

Machine Learning-Based Unbalance Detection of a Rotating Shaft Using Vibration Data

Oliver Mey, Willi Neudeck, André Schneider and Olaf Enge-Rosenblatt

Fraunhofer IIS/EAS, Fraunhofer Institute for Integrated Circuits

Division Engineering of Adaptive Systems

Dresden, Germany

oliver.mey@eas.iis.fraunhofer.de

Abstract—Fault detection at rotating machinery with the help of vibration sensors offers the possibility to detect damage to machines at an early stage and to prevent production downtimes by taking appropriate measures. The analysis of the vibration data using methods of machine learning promises a significant reduction in the associated analysis effort and a further improvement in diagnostic accuracy. Here we publish a dataset which is used as a basis for the development and evaluation of algorithms for unbalance detection. For this purpose, unbalances of various sizes were attached to a rotating shaft using a 3D-printed holder. In a speed range from approx. 630 RPM to 2330 RPM, three sensors were used to record vibrations on the rotating shaft at a sampling rate of 4096 values per second. A development and an evaluation dataset are available for each unbalance strength. Using the dataset recorded in this way, fully connected and convolutional neural networks, Hidden Markov Models and Random Forest classifications on the basis of automatically extracted time series features were tested. With a prediction accuracy of 98.6 % on the evaluation dataset, the best result could be achieved with a fully-connected neural network that receives the scaled FFT-transformed vibration data as input.

I. INTRODUCTION

Progress in the field of machine learning has led to impressive results in recent years, for example in the areas of image recognition [1]–[4], natural language processing [5]–[9] or reinforcement learning [10]–[13]. In addition to these examples, which are very present in the media, these algorithms also offer great potential for industrial applications [14]–[18]. For example, the analysis of vibrations on rotating shafts to detect unbalances or to detect damage to roller bearings has proven to be very promising [19]–[27]. Here, we focus on the first mentioned use case. Unbalances on rotating shafts can cause decreased lifetimes of bearings or other parts of the machinery and, therefore, lead to additional costs. Hence, early detection of unbalances helps to minimize maintenance expenses, to avoid unnecessary production stops and to increase the service life of machines. Algorithmic detection of unbalances is accompanied with the least additional effort. The automation achieved in this way also enables live analysis of streamed data, which means that unbalances can be detected and corrected with almost no time delay, even before potential damage to the drive train occurs.

We observe that there are only a few publicly available condition monitoring (CM) datasets with the help of which

algorithms can be tested and compared. There are e.g. datasets for CM with hydraulic systems [28] and for detecting bearing damage [29]–[31], but there seems to be no dataset for detecting unbalances, which in turn can be a cause of bearing damage. For this reason, we publish a dataset for the detection of unbalances based on vibration data along with this study (available in the Fraunhofer Fordatis database [32]). In addition, we carry out analyses to determine which algorithms can detect the unbalance as accurately as possible and up to which unbalance strength these can still be reliably recognized by each algorithm. The Python code used for the investigations conducted in this study is open-sourced in a Github repository [33].

II. MEASUREMENT SETUP

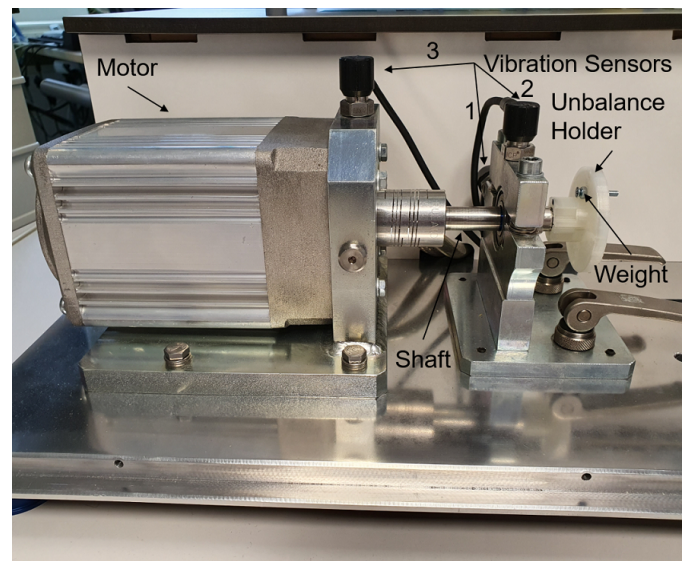


Fig. 1. Measurement setup

The setup for the simulation of defined unbalances and the measurement of the resulting vibrations is powered by an electronically commutated DC motor (WEG GmbH, type UE 511 T), which is controlled by a motor controller (WEG GmbH, type W2300) and is fixed to the aluminum base plate by means of a galvanized steel bracket. The motor controller allows for a rotation speed between approximately 300 and

