Acoustic Resonance Testing of Small Data on Sintered Cogwheels

Yong Chul Ju 1,* , Ivan Kraljevski 1,1, Heiko Neunübel 2, Constanze Tschöpe 3 and Matthias Wolff 4

1 Cognitive Material Diagnostics Project Group of Fraunhofer Institute for Ceramic Technologies and Systems IKTS, 03046 Cottbus, Germany
2 Condition Monitoring Hardware and Software Group of Fraunhofer Institute for Ceramic Technologies and Systems IKTS, 01109 Dresden, Germany
3 Machine Learning and Data Analysis Group of Fraunhofer Institute for Ceramic Technologies and Systems IKTS, 01109 Dresden, Germany
4 Chair of Communications Engineering, Brandenburg University of Technology Cottbus-Senftenberg, 03046 Cottbus, Germany
* Correspondence: yong.chul.ju@ikts.fraunhofer.de

Abstract: Based on the fact that cogwheels are indispensable parts in manufacturing, we present the acoustic resonance testing (ART) of small data on sintered cogwheels for quality control in the context of non-destructive testing (NDT). Considering the lack of extensive studies on cogwheel data by means of ART in combination with machine learning (ML), we utilize time-frequency domain feature analysis and apply ML algorithms to the obtained feature sets in order to detect damaged samples in two ways: one-class and binary classification. In each case, despite small data, our approach delivers robust performance: All damaged test samples reflecting real-world scenarios are recognized in two one-class classifiers (also called detectors), and one intact test sample is misclassified in binary ones. This shows the usefulness of ML and time-frequency domain feature analysis in ART on a sintered cogwheel dataset.

Keywords: acoustic resonance testing (ART); deep learning; machine learning; small data; non-destructive testing (NDT)

1. Introduction

Since the industrial age cogwheels (the term cogwheels may be considered as gears but not gearboxes or bearing systems) have been indispensable components in manufacturing, e.g., the textile and automotive industries, and they still play a significant role even in this information age, e.g., robotics and aerospace. This makes developing reliable and cost-effective non-destructive testing (NDT) methods an integral part of quality control (QC).

The field involved with cogwheels is vast, and yet most work in the literature has been performed in the context of gearboxes or bearing systems [1–7], i.e., many gears are inside in a system and attached to each other. For such systems, the main focus of fault detection lies on a system failure dealing with conditions monitoring in lifespan analysis, which occurs mainly due to malfunctioning components suffering from wear, abrasion and pollution, such as sand or lubrication.

However, this necessarily leads to different directions of research, when the structural health diagnosis of cogwheels in manufacturing process, e.g., sintering, comes into focus. Moreover, this often encounters small data problems simply due to lacking data of defective parts; see, e.g., [4]. Here, small data problems refer specifically to the situation when there are not enough data available for training algorithms in machine learning (ML), which poses difficulties in various fields as, e.g., can be seen in [8].

Although gearboxes or bearing systems related work have made progress [5–7], usually by means of modern deep learning (DL) [9], the employed methods are not always
applicable to small data problems as discussed in [10,11], e.g., 22 layers of GoogLeNet [12] used in [6].

Moreover, not all studies reflect real-world scenarios, e.g., the tooth of a gear is intentionally cut off [5]. In addition, many studies do not mention the sample size of a dataset nor countermeasures against overfitting whether the proposed work is suitable for small data problems, e.g., [7]. Additionally, ML methods employed in some work dealing with small data are still limited to shallow learning [3], i.e., traditional ML [10]. In order to show missing points in the field and different aspects of this study, we provide an overview of signal-based methods in Table 1.

Table 1. Comparison of gearboxes, bearings, and cogwheels related work, where NM stands for “not mentioned. The symbols “✓” and “–” denote “being affirmative” and “not applicable”, respectively.

|                  | Qu et al. [1] | Haidong et al. [2] | Oh et al. [3] | Saufi et al. [4] | Usman et al. [5] | König et al. [6] | Žvirblis et al. [7] | This Work |
|------------------|--------------|-------------------|--------------|------------------|------------------|------------------|------------------|-----------|
| Gearboxes or Bearings | ✓            | ✓                 | ✓            | ✓                | ✓                | ✓                | ✓                | ✓         |
| Cogwheels        | –            | –                 | –            | –                | –                | –                | –                | –         |
| Shallow Learning  | NM           | ✓                 | ✓            | ✓                | ✓                | NM               | NM               | ✓         |
| Deep Learning    | NM           | ✓                 | ✓            | ✓                | ✓                | –                | ✓                | ✓         |
| Sample Size      | NM           | NM                | 300          | 750              | 5000             | NM               | NM               | 180–232   |
| Small Data       | –            | –                 | ✓            | ✓                | –                | –                | –                | ✓         |

When it comes to NDT, other than signal-based approaches, there also exist image-based methods and they have made considerable progress since modern DL-based algorithms have become part of the mainstream across most research disciplines [13,14] due mainly to the work by Krizhevsky et al. [15]. However, in this study, we focus solely on signal-based methods on the grounds that image-based approaches are not as cost-effective as signal-based ones [16], and the methods become futile when defects in images are invisible as in our case. In addition, among different ML algorithms, we call them modern when the employed approaches are involved with DL-based ones; otherwise, we call them classical.

