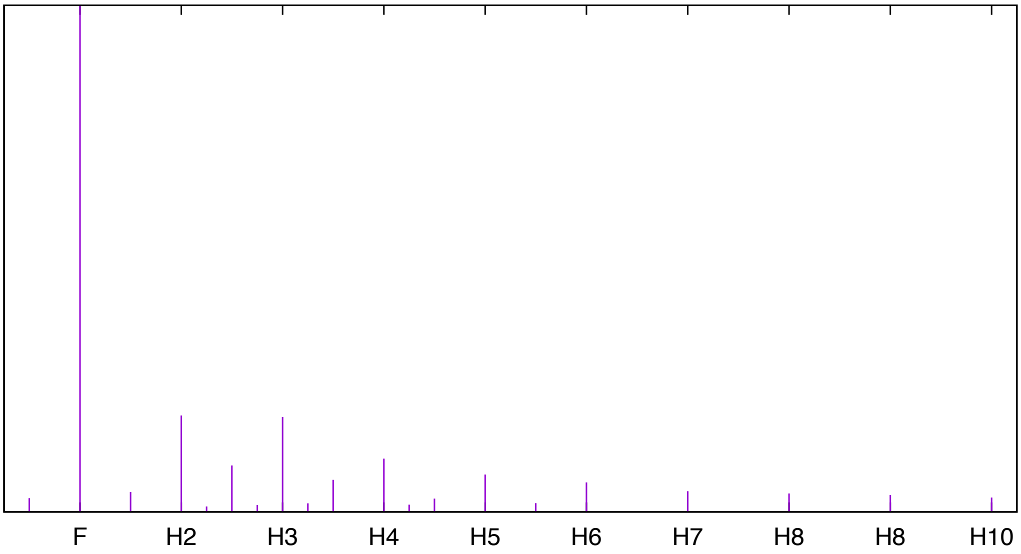

Figure 3. The means of the "statistically relevant" bins as determined by statistical reduction.

3. Statistical Significance and Reduction

Now that we have a meaningful feature vector we want to identify the statistically most significant dimensions and components and discard the rest. This is important because the harmonic feature space likely contains a significant amount of irrelevant information caused by dependencies and redundancies. Our goal is to identify the most influential dimensions and components, the ones with a very high variance [10,11]. Higher dimensionality feature spaces also lead to exponentially more complex classification [21]. We find that the reduction is not only wanted but also necessary for the performance of linear classifiers and clustering algorithms.

We first compute the *dimension-wise* (each dimension separate) mean $\mu(X)$ and *dimension-wise* standard deviation $\sigma(X)$ over all the recordings of each machine. Bins with a small variance reflect features that are generally the same across all recordings and will not assist with further discrimination of the type/class of machine. Similarly bins with a large variance reflect features that vary across recordings and might be particularly useful for such discrimination.

We eliminate all bins whose variance falls below some set threshold. This results in a very significant reduction in the dimension of our data. Figure 3 shows the bins that remain to have a variance larger than the threshold. Immediately a pattern emerges: our recordings exhibit a fingerprint of the quarter, half, and full harmonics. For a 3-axis system with an average of $N/2 = 2000$ relevant FFT data points per axis we get a reduction to about 50–80 data points as can be seen in Figure 3.

Next we examine the *component-wise* dependencies which are the dependencies between the dimensions, interpreted as the dependencies between the harmonics, e.g., possible links between the second and fourth harmonic. This is accomplished by *principal component analysis* (PCA). Again, we examine this from a variance point of view.

We compute a transform T to change the basis of our feature space such that it minimizes the reconstruction error:

$$\min_{u_m} \sum_{k=0}^{K} ||x^k - u_m u_m^\intercal x^k||$$

having M orthogonal vectors u_m. In other words, we choose the u_m to minimize the information lost when reconstructing the features x^k in the transformed space. The total variance is hence explained by a set of components, each contributing a certain amount to the total. Conveniently, as per PCA, the u_m are the eigenvectors of the covariance matrix Σ. The components can be computed by eigenvalue decomposition. The question is, how many of these components do we need?