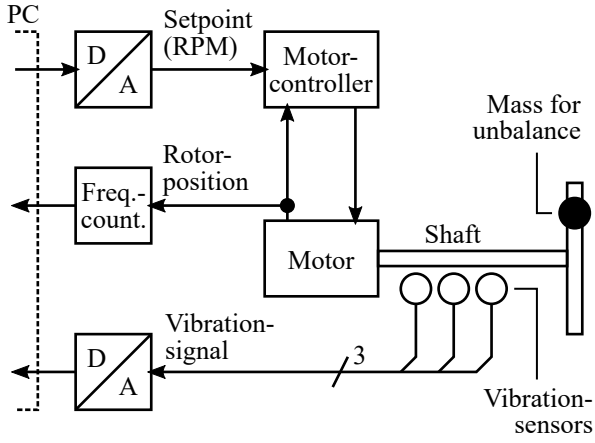


Fig. 2. Block diagram of the measurement setup

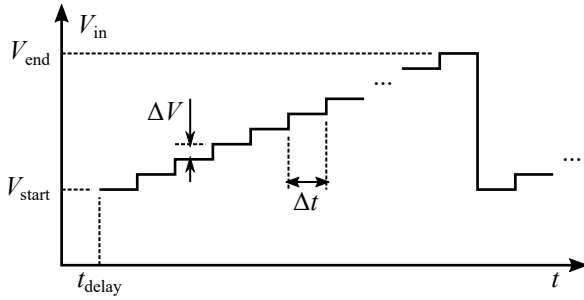


Fig. 3. Setpoint for the rotation speed during data acquisition. The setpoint is encoded as a Voltage V_{in} , which is varied according to the diagram above.

2300 revolutions per minute (RPM), which can be continuously adjusted by varying a voltage that is applied to the motor controller. The motor powers a shaft with a diameter of 12 mm which is connected to another shaft of the same diameter and a length of 75 mm by a coupling (Orbit Antriebstechnik GmbH, type PCMR29-12-12-A). This shaft in turn passes through a roller bearing which is clamped in a roller bearing block (material: galvanized steel). The unbalance holder is attached directly behind it. This part was made using a 3D printer (Ultimaker 3, material: nylon) and consists of a disc (diameter: 52 mm) with axially symmetric recesses, in which weights can be inserted to simulate unbalances. Vibration sensors (PCB Synotech GmbH, type PCB-M607A11 / M001AC) are attached to both the bearing block and the motor mounting and are read out using a 4-channel data acquisition system (PCB Synotech GmbH, type FRE-DT9837). As shown in Figure 2, the rotation speed of the motor is acquired using a frequency counter in the DT9837, which digitizes the periodicity of the rotor position signal from the motor. A photo of the measurement setup is shown in Figure 1.

III. THE DATASET

Using the setup described in Section II, vibration data for unbalances of different sizes was recorded. By varying the level of unbalance, different levels of difficulty can be

achieved, since smaller unbalances obviously influence the signals at the vibration sensors to a lesser extent. Several further requirements were taken into account: The dataset should be reproducible, relevant for industrial applications and it should represent a use case that is as realistic as possible. This requires the recording of vibration data for varying rotational speed, as an unbalance detector might have to work under varying conditions in some industrial applications. To ensure a high level of (re-)usability of the dataset, we provide tabular data in the *csv*-format.

In total, datasets for 4 different unbalance strengths were recorded as well as one dataset with the unbalance holder without additional weight (i.e. without unbalance). Each dataset is provided as a *csv*-file with five columns:

V_{in}	The input voltage to the motor controller V_{in} (in V),
Measured_RPM	the rotation speed of the motor (in RPM; computed from speed measurements using the DT9837),
Vibration_1	the signal from the first vibration sensor,
Vibration_2	the signal from the second vibration sensor, and
Vibration_3	the signal from the third vibration sensor.

The sampling rate in each column amounts to 4096 values per second (the rotation speed has been upsampled accordingly).

In order to enable a comparable division into a development dataset and an evaluation dataset, separate measurements were taken for each unbalance strength, respectively. This separation can be recognized in the names of the *csv*-files, which are of the form “1D.csv”: The digit describes the unbalance strength (“0” = no unbalance, “4” = strong unbalance), and the letter describes the intended use of the dataset (“D” = development or training, “E” = evaluation).

The unbalance on the measurement setup was completely dismantled and reassembled between the measurement of the development and the evaluation datasets. This causes an additional variability between them and increases the significance of the evaluation of the algorithms that are to be trained on the data. For the development datasets, the motor voltage V_{in} was increased from $V_{start} = 2.0$ V to $V_{end} = 10.05$ V in steps of $\Delta V = 0.05$ V; see Fig. 3. For the evaluation datasets, the motor voltage was increased in steps of $\Delta V = 0.1$ V from $V_{start} = 4.0$ V to $V_{end} = 8.1$ V. At each step the motor voltage value is kept constant for $\Delta t = 20$ s. The voltage profiles were run through twice for each data record. The rotation speed of the motor was found to be approximately

$$\frac{n}{\text{RPM}} \approx 212 \cdot \frac{V_{in}}{V} + 209$$

within the range $2 \text{ V} \leq V_{in} \leq 10 \text{ V}$.