Hence, given a summary about the matter in Table 1, apart from gearboxes or bearing systems, to the best of our knowledge, there has actually been no extensive study on sintered cogwheel small data using acoustic resonance testing (ART) [17] with the help of traditional and modern ML methods in the context of non-destructive testing (NDT).

**Our Contributions:** In this work, we address the aforementioned issues and intend to bridge the gap: We collect a small dataset on cogwheels and perform time-frequency domain feature analysis.

Afterwards, we apply not only classical ML algorithms but also modern DL-based ones to the obtained feature sets in the way of one-class as well as binary classification. In this way, in spite of having small data, our approach is able to achieve robust performance: All defective test samples reflecting real-world scenarios are recognized in two one-class classifiers (also called detectors) and one intact test sample is misclassified in binary classification. This suggests that ART can be an attractive tool on cogwheel data in QC when taking advantage of the combination of ML algorithms and time-frequency domain feature analysis.

**Paper Organization:** The paper is organized as follows: After we give a brief exposition on data acquisition and feature analysis in Section 2, we provide information on training of ML algorithms in Section 3. Then, we present the result of experiments in Section 4. Finally, the paper is closed with our concluding remarks.
2. Materials Furthermore, Methods

2.1. Data Acquisition, Measurement and Sensors

2.1.1. Test Objects

In the experiment, five cogwheels (chain wheels) are examined. They are made of sintered iron and inductively hardened in surface layers. The weight of the cogwheels is approximately 140 g, the outer diameter is 79 mm, and the thickness amounts to 7–9 mm.

2.1.2. Examination Setup of Objects

The testing station for cogwheels, including a lifting device, was developed in Fraunhofer IKTS in Dresden, Germany. It is equipped with a three-point mounting system and is pneumatically controlled. In order to guarantee repeated and reproducible placement, the cogwheel is fixed when it is placed. The cogwheel is raised up to three tip points with compressed air and thus distanced from the test bench. Moreover, in these three tip points, one transmitter and two receivers (channel 1 and channel 2) are mounted, see Figure 1a.

2.1.3. Measurement Method of Signal

For the measurement of signals, a multi-channel acoustic measurement system (MAS) was used: four channels, analog input amplifiers and digitization of the measurement signals, output amplifying stage for exciting acoustic converters, CAN interface to PC. In addition, two preamplifiers (40 dB; 10–500 kHz), one ultrasonic piezo actuator (transmitter) and two ultrasonic piezo sensors (receiver) are used. Each actuator and sensor is with a hard metal tip. The operating software for MAS has the following functionalities: configuring measurement channels, generating and sending excitation functions, as well as recording and storing measured signals in a time-synchronous way.

2.1.4. Measurement on Cogwheels

For collecting data, the aforementioned five sintered cogwheels are used. Four of them are in intact condition, and one has defects. Concerning the defects, they were introduced by a company specialized in this area. These are designed in such a way that real-world scenarios are reflected and thereby almost indistinguishable from real ones.

![Figure 1. Acoustic measurement configuration of the cogwheel. (a) Test station for the acoustic measurement of the cogwheel; (b) Marked positions in the cogwheel.](image-url)
For more details, we refer to [18–20] and the references therein. For each gear wheel, the raw signal of acoustic response that goes through a preamplifier was recorded by two receivers (channels 1 and 2) with a sampling rate of 1041.67 kHz with respect to ten different positions:

Although the receivers are mounted in fixed positions, the measurements of structural vibrations are actually obtained in different positions by rotating the wheel, which makes the data acquisition process less biased in terms of the positions of receivers. The reference point for positioning the gear is rotated in a counterclockwise direction every four teeth of the gear wheel and marked from P00 to P09, see Figure 1b. Moreover, each observation is labeled as either “OK” for intact samples or “UNK” for defective ones, respectively. The dataset is organized with respect to three excitation signals:

1. Chirp signal ranging from 1 kHz to 200 kHz (Crp1k-200k).
2. RC2 impulse with 75 kHz (RC2-75k), where RC2 is defined by \( RC2 = 0.5 \left( 1 - \cos(\pi f t) \right) \cos(2\pi ft) \).
3. Sinc function with 150 kHz (Sinc-150k).

As described in Tables 2 and 3, there are 160 intact samples of recording and 20 defective ones for Crp1k-200k and RC2-75k and 212 intact and 20 damaged for Sinc-150k. The dimension of each observation amounts to 104,674, see Figure 2a,d.

---

**Figure 2.** Signals of an intact and a defective sample and corresponding spectrograms of PFA and SFA for Crp1k-200k. For comparison between intact and defective samples, the most noticeable regions are marked with red boxes: (b) vs. (e) (second column) and (c) vs. (f) (third column). (a) Signal with an intact sample; (b) PFA of (a); (c) SFA of (b); (d) Signal with a defective sample; (e) PFA of (d); (f) SFA of (e).
Concerning sensor fusion methods, the late fusion approach is adopted in a sense that pseudo probability scores that are obtained from trained models using each channel are averaged to make a final prediction by incorporating the threshold of equal error rate (EER), see Figure 3. We provide more details on how the aforementioned pseudo probability scores are obtained depending on the deployed ML algorithms in Section 4.1.