After normalizing all eigenvectors and sorting them by the magnitude of the corresponding eigenvalue λ_m, we have to decide how much of the original variance is important to us as the total variance is explained by a sum of eigenvalues $\sigma_x^2 = \sum \lambda_m$. The *principal component* u_1 will preserve the most information as it is the direction of the data's largest variance, u_2 represents the second principal component, and so on. In our application using the Kaiser rule and choosing only components with eigenvalues larger than 1 works very well, resulting in the selection of the first 5–10 principal components of the total 50. In other words, only about 10–20% of the data meets the Kaiser criterion. Less than half of the components are needed to preserve 95% of the variance.

Using the selected principal components, the data are, hence, transformed into the new *feature space* by computing:

$$x^{\text{pcs}} = T^\intercal x, \; T = \begin{bmatrix} u_1 & u_2 & \dots & u_M \end{bmatrix}$$

The transformations are computed and stored. Note that the transformations are not label-dependent as in multi-class SVM. The stored transformations are used to transform new data that needs to be analyzed or classified. So this transform is constructed on the training set of the data, and that same transform is used on the testing set. To evaluate the influence or sensitivity of the number of components we performed repeated evaluations while changing the number of components.

4. Clustering and Classification

After the harmonic feature space transform and reduction we begin the analysis of our set of *meaningful* feature vectors. Classically there are two approaches: *supervised learning* where the number of groupings (classes) are interpreted from the labels and *unsupervised learning* where the number of groupings (clusters) are determined from some measurement in the feature space. While we have labels, we are interested in an unsupervised learning approach as we aim to find a relation between samples. As stated, clustering is the method of creating a partition of the data using some measurement criterion [15]. Once we know the number of clusters, we can easily compute the clustering. The key question is, how many clusters do we want?

The "best" number of clusters is not necessarily equal to the number of existing labels as we also would like to understand differences between machines with the same label, so a recall-accuracy is not an applicable measure. We approach the problem using an *entropy-based* measurement that represents the tradeoff of two points of view. We have labels that allow us to express ideal clustering conditions such as two machines with different drive types should be in two different clusters or machines with the same serial number should be in the same cluster. We express this mathematically in the next paragraphs.

In supervised learning we are given labels y^k, e.g., the serial numbers of the machines. This means we have a set of N disjoint classes $\Omega = \{\Omega_1, \cdots, \Omega_N\}$, Ω_n being the set of the data points belonging to the class n so $\Omega_n = \{x^k | y^k = n\}$. In clustering we compute a partition into K-clusters $C = \{C_1, \cdots, C_K\}$, again disjoint, with each x^k assigned to exactly one of the clusters.

We prefer clusterings to have recordings of the same class (e.g., type of machine) in the same cluster. This can be satisfied trivially by putting all recordings into one cluster. We also prefer clusterings that have recordings of different classes (type of machines) in different clusters, which is trivially satisfied by putting all recordings into their own cluster. Hence we propose a trade-off technique by combining these two goals.

We choose a measurement based on entropy. The entropy of a partition $S = \{S_i\}$ is given by

$$\mathcal{E}(S) = -\sum p(S_i)\log(p(S_i))$$

where $p(S_i)$ is the probability of samples belonging to S_i so $p(S_i) = \frac{|S_i|}{\sum |S_i|}$. $\mathcal{E}(S)$ is 0 if there is only one class in the partition. $\mathcal{E}(S)$ is at its maximum if there are as many classes as samples in the partition: $-\frac{1}{N}\sum \log(\frac{1}{N}) = \log(N)$.

Now given two partitions and using the Bayesian probability

$$p(X|Y) = \frac{p(X \cap Y)}{p(Y)}$$

the *conditioned entropy* is

$$\mathcal{CE}(C|\Omega) = -\sum_k p(\Omega_k)\sum_i p(C_i|\Omega_k)\log(p(C_i|\Omega_k)).$$

Note, if Ω is one single set then this is just $\mathcal{E}(C)$ and if C is one single set this is 0. So, as desired, having all samples of the same class in the same cluster means $\mathcal{CE}(C|\Omega) = 0$ and having all samples of different classes in different clusters means $\mathcal{CE}(\Omega|C) = 0$.