An overview of the parameters of the recorded datasets can be found in Table I. This includes the masses and radii for all the used unbalances. Since the absolute value of the centrifugal force \vec{F}_{Cf} as a function of the rotation speed ω can under a point mass approximation be expressed as

$$|\vec{F}_{Cf}(\omega)| = mr\omega^2,$$

TABLE I
PARAMETERS OF USED DATASETS

ID	Radius [mm]	Mass [g]	Unbalance Factor [mm g]	Number of Samples Development	Evaluation
0D / 0E	-	0	0	6438	1670
1D / 1E	14 ± 0.1	3.281 ± 0.003	45.9 ± 1.4	6434	1673
2D / 2E	18.5 ± 0.1	3.281 ± 0.003	60.7 ± 1.9	6434	1669
3D / 3E	23 ± 0.1	3.281 ± 0.003	75.5 ± 2.3	6430	1672
4D / 4E	23 ± 0.1	6.614 ± 0.007	152.1 ± 2.3	6430	1675

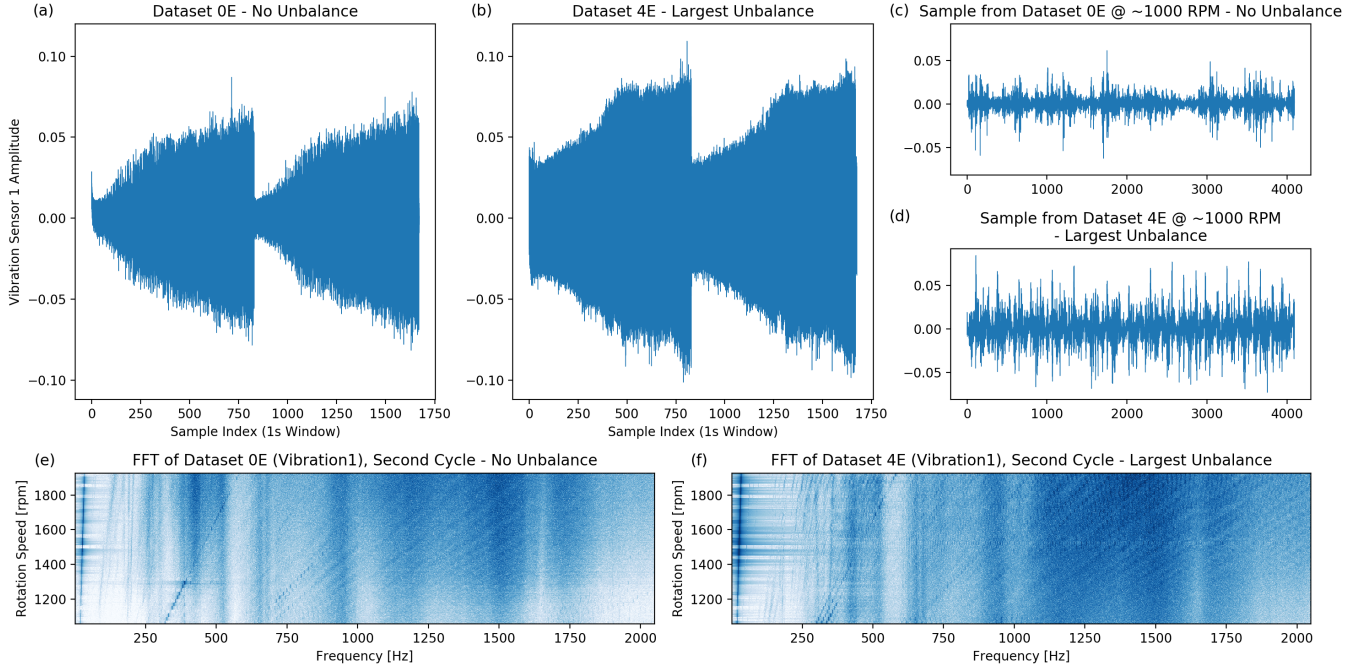


Fig. 4. Example measurements from the dataset: Data from vibration sensor 1 for a complete measurement for the case of no unbalance and the largest unbalance ((a) and (b), respectively). For both cases, also a one second sample is extracted ((c) and (d), respectively), as well as the FFT transformation of the second measurement cycle ((e) and (f), respectively).

the product of the mass m and the radius r is a direct measure of the unbalance strength. In Table I there is also a column called *Number of Samples* for the development and the evaluation dataset. To calculate these values, the first 50,000 values from each dataset were removed (approx. 10 s) and the rest was then divided into windows, each corresponding to one second or 4096 values. The number of samples, therefore, equals the time range of the usable data of each measurement. When calculating prediction accuracies based on these samples, though, it has to be taken into account that they are part of a continuous measurement and are therefore not completely independent from each other. Nevertheless, these accuracies are useful to compare the classification performance of different algorithms on the given datasets.

An overview of the amplitude progression at vibration sensor 1 over an entire measurement in the cases of no unbalance and the largest unbalance is shown in Figures 4(a) and 4(b). For each of the two cases an example curve for one second as well as the FFT transformation of a part of the measurement data is also depicted (Figure 4(c)-(f)).

IV. CLASSIFICATION OF THE UNBALANCE STATE

A. Approach 1: Convolutional Neural Network on Raw Sensor Data

Convolutional Neural Networks (CNNs) are able to recognize patterns in data and to perform classification tasks based on these recognized patterns. An unbalance classification with CNNs, which receive the windowed data directly as input, is therefore promising. The advantage here is that no further data preprocessing is necessary and the effort involved in creating the algorithm is therefore comparatively low. Windowed samples from the data stream ‘Vibration_1’ were directly used as input. Figure 5 shows the CNN architecture used for the classification in this study. Since it was shown that an over-parameterization of a neural network with regard to the number of training samples can have a positive effect on the overall performance [34], the depth of the network and thus the number of model parameters was varied. This was achieved by varying the number of convolutional blocks N_{conv} , consisting of a convolutional layer, batch normalization, activation function and max pooling. After the convolution blocks, one

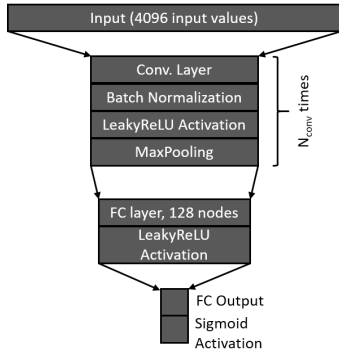


Fig. 5. Sketch of the used neural network architecture for the classification of the raw vibration samples. N_{conv} describes the number of hidden convolutional and pooling layers used.

fully-connected (FC) layer leads to the final output layer. Since only the task, whether or not an unbalance is present was used for classification, the output layer consists of one single node with a sigmoid activation.