Figure 3. A schematic view of workflow in our approach. Channel 1 and channel 2 are abbreviated as ch1 and ch2, respectively. \( P_1 \) and \( P_2 \) denote pseudo probability scores obtained from trained models with channel 1 and channel 2, respectively.

### 2.2. Feature Analysis

#### 2.2.1. Primary Feature Analysis (PFA)

To perform the PFA, a short time Fourier transform (STFT) is first computed, and the resulting frequency-time dependent features are presented in the form of a spectrogram. The STFT is performed on a signal frame of Blackman analysis window with a length of 512 signal samples and MEL filter bank with a triangular function. The frame shift is 160 samples, which yields PFA features with the dimensions of \( 652 \times 30 \); see Figure 2b,e.

#### 2.2.2. Secondary Feature Analysis (SFA)

SFA is performed based on the PFA. First, the features are rescaled to have a mean of 0 and a standard deviation of 1. Afterwards, delta features are computed by subtracting consecutive frames and principal component analysis (PCA) is performed for feature dimensionality reduction. This leads to feature vectors with a dimension of \( 652 \times 24 \), see Figure 2c,f.

### 3. Training of Classifiers

Given the dataset, the main goal of our experiment is to investigate which combination of ML methods and feature sets are appropriate for recognizing real-world defects. To this end, we first considered one-class-based methods as applied in anomaly detection in order to deal with the limitations in sample size and imbalance of the acquired dataset:

#### Table 2. Sample size of Crp1k-200k and RC2-75k, where Z01–Z05 refers to the number of cogwheels.

|       | Z01 | Z02 | Z03 | Z04 | Z05 | Total |
|-------|-----|-----|-----|-----|-----|-------|
| intact (OK) | 100 | 20  | 20  | 20  |     | 160   |
| defective (UNK) |     | 20  | 20  |     |     | 20    |

#### Table 3. Sample size of Sinc-150k, where Z01–Z05 refers to the number of cogwheels.

|       | Z01 | Z02 | Z03 | Z04 | Z05 | Total |
|-------|-----|-----|-----|-----|-----|-------|
| intact (OK) | 152 | 20  | 20  | 20  |     | 212   |
| defective (UNK) |     | 20  | 20  |     |     | 20    |
• hidden Markov model (HMM),
• support-vector machine (SVM),
• isolation forest (IF), and
• autoencoder of bottleneck type (AE-BN).

Moreover, we also applied the following methods in the way of binary classification:
• feed-forward neural networks (FFNNs), and
• convolutional neural networks (CNNs).

Although NN-based methods, such as CNNs, are well-known to be useful for constructing feature maps from raw signals [9], this comes at the expensive price of a large dataset for training [21]. Moreover, this is often not a viable option as in our situation.

On this account, we restrict ourselves to PFA and SFA feature sets for training.

3.1. Configuration of Experiments

The dataset is prepared in a way that there is no overlap between training and test sets. Stratified five-fold cross validation (CV) is employed during all experiments to ensure good representation of the whole classes in the training and test folds. For one-class classification, this strategy is realized in such a way that training is performed only on intact samples without a designated fold and tested against all damaged ones with the reserved fold as illustrated in Figure 4. The reasoning behind this is to circumvent overfitting as much as possible by exploiting the common properties of small dataset, i.e., few damaged samples compared to intact ones.

**Figure 4.** An illustration of stratified five-fold CV in terms of fold 2.

3.1.1. Hidden Markov Models

HMMs can be viewed as an extension of a mixture model, where the choice of the mixture component for each observation is not selected independently but depends on the choice of component for the previous observation. This is called the Markov property [22]. Since HMMs are useful for dealing with sequential data, they are widely used in speech recognition [23] and natural language processing [24].

However, they have also been successfully applied in advanced NDT [25]. Although long short-term memory (LSTM) is known to be good at dealing with variable length of sequential data [26], we instead make use of a simpler model HMM considering that our feature sets PFA and SFA have fixed dimensions. Our HMM is designed in such a way that ten hidden states release observations that correspond to our acquired dataset via one Gaussian probability density function with a full covariance matrix in each state. To detect anomalies, we used the interquartile range by measuring a score characterizing how well
our model describes an observation point. The experiments are conducted by means of
the dLabPro package [27], and the model parameters are estimated by the Baum–Welch
algorithm [28].

3.1.2. Support-Vector Machines

The SVM is a generalization of the maximal margin classifier, and it classifies data
points by constructing a separating hyperplane that distinguishes one class from others [29].
SVMs are extremely powerful ML algorithms to solve various classification problems in that
not only are they less prone to overfitting due to large margins but they are also relatively
manageable to solve due to convex nature. Moreover, it is also well-known that they are
effective in dealing with high dimensions of features—particularly when the number of
features are much more than training samples—by making use of kernel tricks regarding
nonlinear classification problems.