We define the *symmetric conditioned entropy* as

$$\mathcal{SCE}(\Omega, C) = \frac{\mathcal{CE}(\Omega|C) + \mathcal{CE}(C|\Omega)}{\mathcal{E}(\Omega) + \mathcal{E}(C)}.$$

The *symmetric conditioned entropy* now presents a trade-off, by keeping Ω fixed and varying the clustering C the measurement proves to be convex. If C is a renaming of Ω, i.e., same partition indexed differently, then $\mathcal{SCE}(\Omega, C) = 0$. The other two border cases are given above. More generally, the measure is designed for the case where the number of classes and clusters differ, i.e., non-square confusion matrices. The measurement is normalized by dividing it through $\mathcal{E}(\Omega) + \mathcal{E}(C)$ which allows us to compare clusterings with different maximal entropy. The above is different from the "normalized mutual information" referred to in the literature. The *symmetric conditioned entropy* is a permutation-independent convex measure defining a unique minimum.

Our strategy is to hierarchically split the data set into a tree; the *family tree* [19]. We start with the entire data set and compute the clustering that minimizes the *symmetric conditioned entropy* using for example the drive type or serial number as Ω, resulting in a clustering. Next we apply clustering again to these clusters until we split each cluster down to a *symmetric conditioned entropy* of 0. This tree can then be converted into a classifier by performing a Naive Bayesian classification at each level. We would like to point out that often the path in the tree leading to the decision is more interesting than the decision itself as it gives the "family history" of that particular sample or decides that this sample presents an abnormality.

Computationally, the classes Ω are fixed input data, given the X_k we use FastCluster [17] to obtain a precomputed structure containing the information of all possible clusters. Then we perform a binary search evaluating the *symmetric conditioned entropy* on the different cluster sizes.

Unlike a multi-class SVM, the advantage of this approach is that there is no class-specific scaling or normalization for unlabelled samples After applying the same reduction we computed from the learning data, we use the transformation that we stored to compute the family history of an unknown recording which we then use to classify or make decisions.

5. Case Study

Through our industrial partnership we obtained a large collection of accelerometer data from a variety of *vibrating screens*. The data was not collected strictly for this research, some machines were "recorded" sporadically and inconsistently as most customers see no reason (yet) to periodically take recordings. We identified 1500 recordings from 120 unique machines that would be suitable for our analysis. The machines' labels include serial number, data on manufacturer, drive type, size, and number of decks. However, it is possible that any given machine could have been tuned, broken, or repaired in-between recording times which we have no labelled indication of.

Clustering on time-domain features such as RPM or forces did not result in interesting information. Clustering the phases gives insight into the orbit structure, but the key information is in the relative amplitudes represented in the harmonic feature space.

To perform the analysis outlined in this paper we implemented a framework in C++ using FastCluster [17], libsvm [14], Eigen [22] to compute the PCA, and fftw3 [23] for the Fourier transforms. We perform the feature extraction from the raw sensor data and store the feature vectors for further processing. The statistical reductions, PCA, and clustering are all fast, taking only a few seconds in total.

The vibrating screens operate in the range of 10–30 Hz and were recorded with accelerometers attached at strategic positions with a sampling frequency $f_s = 1$ kHz. We transformed the data into the harmonic feature space with an FFT of size varying around 4096 placing the first harmonic into bin 120, allowing for 10 integer harmonic bins. Next all the outliers were removed, outliers are particularly detrimental to the clustering algorithm (since one outlier could "steal" a whole cluster to itself). We considered an outlier to be any sample with a z-score greater than 3 to the mean of the machine, in our data only 2% of the recordings fit this definition giving us confidence in the quality of the data. Then we proceed to perform a dimension-wise analysis to determine the dimensions with a high variance (likely corresponding to features particularly useful for discrimination). As stated, this leads to a reduction to about 80–100 independent variables. After this we perform a principal component analysis to transform the remaining dimensions into an independent feature space. Using a Kaiser Criterion as explained at the end of Section 3 we obtain 5–15 significant components with eigenvalues larger than 1.

We, as humans do, have the tendency to hang on to what we computed, working with large feature vectors. Looking at the variances we saw low numbers, and realized we should follow our principle of eliminating low variance data. Increasing the variance thresholds (both component-wise and in the PCA) gave us a reduction to feature vectors of about 40 and the PCA reduced this further to 5–10. Now the clustering works very well as we obtained scores in the high 80s to 90s comparing different classes of machines. We consider this to be an important lesson.

We noticed that 2–3 of the high harmonics (8th to 10th range) of the z-axis were considered significant, and decided to remove these manually from the feature vector, we consider them drive shaft noise. Such analysis is possible because we were able to interpret the physical meaning of the principal components, giving very interesting insights into the physics of the machines. This changed accuracy from the 60% range to the 80% and higher.