To better monitor the training process, the development dataset was randomly divided into 90 % training data and 10 % test data. During the training phase, the error function on the training data was minimized. The error function based on the test data was also monitored during the training. The model with the lowest error on the test data was kept in each case and afterwards tested on the corresponding evaluation datasets of the same unbalance factors.

For a first attempt, the classification of whether an unbalance is present or not was trained using the data record without unbalance and only one single data record with unbalance each time. Afterwards, all trained models were tested on the corresponding evaluation datasets of the same unbalance factors. The resulting accuracies are shown in Figure 6(a).

Overall, a rather weak prediction accuracy can be observed in this classification task. It is striking that in particular the second smallest unbalance can hardly be distinguished from the not unbalanced case (dataset pair ‘0E’ and ‘2E’). With a high prediction accuracy this is only possible for the largest unbalance (dataset pair ‘0E’ and ‘4E’). With regard to the depth of the CNN, the best results are achieved with 4 and 2 convolutional blocks, with an average of 79.0 % and 82.2 % respectively.

In a further experiment, not only one unbalance strength at a time was used as training data, but all. Classifications were nevertheless made as to whether there was an unbalance or not. Therefore there is also one accuracy score describing the performance of the classification algorithm on this task. However, to gain insights into the distribution of correct and incorrect classifications, the resulting models were additionally evaluated in relation to the individual data records, resulting in one accuracy score per unbalance class (plotted in Figure 6(b)). With this classification task, a significantly better performance of the algorithms used can be observed. On the one hand, this is obviously due to the larger amount of training data

available for each individual training. On the other hand, this variant also trains higher variability in the exact mounting of the unbalance. Since the unbalances have been completely disassembled between the measurement of the development and the evaluation datasets, minor changes in the vibration behavior of the entire system can be caused. As in the previous experiment, the best results are achieved with CNNs of 2 (94.0 %) and 4 (93.6 %) convolution blocks. With only one convolution block a deviating behavior is obtained: A high detection accuracy, even with small unbalances, is achieved by a reduced detection accuracy of the unbalanced case.

The same trained models were also evaluated as a function of the rotation speed (shown in Figure 6(c)). It can be seen that there are areas, for example around 1600 RPM, where all algorithms have a high prediction accuracy and areas in which this accuracy is low for all models (for example around 1100 RPM). In other areas, however, the prediction accuracy is widely spread (around 1500 RPM).

B. Approach 2: Fully-Connected Neural Network on FFT-transformed Data

For this approach, the FFT was calculated for each of the windows of one second or 4096 values of the first vibration sensor stream (‘Vibration_1’). According to the Shannon-Nyquist sampling theorem, this results in 2048 physically meaningful Fourier coefficients for each window, which can be used for classification. Again, development dataset transformed in this way was randomly divided into 90 % training data and 10 % test data. Afterwards, the FFT data were scaled as follows: For each Fourier coefficient, the respective median and interquartile spacing of quantiles 5 and 95 was calculated based on the extent of the training dataset (2048 values for the median and the interquartile spacing, respectively). The median values were then subtracted from the FFT values and the result was divided by the interquartile values. Fully connected (FC) neural networks were then trained on the training data. An illustration of the used network architectures is shown in Figure 7. The input consisting of 2048 Fourier coefficients in each sample was followed by N_{hidden} hidden and fully connected layers with LeakyReLU activation and the output layer. Neural networks of this type with zero (equivalent to logistic regression) to four hidden layers were trained using the respective training data. As described in approach 1 (Section IV-A), in the first experiment the dataset of the unbalance-free case and the datasets with unbalance were paired and the respective models were trained based on these datasets and evaluated on the corresponding dataset pairs from the evaluation data.

The resulting accuracies are shown in Figure 6(d). It is apparent that the trained models are only partially able to accurately conduct the classification between zero and the smaller unbalances. While the largest unbalance could be classified correctly in almost every case, the picture is inconsistent in the remaining cases. The data record ‘2E’ can also be classified with a high accuracy, while this in turn works worse for the data record ‘3E’. A monotonically increasing prediction