Our experiments were implemented using the scikit-learn [30] interface relying on
the LIBSVM library [31]. SVM models were trained using the radial basis function (RBF)
kernel, and the following parameters were tuned on about 20% of the training set to obtain
optimal results: (1) regularization parameter C (from \(10^{-5}\) to \(10^7\)), and (2) \(\gamma\), which defines
how far a single sample influences (from \(10^{-10}\) to \(10^{-1}\)), or (3) \(\nu\), which has the ability to
to control over the number of support vectors (from \(10^{-3}\) to 1), if necessary.

3.1.3. Isolation Forest

Isolation forest belongs to the family of ensemble methods and is a tree-based anomaly
detection algorithm that isolates observations as outliers based on the anomaly score
delivered by profiling a randomly selected feature with a random split value between
minimum and maximum values of the selected feature [32,33]. This has been a useful
 technique in wide range of fields, e.g., finding anomalies in hyperspectral remote sensing
images [34], detecting anomalous taxi trajectories from GPS traces [35], or in analyzing
partial discharge signals of a power equipment [36].

Our experiments are realized by scikit-learn [30]: the minimum split number is set to
2, and the maximum depth of each tree is defined by \(\lceil \log_2 n \rceil\), where \(n\) denotes the number
of samples used to build the tree.

3.1.4. Autoencoder of Bottleneck Type

An autoencoder refers to a type of ANN which aims at approximating original input
signal in an unsupervised way [37], which is composed of two parts: encoding and decod-
ing layers. The encoding layers are responsible for finding an efficient representation of
the input vectors by learning useful features, and decoding layers attempt to reconstruct
the input signal as close as possible from the acquired encoded information. Since AEs are
capable of generating the compact representation of input data, which is extremely useful
in terms of feature learning, there is an enormous potential to solve various problems, such
as anomaly detection [38], image denoising [39] and shape recognition [40].

Our experiments were performed by leveraging Keras [41] with TensorFlow [42] and
the following feed-forward bottleneck type architecture is employed: input-512-64-512-
output. As shown in Figure 5, the input and output size are equal to the dimensions of the
vectorized feature sets, i.e., 19,560 for PFA and 15,648 for SFA, respectively.

All layers are fully connected and activated by leaky rectified linear unit (LReLU) to
overcome vanishing gradient [43]. In addition, to deal with internal covariant shift batch
normalization (BNorm) is applied to each layer [44].

Moreover, as countermeasures against overfitting, which, in our case, is of grave con-
ern particularly due to small data, random dropout with a 0.5 rate in internal layers [45] and
the early stopping strategy making use of the patience parameter with 25 are considered [46],
where the patience specifies the number of epochs with no improvement in terms of the
used loss function, after which, training will be halted [41]. Given the maximum number
of epochs to be 500 in our experiments, the early stopping criterion comes into play in a
range from epochs 132 to 445 depending on the folds in the datasets.

![Figure 5. The architecture of the autoencoder of bottleneck type, where $n$ denotes the dimensions: 19,560 for PFA and 15,648 for SFA.](image)

Our AE-BNs have about 20 million parameters, and for training, adaptive moment estimation (Adam) optimization [47] is incorporated along with $L_1$ regularization to obtain sparse solutions. Hyperparameter optimization using grid search is conducted on about 20% of training set in pre-training stages to obtain suitable parameter values, such as training batch size 512 and the aforementioned dropout rate 0.5.

3.1.5. Deep Learning for Binary Classification

DL may be defined as a class of ML algorithms that typically make use of multilayer NNs in order to progressively extract different levels of representations of input data, which correspond to a hierarchy of features [48]. While the input data are being processed in multiple layers, each layer allows to reveal additional features of the input in such a way that higher level features are described in terms of lower level ones to help understand the data. As in [49], this can be understood in the following example from image classification:

Given an image of a dog as input, for instance, pixel values are detected in the first layer; edges are identified in the second layer; combinations of edges and other complex features based on the edges from the previous layer are identified in next several layers; and finally the input image is recognized as a dog in output.

Apart from the different levels of abstraction, due to the capability of nonlinear information processing, DL-based approaches have recently become popular in many fields, including, but not limited to, image processing, computer vision, speech recognition, and natural language processing [50]. As in the case of AE-BN, our DL routines were also realized by Keras [41] using TensorFlow [42], and the following architectures were employed: Three hidden layers are stacked and fully connected, see Figure 6. These hidden layers are incorporated with 600, 300 and 100 nodes and activated by the LReLU function. In addition, BNorm and a dropout rate of 0.5 are employed in each layer. Other configurations are similar to those of AE-BE: The Adam optimizer along with $L_1$ regularization, early stopping by means of the patience parameter with 25, batch size with 256 and maximum number of epochs with 200 are considered. From one node in the output layer, binary
classification is realized using binary cross-entropy loss by mapping “UNK” to 0 and “OK" to 1. Our FFNN has about 12 million parameters.