We present two examples, a large-scale view of families of many machines that are further divided, and the small scale view of clustering to individual serial numbers.

For the small scale clustering, using the serial numbers, we picked 6 machines with 4 different operating principles, so two of the classes have two machines as indicated by the naming pattern, e.g., M4.F3. We show the same data processed in 3 different ways and compute the entropy of each clustering. Table 1 shows the result of supervised learning using SVM. SVM shows no family structure, and all but one machines are in several classes, the resulting entropy is 0.18. Table 2 presents a square clustering to 6 classes. The resulting entropy is 0.10. Table 3 shows the clustering minimizing the *symmetric conditioned entropy*, the resulting entropy is 0.08. The harmonic feature vectors givs the desired separability, the *symmetric conditioned entropy* separates the 6 machines better than a 6-clustering.

Table 1. The confusion matrix for data from six types of machines classified by SVM, to show separability.

	0	**1**	**2**	**3**	**4**	**5**
M1.F1	18			2		
M2.F2	1	16				
M3.F3	5		16			
M4.F3	3			11		
M5.F4	2				8	
M6.F4	2					10

Table 2. The confusion matrix for data of 6 machines 6-clustered.

	0	**1**	**2**	**3**	**4**	**5**
M1.F1	20					
M2.F2		17				
M3.F3			19	2		
M4.F3			1	11	2	
M5.F4					10	
M6.F4			2		6	4

Table 3. The confusion matrix for data from six types of machines clustered into ten clusters as determined by optimizing the *symmetric conditioned entropy*.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
M1.F1	20									
M2.F2		17								
M3.F3			9	10	2					
M4.F3					6	6	2			
M5.F4							5		5	
M6.F4			2				4	4		2

We also generated the dendrogram presented in Figure 4 of this data colouring recordings of machines with the same serial number with the same colour. It nicely shows the family relationship, and how well the feature vectors capture the "characteristics" of the machines. Since the dendrogram of the complete data set is is too large to present, we use a heat-map.

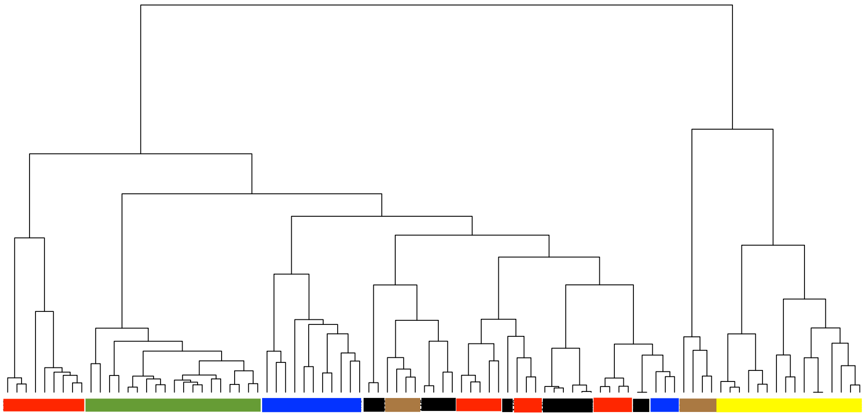

Figure 4. Dendrogram, same 6 machines as in Figure 3, colour indicates machine.

For large scale clustering and family tree generation we start with all recordings we have available, the initial entropy is 6.9. Two views are presented, an overall view and a "zoomed-in" view of one row of the family tree. We show this in two ways: Figure 5 presents the family tree of 1800 recordings using a heat map. The recordings are on the x-axis sorted by serial number, not by family as we do not know this beforehand and the y-axis represents the clustering level. The columns of equal colour represent data clustered to the same cluster, this is raw data, no outliers removal was done besides initial z-score elimination. One can observe that there is very little fragmentation, and as we increase the number of clusters (y-axis) we move into the serial number level as can be seen by the narrow columns of equal colour. Note the x-axis (the recordings) are sorted according to the clusters here. Zooming into the second-lowest level, Table 4 gives the clustering, As this level we see large-scale family resemblances. Our industry partner, who knows all these machines by name, interpreted that at this level clustering is primarily dictated by screen size and excitation source (we anonymized the serial numbers here). We then further split a cluster, the 196, resulting in Table 5, which is at the serial number level. The machines M2.F1 and M3.F1 seem to have been changed in between the recordings, but we are not sure from the data we have. Many interesting insights into vibrating screens could be gained from this analysis.