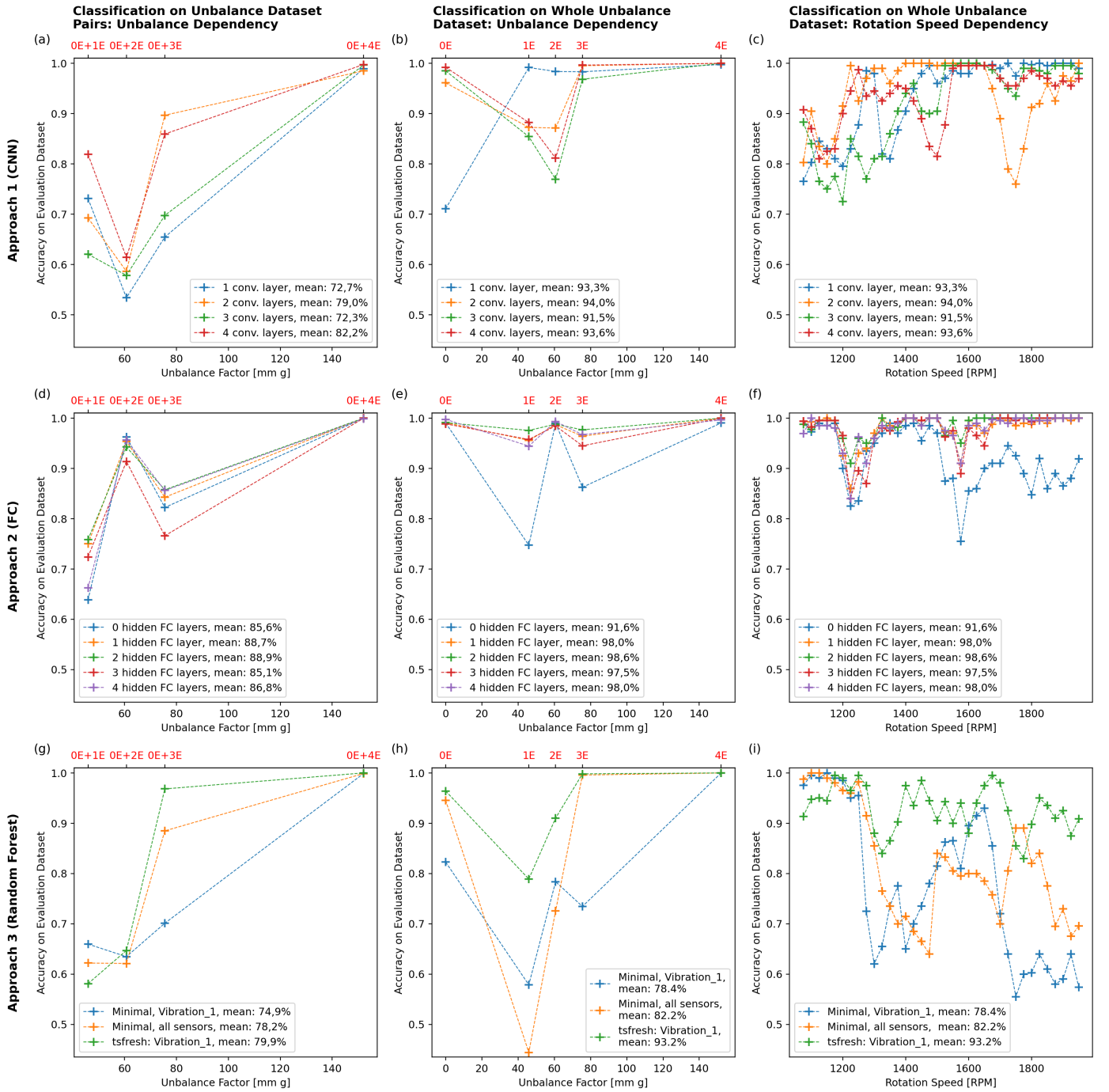


Fig. 6. Evaluation accuracies of the used classification approaches 1 (a-c), 2 (d-f) and 3 (g-i) on the task whether or not an unbalance exists. The results for models trained and tested using pairs of the dataset without unbalance and one dataset of a single unbalance strength are plotted in the first column (a,d,g). The results for models trained and tested using all measured unbalance strengths are shown in the second column (b,e,h). The models from the second column are additionally evaluated as a function of the rotation speed (third column, (c,f,i)). In the first two columns, the corresponding IDs of the datasets that were used for the evaluation are marked in red above the diagrams. Connections between the data points are only shown for the sake of clarity and do not imply the continuity of the observed relationships.

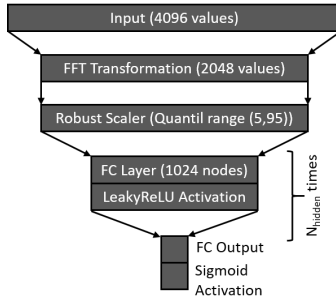


Fig. 7. Sketch of the used neural network architecture for the classification of the FFT-transformed vibration samples. N_{hidden} describes the number of hidden FC layers used.

accuracy as a function of the unbalance strength was expected. One reason for the deviation observed here could be again effects caused by reassembling of the unbalance after each measurement. Additionally, there is a slight trend visible, that neural networks with one or two layers reach a better overall performance in this task, possibly a better generalization could be achieved in these cases.

In the next experiment, again all datasets instead of pairs of datasets were used and classification was conducted, whether or not there was an unbalance at all. The results are depicted in Figure 6(e). The number of hidden FC layers was again varied between 0 and 4. Overall, an accuracy of 0.916 (zero hidden layers) to 0.986 (two hidden layers) was achieved on the evaluation dataset for the classification task. While the largest unbalance is recognized almost perfectly by all methods, there is no clear tendency for the other unbalance strengths, similar to the previous experiment. As with the CNN approach, the larger amount of training data also leads to better performance overall.