![Figure 6](image_url)

**Figure 6.** The architecture of SNN, where $n$ denotes the dimensions: 19,560 for PFA and 15,648 for SFA.

In the case of CNN, three 2-D convolution layers with the kernel size of 3 $\times$ 3 are employed, each of which has 16, 32 and 64 feature maps and is downsampled with the stride of 2 $\times$ 2. Then, the LReLU activation function, BNorm and dropout rate with 0.75 and a 2-D max pooling layer with 2 $\times$ 2, which is another way to deal with overfitting, are applied to each layer. Then, the result is flattened and fed into a fully connected layer with 50 nodes activated by LReLU, where BNorm and the dropout rate 0.75 are also used.

As can be noticed, a relative high dropout rate is chosen for reducing model complexity in the light of overfitting owing to the small size of defective samples. Compared to the case of FFNN, other configurations for training remain unchanged except the maximum number of epochs at 300. The architecture of CNN is provided in Table 4. Binary classification is implemented in the same way as in the case of FFNN. Our CNN has approximately sixty thousand parameters.
Table 4. The architecture of CNN. The reported dimensions are for PFA.

| No. | Layer Type     | Filter Size | Kernel Size | Stride | Input Size         | Output Size         |
|-----|----------------|-------------|-------------|--------|--------------------|---------------------|
| 1   | Conv2D         | 16          | 3 × 3       | 2 × 2  | 652 × 30 × 1      | 326 × 15 × 16       |
| 2   | MaxPooling     | 2           | 2 × 2       | 2 × 2  | 326 × 15 × 16     | 163 × 8 × 16        |
| 3   | Conv2D         | 32          | 3 × 3       | 2 × 2  | 163 × 8 × 16      | 82 × 4 × 32         |
| 4   | MaxPooling     | 2           | 2 × 2       | 2 × 2  | 82 × 4 × 32       | 41 × 2 × 32         |
| 5   | Conv2D         | 64          | 3 × 3       | 2 × 2  | 41 × 2 × 32       | 21 × 1 × 64         |
| 6   | MaxPooling     | 2           | 2 × 2       | 2 × 2  | 21 × 1 × 64       | 11 × 1 × 64         |
| 7   | Flatten        |             |             |        | 11 × 1 × 64       | 704                 |
| 8   | Fully-connected|             |             |        | 704               | 50                  |
| 9   | Fully-connected|             |             |        | 50                | 1                   |

4. Results and Discussion

4.1. Evaluation Metrics

In order to evaluate different classification algorithms, we provide the following performance metrics: balanced accuracy rate (BAR) along with corresponding 95% confidence interval (CI) [51], area under curve (AUC), Matthews correlation coefficient (MCC) [52] and the histogram of scores computed by one-class classifiers along with a classification margin (CM) if classes are clearly separable, i.e., if EER equals to 0.

Since scores are close to 0 and 1 for defective and intact classes, respectively, CM is defined by

$$CM = \frac{\min(S_{\text{OK}}) - \max(S_{\text{UNK}})}{\max(S_{\text{OK}}) - \min(S_{\text{UNK}})} \times 100\%$$

where $S_{\text{OK}}$ and $S_{\text{UNK}}$ denote the scores of the classes “UNK” and “OK” and $\max(\cdot)$ and $\min(\cdot)$ stand for the maximum and minimum of the scores of the designated class, respectively.

This measure represents a ratio of a maximum margin of scores between classes to the whole spectrum of scores from both classes, where a maximum margin of scores can be computed by subtracting the maximum score of the defective class from the minimum score of the intact class.

To make an inference of a class $c \in \{\text{OK, UNK}\}$ for a test set, the aforementioned scores for each detector are defined and computed based on [53] in the following way:

- HMM:

  $$S_c := P(c|x) = 10^{-\frac{\text{NLL}(x)}{\gamma}} \quad \text{for } x \in D_{\text{test}},$$

  where $D_{\text{test}}$ denotes a test set, NLL stands for the negative log likelihood, and $\gamma = \frac{1}{|D_{\text{test}}|} \sum_{x \in D_{\text{test}}} \text{NLL}(x)$ with $|\cdot|$ as the cardinality of a set.

- SVM:

  $$S_c := P(c|x) = \frac{1}{1 + \text{e}^{-\text{score}(x)}} \quad \text{for } x \in D_{\text{test}},$$

  where $\text{score}(x)$ denotes the distance from $x$ to the separating hyperplane.

- IF:

  $$S_c := P(c|x) = \frac{1}{1 + \text{e}^{-\text{score}(x)}} \quad \text{for } x \in D_{\text{test}},$$

  where $\text{score}(x)$ is defined as in [33].

- AE-BN:

  $$S_c := P(c|x) = \frac{1}{1 + \text{e}^{-\text{score}(x)}} \quad \text{for } x \in D_{\text{test}},$$

  where $\text{score}(x)$ denotes the mean squared error (MSE) of the cross entropy loss function.
4.2. One-Class Classification

When it comes to one-class classification, despite small and imbalanced data, SVM and AE-BN perform equally well in terms of BAR across all feature sets and all three excitation functions, see Table 5. Please note that the given BAR and CI are based on beta distribution, which necessarily leads to asymmetric CIs and slightly lower values of BAR than the conventional accuracy although all test samples are correctly classified in the case of SVM and AE-BN.