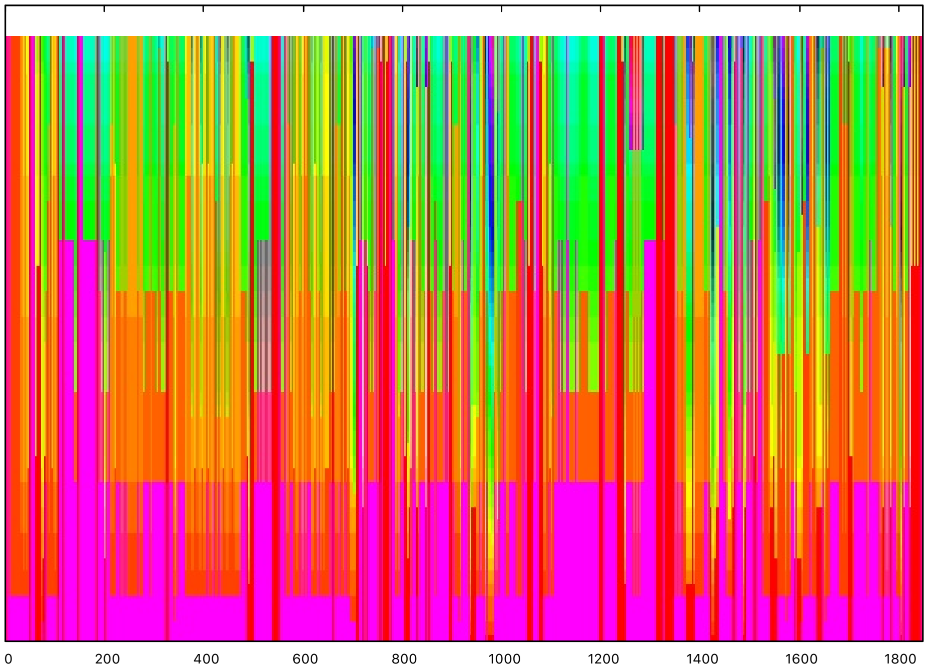

Figure 5. Family tree of 1800 recordings presented as heat map. Columns of equal colour represent clusters.

Table 4. Clustering of 684 recordings (from a cluster) re-clustered, 91% score, 2-cluster vs. *symmetric conditioned entropy*.

	0	1		0	1	2	3
C1	43	327	C1	44	321	5	
C2	314		C2	**196**		106	12

Table 5. Clustering 196 recordings from the highlighted cluster in Table 4 to serial number level.

	0	**1**	**2**	**3**	**4**
M1.F1	14	1			
M2.F1	7	7			
M3.F1	7	7			
M4.F1	11				
M5.F1	11				
M6.F1	12				
M7.F1		14			
M8.F1	4	10	2		
M1.F2		12	2		
M2.F2		12			
M3.F2		14			
M4.F2	1	17			
M5.F2	2	7			
M6.F2		7		2	
M7.F2			3		10

6. Conclusions and Future Works

The presented methods provides a systematic approach to the plant engineer to obtain meaningful insight into the data without the need for an existing model. The harmonic feature space and the statistical reduction results in high-quality feature vectors, allowing us to explore similarities and differences of rotating machinery. The significant statistical reduction was key, eliminating components with low variance results in better separability of the data.

The *symmetric conditioned entropy* allowed us to compute a unique number of clusters optimizing the tradeoff of two clustering goals. Clustering using the *symmetric conditioned entropy* results in better insights than clustering into a fixed number of classes. In this application clustering performs better than SVM especially as no class-specific transforms are needed.

The approach presented in this paper is general, and initial experiments show that it is well suited for bearing fault analysis. While we only examined periodic operating machinery we are developing a complementary approach for non-stationary operation conditions using wavelet-based features.