For the rotation speed dependent evaluation (Figure 6(f)), it can be seen that all models have a drop in prediction accuracy in the ranges around 1200 RPM and 1550 RPM. Outside these ranges, all algorithms except those with zero hidden layers achieve an accuracy of almost 100 %. One reason for the worse performance in the described ranges could be resonant oscillations of the measurement setup, resulting in a reduction of the signal-to-noise ratio of the signals caused by the unbalance.

C. Approach 3: Random Forest on Automatically Extracted Timeseries Features

In order to compare the generalization ability of all employed algorithms to a common baseline, and to find out to what extent a higher computational effort has an impact on a possibly improved prediction accuracy, classification was carried out using a minimum set of features. This small feature set consists of the mean of the ‘Measured_RPM’ values, as well as the standard deviation and the kurtosis of the vibration values, which were calculated for each of the previously partitioned windows of the datasets. This feature calculation was carried out in two variants: First, standard deviation and

kurtosis were only calculated for ‘Vibration_1’, resulting in a total of 3 features (including the mean of the ‘Measured_RPM’ values). In the second variant, all three vibration sensors were used (7 features in total). Both variants shall be denoted to as ‘minimal features’. A Random Forest model was then trained on these minimal features. As with the previous classification approaches, the classification training was conducted once with dataset pairs consisting of one unbalance strength and the unbalance-free case each and once with all existing unbalance strengths. The classification results of the evaluation are shown in the Figures 6(g)-(i). It can be seen that the highest unbalance can be detected almost perfectly by using only 3 features in both experiments. Using 7 features and trained on the whole dataset, even the dataset ‘3E’ can be classified close to 100 % accuracy. With the smaller unbalances, on the other hand, there is a significant decrease in the prediction accuracy and also the unbalance-free case can only be detected to 82.2 % (3 features) or 94.6 % (7 features) in the second experiment (Figure 6(h)). When looking at the rotation-speed-dependent evaluation, the high accuracy below 1200 RPM is particularly striking. In this range, the classification with the minimum set of features even achieves a significantly better result than with approach 1.

Besides the mentioned minimal features, the Python package *tsfresh* offers the possibility of computing a much wider range of features describing time series [35]. Using *tsfresh* (version 0.14.1), 748 features belonging to the class `EfficientFCParameters()` were extracted for ‘Vibration_1’ and afterwards used as input for a random forest algorithm. Since the classification task and algorithm remained the same, and only the number of input features changed, the prediction results on the evaluation dataset are depicted in the Figures 6(g)-(i), as well (green curve). In particular, a significant improvement in the detection rate for the smaller unbalances compared to the minimal features causes that, with a total prediction accuracy of 93.2 % when trained on all unbalance strengths and a mean prediction accuracy of 79.9 % when trained with the dataset pairs, a level similar to that of the CNNs in approach 1 (Section IV-A) is achieved overall.

D. Approach 4: Hidden Markov Model

One can obtain an outline of hidden markov models (HMMs) from the article [36]. Beyond speech recognition (e.g. in the *Sphinx* system [37], [38]), HMMs have also been used in biology (e.g. protein structure and genome research [39]), sports (e.g. recognition of sports activities [40]), and other use cases. In the field of condition monitoring, HMMs have e.g. been employed for the detection of defective roller bearings [41]. The latter paper has had a certain influence on the design of the unbalance detector, that is used in the present section:

A possible approach to determine the unbalance state using a hidden markov model (HMM) is shown in Figure 8: The input signal (4096 consecutive values from ‘Vibration_1’ were used) is cut into (possibly overlapping) snippets of a fixed length. For each snippet, the mel-frequency cepstral components (MFCCs) are computed as features, which are then input into a HMM that is trained to recognize data without

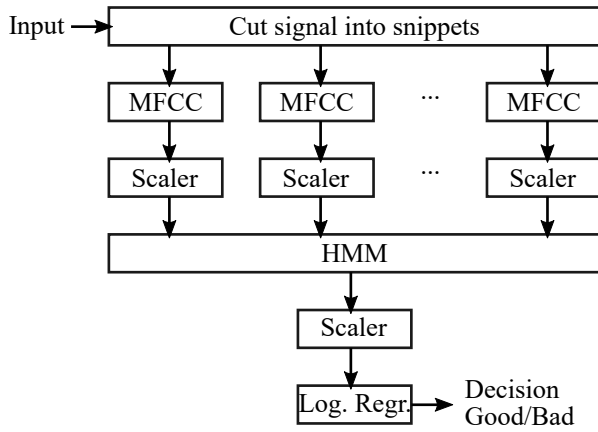


Fig. 8. Block diagram of the unbalance detector using HMM and MFCCs.

unbalance. To facilitate the interpretation of the HMM output, logistic regression is used to decide whether a given input signal results from a measurement with or without unbalance. The scalers in Figure 8 simplify the training process.

Because the MFCC features are sensitive to variations in the rotation speed, it was decided to train several models for different speeds. The training data was therefore assembled in the following way: One-second samples of the ‘Vibration_1’ signal (from the ‘0D’ and ‘3D’ datasets) were selected such that the speed (‘Measured_RPM’) is always within a certain interval. The training data is then randomly split into three sets. One set is used to train the first scaler and the HMM (using data from ‘0D’ only). The second set is used to train the second scaler and the logistic regression to recognize measurements with unbalance. The third set is used to determine hyperparameters (number of MFCC features, number of HMM states, snippet length and overlap), that maximize the balanced accuracy.