In contrast to SVM and AE-BN, HMM has some difficulties with SFA in all three datasets. Moreover, when PFA is combined with either HMM or IF in the Sinc-150k dataset two misclassifications occur: intact samples are recognized as damaged ones, which is less severe than the opposite situation in a production line. The result of BAR is consistent in terms of MCC, see Tables 5 and 6. However, AUC scores tend to be higher particularly for the case of binary classification in spite of the occurrence of one misclassification, see Tables 5 and 7.

As shown in Figures 7–9, the juxtaposed histograms of scores with the help of a CM allow us to further investigate how well classifiers behave with respect to feature sets, thresholds and robustness. The CM is available as long as classes are not overlapped.

From Figures 7d, 8d and 9d, one can notice that among all combinations between classifiers and feature types SVM with SFA delivers a best performance in terms of CM, which is followed by SVM with PFA, AE-BN with SFA and AE-BN with SFA in each dataset. It can be also recognized that IF gives better performance with SFA than with PFA in all databases, which, however, is not the case with HMM or AE-BN. From the perspective of excitation functions, more classifiers are able to recognize all test sets correctly in the dataset Crp1k-200k and RC2-75k than in Sinc-150k.

The result of our approach suggests that one-class classification still allows for reliable anomaly detection even though training is performed only on intact samples. Moreover, our proposed method gives robust performance by showing fairly large CM not only with classical methods but also with modern DL-based ones, e.g., 46% of SVM with SFA and 40% of AE-BN with SFA as shown in Figures 7d and 9h. This makes an important point of our contribution since real-world scenarios of data skewness in a production line, i.e., numerous intact samples but few damaged ones, are considered.

4.3. Binary Classification

While one-class-based experiments show different results depending on combinations of classifiers and feature sets in each dataset, binary classification experiments yield one misclassification in all cases: an intact sample is misclassified as the damaged one, see Table 5. It should be noted that binary classifications, in contrast to the one-class case, make use of not only intact samples but also defective ones for training.

Since the number of flawed samples are much less than that of flawless ones, obtained models from training are prone to overfitting, which forces us to take various countermeasures, such as less complex NN architectures, high dropout rates and a higher weight of regularization. Although FFNN and CNN deliver solid performance in our case, it should be noted that it may sometimes be difficult to deal with small data.

To improve the overall performance of binary classification, it is therefore desirable to provide more data of faulty samples. In this context, data augmentation by considering the physical properties of cogwheels, e.g., numerical simulation, may be a possible approach to deal with the difficulties.
Table 5. BAR (in percent) along with 95% CI of classifiers depending on feature sets and datasets. For each case of classifiers, the best performing ones with respect to excitation signals and used features are denoted in boldface.

|                | Crp1k-200k | RC2-75k | Sinc-150k |
|----------------|------------|---------|-----------|
|                | PFA        | SFA     | PFA       | SFA     | PFA       | SFA     |
| Detectors      |            |         |           |         |           |         |
| HMM            | 99.25 ± 0.73 | 98.09 ± 1.28 | 99.25 ± 0.73 | 91.17 ± 2.73 | 98.83 ± 0.89 | 86.36 ± 3.76 |
| SVM            | 99.25 ± 0.73 | 99.25 ± 0.73 | 99.25 ± 0.73 | 99.25 ± 0.73 | 99.25 ± 0.73 | 99.25 ± 0.73 |
| FFNN           | 97.13 ± 0.49 | 97.13 ± 0.49 | 97.13 ± 0.49 | 97.13 ± 0.49 | 97.13 ± 0.49 | 97.13 ± 0.49 |
| CNN            | 97.13 ± 0.49 | 97.13 ± 0.49 | 97.13 ± 0.49 | 97.13 ± 0.49 | 97.13 ± 0.49 | 97.13 ± 0.49 |

Table 6. Matthews correlation coefficient (MCC) of classifiers depending on feature sets and datasets.

|                | Crp1k-200k | RC2-75k | Sinc-150k |
|----------------|------------|---------|-----------|
|                | PFA        | SFA     | PFA       | SFA     |
| Detectors      |            |         |           |         |
| HMM            | 1.0000     | 0.9757  | 1.0000    | 0.8714  |
| SVM            | 1.0000     | 1.0000  | 1.0000    | 1.0000  |
| IF             | 1.0000     | 1.0000  | 1.0000    | 1.0000  |
| AE-BN          | 1.0000     | 1.0000  | 1.0000    | 1.0000  |
| Binary Classifiers |     |         |           |         |
| FFNN           | 0.9728     | 0.9728  | 0.9728    | 0.9728  |
| CNN            | 0.9728     | 0.9728  | 0.9728    | 0.9728  |

Table 7. Area under curve (AUC) of classifiers depending on feature sets and datasets.