Author Contributions: The contibutions of the E.H. and M.v.M. are equal. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by MITACS project IT17587.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, R.; Yang, B.; Zio, E.; Chen, X. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mech. Syst. Signal Process.* **2018**, *108*, 33–47. [CrossRef]
2. Caesarendra, W.; Tjahjowidodo, T. A review of feature extraction methods in vibration-based condition monitoring and its application for degradation trend estimation of low-speed slew bearing. *Machines* **2017**, *5*, 21. [CrossRef]
3. de A Boldt, F.; Rauber, T.W.; Varejão, F.M. Feature Extraction and Selection for Automatic Fault Diagnosis of Rotating Machinery. In Proceedings of the X Encontro Nacional de Inteligência Artificial e Computacional (ENIAC), Fortaleza, CE, Brazil, 19–24 October 2013.
4. Ullah, I.; Arbab, N.; Gul, W. State of the Art Vibration Analysis of Electrical Rotating Machines. *J. Electr. Eng.* **2017**, *5*, 84–99.
5. Pinheiro, A.A.; Brandao, I.M.; Da Costa, C. Vibration Analysis in Turbomachines Using Machine Learning Techniques. *Eur. J. Eng. Technol. Res.* **2019**, *4*, 12–16.
6. Yunusa-Kaltungo, A.; Cao, R. Towards Developing an Automated Faults Characterisation Framework for Rotating Machines. Part 1: Rotor-Related Faults. *Energies* **2020**, *13*, 1394. [CrossRef]
7. Hoang, D.T.; Kang, H.J. A survey on Deep Learning based bearing fault diagnosis. *Neurocomputing* **2019**, *335*, 327–335. [CrossRef]
8. Zhang, S.; Zhang, S.; Wang, B.; Habetler, T. Machine learning and deep learning algorithms for bearing fault diagnostics—A comprehensive review. *arXiv* **2019**, arXiv:1901.08247.
9. Bagnall, A.; Lines, J.; Bostrom, A.; Large, J.; Keogh, E. The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Discov.* **2017**, *31*, 606–660. [CrossRef] [PubMed]
10. Sorzano, C.; Vargas, J.; Pascual-Montano, A. A survey of dimensionality reduction techniques. *arXiv* **2014**, arXiv:1403.2877.
11. Jolliffe, I.T.; Cadima, J. Principal component analysis: A review and recent developments. *Philos. Trans. R. Soc. A* **2016**, *374*, 20150202. [CrossRef] [PubMed]
12. Borhana, A.A.; Bin Mustaffa Kamal, D.D.; Latif, S.D.; Ali, Y.H.; Ahmed Almahfoodh, A.N.; El-Shafie, A. Fault Detection of Bearing using Support Vector Machine-SVM. In Proceedings of the 2020 8th International Conference on Information Technology and Multimedia (ICIMU), Singapor, Malaysia, 24–25 August 2020; pp. 309–315. [CrossRef]
13. Liu, Y.; Liu, T. Rotating Machinery Fault Diagnosis Based on Support Vector Machine. In Proceedings of the 2010 International Conference on Intelligent Computing and Cognitive Informatics, Kuala Lumpur, Malaysia, 22–23 June 2010; pp. 71–74. [CrossRef]
14. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27:1–27:27. Available online: http://www.csie.ntu.edu.tw/~cjlin/libsvm (accessed on 1 July 2021). [CrossRef]
15. Shirkhorshidi, A.S.; Aghabozorgi, S.; Wah, T.Y. A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data. *PLoS ONE* **2015**, *10*, e0144059. [CrossRef]
16. Li, T.; Ma, S.; Ogihara, M. Entropy-Based Criterion in Categorical Clustering. In Proceedings of the 21st International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004.
17. Mullner, D. Modern hierarchical, agglomerative clustering algorithms. *arXiv* **2011**, arXiv:1109.2378.

18. Aldana-Bobadilla, E.; Kuri-Morales, A. A Clustering Method Based on the Maximum Entropy Principle. *Entropy* **2015**, *17*, 151–180. [CrossRef]
19. Aghagolzadeh, M.; Soltanian-Zadeh, H.; Araabi, B.N. Information Theoretic Hierarchical Clustering. *Entropy* **2011**, *13*, 450–465. [CrossRef]
20. Zieliński, T.P.; Duda, K. Frequency and damping estimation methods—An overview. *Metrol. Meas. Syst.* **2011**, *18*, 505–528. [CrossRef]
21. Verleysen, M.; François, D. The Curse of Dimensionality in Data Mining and Time Series Prediction. In *Computational Intelligence and Bioinspired Systems*; Cabestany, J., Prieto, A., Sandoval, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 758–770.
22. Eigen v3. 2010. Available online: http://eigen.tuxfamily.org (accessed on 1 June 2021).
23. Padua, D. FFTW. In *Encyclopedia of Parallel Computing*; Springer: Boston, MA, USA, 2011; p. 671. [CrossRef]