During development of the HMM approach, it was noticed that the MFCC features within one-second samples of the measurement without unbalance appear to be relatively stationary. And the hyperparameter optimization often results in the use of only one (!) HMM state. The problem at hand might therefore be inadequate for HMMs (which are usually applied to instationary processes).

Figure 9 shows the results of the HMM approach to unbalance detection. It can be seen, that the balanced accuracy is in the range

- 0.56–0.93, mean = 0.65 (for the union of ‘0E’ and ‘1E’),
- 0.57–1.00, mean = 0.80 (for the union of ‘0E’ and ‘2E’),
- 0.74–1.00, mean = 0.95 (for the union of ‘0E’ and ‘3E’), and
- 0.74–1.00, mean = 0.95 (for the union of ‘0E’ and ‘4E’).

V. SUMMARY AND OUTLOOK

For this study, a dataset with vibration data for the classification of unbalance on a rotating shaft with variable speed and unbalance strength was created. Various approaches to

solve the associated classification task were tested. The largest unbalance could be detected by all algorithms with almost perfect prediction accuracy, even if only 3 characteristic values per sample were used for the classification. With the smaller unbalances, on the other hand, wider variations between the different approaches were found. The best way to classify the dataset was to use an FC network with two hidden layers, which received the scaled FFT-transformed vibration data as input. Measured on the entire evaluation dataset, 98.6 % of the cases could be classified correctly. In addition, the examined models showed a very different behavior regarding the dependence on the speed. In future studies, this behavior could be exploited by building ensembles of different models to further increase the prediction accuracy. Strengths and weaknesses of individual models in the different speed ranges would then at least partially compensate each other. Moreover, for the further improvement of the models as well as the understanding of the classifications, for example in a productive company, efforts with regard to enabling a traceability and explainability of the models used are necessary.

REFERENCES

- [1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [2] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” in *AAAI Conference on Artificial Intelligence*, 2017.
- [5] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching Word Vectors with Subword Information,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.
- [6] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, “Supervised Learning of Universal Sentence Representations from Natural Language Inference Data,” May. 2017. Accessed: Mar. 4 2020.
- [7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Oct. 2018.
- [8] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent Trends in Deep Learning Based Natural Language Processing [Review Article],” *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, 2018.
- [9] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “XLNet: Generalized Autoregressive Pretraining for Language Understanding,” *Advances in neural information processing systems*, 2019.
- [10] V. Mnih et al., “Playing Atari with Deep Reinforcement Learning,” Dec. 2013.
- [11] V. Mnih et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [12] D. Silver et al., “Mastering the game of Go without human knowledge,” *Nature*, vol. 550, no. 7676, pp. 354–359, 2017, doi: 10.1038/nature24270.
- [13] D. Silver et al., “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play,” *Science (New York, N.Y.)*, vol. 362, no. 6419, pp. 1140–1144, 2018.
- [14] Z. Li, Y. Wang, and K.-S. Wang, “Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario,” *Adv. Manuf.*, vol. 5, no. 4, pp. 377–387, 2017.
- [15] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, “Deep learning and its applications to machine health monitoring,” *Mechanical Systems and Signal Processing*, vol. 115, pp. 213–237, 2019.