|                | Crp1k-200k | RC2-75k | Sinc-150k |
|----------------|------------|---------|-----------|
|                | PFA        | SFA     | PFA       | SFA     |
| Detectors      |            |         |           |         |
| HMM            | 1.0000     | 0.9994  | 1.0000    | 0.9885  |
| SVM            | 1.0000     | 1.0000  | 1.0000    | 1.0000  |
| IF             | 1.0000     | 1.0000  | 1.0000    | 1.0000  |
| AE-BN          | 1.0000     | 1.0000  | 1.0000    | 1.0000  |
| Binary Classifiers |     |         |           |         |
| FFNN           | 1.0000     | 1.0000  | 1.0000    | 1.0000  |
| CNN            | 1.0000     | 1.0000  | 1.0000    | 1.0000  |
Figure 7. Cont.
Figure 7. Histogram of one-class classifiers with the dataset Crp1k-200k. CM in Equation (1) is reported if available. Values on the x-axis are normalized classifier scores according to Equations (2)–(5).

(a) HMM with PFA; (b) HMM with SFA; (c) SVM with PFA; (d) SVM with SFA; (e) IF with PFA; (f) IF with SFA. (g) AE-BN with PFA; (h) AE-BN with SFA.

Figure 8. Cont.
Figure 8. Histogram of one-class classifiers with the dataset RC2-75k. CM in Equation (1) is reported if available. Values on the x-axis are normalized classifier scores according to Equations (2)–(5). (a) HMM with PFA; (b) HMM with SFA; (c) SVM with PFA; (d) SVM with SFA; (e) IF with PFA; (f) IF with SFA; (g) AE-BN with PFA; (h) AE-BN with SFA.

Figure 9. Cont.
Figure 9. Histogram of one-class classifiers with the dataset Sinc-150k. CM in Equation (1) is reported if available. Values on the x-axis are normalized classifier scores according to Equations (2)–(5). (a) HMM with PFA; (b) HMM with SFA; (c) SVM with PFA; (d) SVM with SFA; (e) IF with PFA; (f) IF with SFA; (g) AE-BN with PFA; (h) AE-BN with SFA.
5. Conclusions

In this article, we presented the ART approach on small data of sintered cogwheels by utilizing not only classical ML algorithms but also modern ones. In consideration of data imbalances, our experiments were performed in two ways: one-class classification and binary classification. Our experimental results with a large safety margin classification demonstrated that one-class classifiers (detectors) had considerable potential to serve as an effective and thereby attractive tool in a reliable anomaly detection system in NDT. In addition, the experiments of binary classifiers support that they were still able to deliver robust performance in spite of small data. This shows the usefulness of ML along with time-frequency domain feature analysis on the cogwheel dataset in ART for QC.

Author Contributions: Conceptualization, C.T. and M.W.; methodology, Y.C.J.; software, Y.C.J. and I.K.; validation, Y.C.J.; formal analysis, Y.C.J.; investigation, Y.C.J.; resources, C.T. and M.W.; data curation, H.N.; writing—original draft preparation, Y.C.J. and H.N.; writing—review and editing, Y.C.J., I.K., H.N., C.T. and M.W.; visualization, Y.C.J.; supervision, C.T. and M.W.; project administration, C.T. and M.W.; funding acquisition, C.T. and M.W. All authors have read and agreed to the published version of the manuscript.

Funding: Parts of the study were supported by the Brandenburg Ministry of Science, Research and Cultural Affairs (project “Kognitive Materialdiagnostik”, grant #22-F241-03-FhG/007/001).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References
1. Qu, Y.; He, D.; Yoon, J.; Van Hecke, B.; Bechhoefer, E.; Zhu, J. Gearbox Tooth Cut Fault Diagnostics Using Acoustic Emission and Vibration Sensors—A Comparative Study. Sensors 2014, 14, 1372–1393. [CrossRef] [PubMed]
2. Haidong, S.; Hongkai, J.; Xingqiu, L.; Shuaipeng, W. Intelligent fault diagnosis of rolling bearing using deep wavelet auto-encoder with extreme learning machine. Knowl.-Based Syst. 2018, 140, 1–14. [CrossRef]
3. Oh, S.W.; Lee, C.; You, W. Gear Reducer Fault Diagnosis Using Learning Model for Spectral Density of Acoustic Signal. In Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 16–18 October 2019; pp. 1027–1029. [CrossRef]
4. Saufi, S.R.; Ahmad, Z.A.B.; Leong, M.S.; Lim, M.H. Gearbox Fault Diagnosis Using a Deep Learning Model With Limited Data Sample. IEEE Trans. Ind. Inform. 2020, 16, 6263–6271. [CrossRef]
5. Usman, M.; Anwar, S.; Akmal, M.; Hafeez, A. AI Detect: A Machine Learning Based Approach for Fault Identification in Gear Bearing System using Low-Frequency Data. In Proceedings of the 2020 14th International Conference on Open Source Systems and Technologies (ICOST), Lahore, Pakistan, 16–17 December 2020; pp. 1–6. [CrossRef]
6. König, F.; Sous, C.; Ouald Chaib, A.; Jacobs, G. Machine learning based anomaly detection and classification of acoustic emission events for wear monitoring in sliding bearing systems. Tribol. Int. 2021, 155, 106811. [CrossRef]
7. Žvirblis, T.; Petkevičius, L.; Vaitkus, P.; Šabanovič, E.; Skrickij, V.; Kilikevičius, A. Investigation of Deep Neural Networks for Hypoid Gear Signal Classification to Identify Anomalies. In Proceedings of the 2020 IEEE eighth Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), Vilnius, Lithuania, 22–24 April 2021, pp. 1–6. [CrossRef]
8. Zhang, Y.; Ling, C. A strategy to apply machine learning to small datasets in materials science. npj Comput. Mater. 2018, 4. [CrossRef]
9. Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning; MIT Press: Cambridge, MA, USA, 2016. Available online: http://www.deeplearningbook.org (accessed on 27 June 2022).
10. Saufi, S.R.; Ahmad, Z.A.B.; Leong, M.S.; Lim, M.H. Challenges and Opportunities of Deep Learning Models for Machinery Fault Detection and Diagnosis: A Review. IEEE Access 2019, 7, 122644–122662. [CrossRef]
11. Brigato, L.; Iocchi, L. A Close Look at Deep Learning with Small Data. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 2490–2497. [CrossRef]
12. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [CrossRef]
13. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning, Nature 2015, 521, 436–444. [CrossRef]
14. Schmidhuber, J. Deep learning in neural networks: An overview. Neural Netw. 2015, 61, 85–117. [CrossRef] [PubMed]
15. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*; Pereira, F., Burges, C., Bottou, L., Weinberger, K., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2012; Volume 25, pp. 1097–1105.