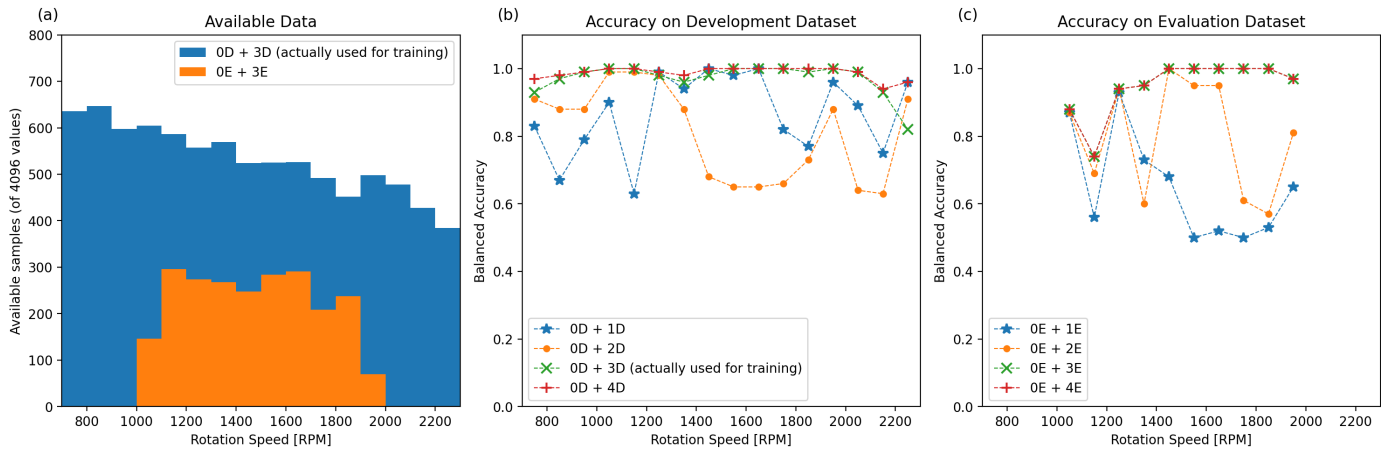


Fig. 9. Classification results using the Hidden Markov Model with MFCC features

- [16] C. Bayer, O. Enge-Rosenblatt, M. Bator, and U. Mönks, "Sensorless drive diagnosis using automated feature extraction, significance ranking and reduction," in *IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, 2013, Cagliari, Italy, Proceedings.
- [17] C. Lessmeier, O. Enge-Rosenblatt, C. Bayer, and D. Zimmer, "Data Acquisition and Signal Analysis from Measured Motor Currents for Defect Detection in Electromechanical Drive Systems," in *2nd European Conference of the Prognostics and Health Management Society 2014 – PHME 2014*, 2014, Nantes, France, Proceedings.
- [18] O. Enge-Rosenblatt, C. Bayer, and A. Brüning, "Complex Analysis of Monitoring and Process Data for Effective Manufacturing," in *16th European Advanced Process Control and Manufacturing Conference – APCM*, Reutlingen, Germany, 11-13 April, 2016, Proceedings.
- [19] A. Dadouche, A. Rezaei, V. Wickramasinghe, W. Dmochowski, J. W. Bird, and F. Nitzsche, "Sensitivity of Air-Coupled Ultrasound and Eddy Current Sensors to Bearing Fault Detection," *Tribology Transactions*, vol. 51, no. 3, pp. 310–323, 2008.
- [20] S. Gopinath, "Study on electric motor mass unbalance based on vibration monitoring analysis technique," in *2nd International Conference on Mechanical and Electrical Technology (ICMET)*, 2010, Singapore.
- [21] O. Janssens et al., "Convolutional Neural Network Based Fault Detection for Rotating Machinery," *Journal of Sound and Vibration*, vol. 377, pp. 331–345, 2016.
- [22] J. Juan Carbajal-Hernández, L. P. Sánchez-Fernández, I. Hernández-Bautista, J. d. J. Medel-Juárez, and L. A. Sánchez-Pérez, "Classification of unbalance and misalignment in induction motors using orbital analysis and associative memories," *Neurocomputing*, vol. 175, pp. 838–850, 2016.
- [23] M. M. Tahir, A. Hussain, S. Badshah, A. Q. Khan, and N. Iqbal, "Classification of unbalance and misalignment faults in rotor using multi-axis time domain features," in *2016 International Conference on Emerging Technologies*, Islamabad.
- [24] Z. Chen, S. Deng, X. Chen, C. Li, R.-V. Sanchez, and H. Qin, "Deep neural networks-based rolling bearing fault diagnosis," *Microelectronics Reliability*, vol. 75, pp. 327–333, 2017.
- [25] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: A review," *Mechanical Systems and Signal Processing*, vol. 108, pp. 33–47, 2018.
- [26] L. Guo, Y. Lei, S. Xing, T. Yan, and N. Li, "Deep Convolutional Transfer Learning Network: A New Method for Intelligent Fault Diagnosis of Machines With Unlabeled Data," *IEEE Trans. Ind. Electron.*, vol. 66, no. 9, pp. 7316–7325, 2019.
- [27] Y. Tian and X. Liu, "A Deep Adaptive Learning Method for Rolling Bearing Fault Diagnosis Using Immunity," *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 750–762, 2019.
- [28] N. Helwig, E. Pignatelli, and A. Schütze, "Condition monitoring of a complex hydraulic system using multivariate statistics," in *Instrumentation and Measurement Technology Conference (I2MTC)*, 2015 IEEE International, 2015, Pisa, Italy.
- [29] C. Lessmeier, J. K. Kimotho, D. Zimmer, and W. Sextro, "Condition Monitoring of Bearing Damage in Electromechanical Drive Systems by Using Motor Current Signals of Electric Motors: A Benchmark Data Set for Data-Driven Classification," *Proceedings of the European conference of the prognostics and health management society*, pp. 5–8, 2016.
- [30] Case Western Reserve University Bearing Data Center Website (<http://csegroups.case.edu/bearingdatacenter/home>).
- [31] H. Huang and N. Baddour, "Bearing Vibration Data under Time-varying Rotational Speed Conditions," *Data in Brief*, vol. 2018, pp. 1745–1749, 21.
- [32] O. Mey, W. Neudeck, A. Scheider, and O. Enge-Rosenblatt, "Vibration Measurements on a Rotating Shaft at Different Unbalance Strengths", *Fraunhofer Fordatis*, <https://fordatis.fraunhofer.de/handle/fordatis/151.2>
- [33] <https://github.com/deepinsights-analytica/ieee-etfa2020-paper>
- [34] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias-variance trade-off," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116, no. 32, pp. 15849–15854, 2019.
- [35] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)," *Neurocomputing*, vol. 307, pp. 72–77, 2018.
- [36] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, 1986.
- [37] K.-F. Lee, H.-W. Hon, and R. Reddy, "An overview of the SPHINX speech recognition system," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1990.
- [38] W. Walker et al., "Sphinx-4: A flexible open source framework for speech recognition," *Tech. Rep. SMLI TR-2004-139*, 2004.
- [39] S. R. Eddy, "Hidden markov models," *Current Opinion in Structural Biology*, 1996.
- [40] P. Hausberger, A. Fernbach, and W. Kastner, "IMU-based smart fitness devices for weight training," in *42nd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2016.
- [41] T. Marwala and C. B. Vilakazi, "Computational intelligence for condition monitoring," 2007.