16. Kraljevski, I.; Duckhorn, F.; Jü, Y.C.; Tschöpe, C.; Richter, C.; Wolff, M. Acoustic Resonance Recognition of Coins. In Proceedings of the 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Dubrovnik, Croatia, 25–28 May 2020; pp. 1–6. [CrossRef]

17. Coffey, E. Acoustic resonance testing. In Proceedings of the 2012 Future of Instrumentation International Workshop (FIWI), Gatlinburg, TN, USA, 8–9 October 2012; pp. 1–2. [CrossRef]

18. Kemppainen, M.; Virkkunen, I. Crack characteristics and their importance to NDE. *J. Nondestruct. Eval.* 2011, 30, 143–157. [CrossRef]

19. Koskinen, A.; Leskelä, E. Differences in different indications of three artificially produced defects in ultrasonic inspection. In Proceedings of the BALTCIA IX—International Conference on Life Management and Maintenance for Power Plants, Helsinki, Finland, 11–13 June 2013; Volume 106, pp. 581–602.

20. Koskinen, T.; Virkkunen, I.; Siljama, O.; Jeness-Juhler, O. The Effect of Different Flaw Data to Machine Learning Powered Ultrasonic Inspection. *J. Nondestruct. Eval.* 2021, 40, 1–13. [CrossRef]

21. Feng, S.; Zhou, H.; Dong, H. Using deep neural network with small dataset to predict material defects. *Mater. Des.* 2019, 162, 300–310. [CrossRef]

22. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.

23. Jelinek, F. Statistical Methods for Speech Recognition; MIT Press: Cambridge, MA, USA, 1998.

24. Manning, C.; Schütze, H. *Foundations of Statistical Natural Language Processing*; MIT Press: Cambridge, MA, USA, 1999.

25. Tschöpe, C.; Wolff, M. Statistical Classifiers for Structural Health Monitoring. *IEEE Sens. J.* 2009, 9, 1567–1576. [CrossRef]

26. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* 1997, 9, 1735–1780. [CrossRef] [PubMed]

27. Wolff, M. dLabPro: A Signal Processing and Acoustic Pattern Recognition Toolbox. 2014. Available online: https://github.com/mattias-wolff/dLabPro (accessed on 27 June 2022).

28. Baum, L.E.; Petrie, T. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *Ann. Math. Stat.* 1966, 37, 1554–1563. [CrossRef]

29. Schölkopf, B.; Platt, J.C.; Shawe-Taylor, J.C.; Smola, A.J.; Williamson, R.C. Estimating the Support of a High-Dimensional Distribution. *Neural Comput.* 2001, 13, 1443–1471. [CrossRef]

30. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* 2011, 12, 2825–2830.

31. Chang, C.C.; Lin, C.J. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.* 2011, 2. [CrossRef]

32. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation Forest. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 413–422. [CrossRef]

33. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation-Based Anomaly Detection. *ACM Trans. Knowl. Discov. Data* 2012, 6, 1–39. [CrossRef]

34. Li, S.; Zhang, K.; Duan, P.; Kang, X. Hyperspectral Anomaly Detection With Kernel Isolation Forest. *IEEE Trans. Geosci. Remote Sens.* 2020, 58, 319–329. [CrossRef]

35. Zhang, D.; Li, N.; Zhou, Z.H.; Chen, C.; Sun, L.; Li, S. IBAT: Detecting Anomalous Taxi Trajectories from GPS Traces. In Proceedings of the 13th International Conference on Ubiquitous Computing, Association for Computing Machinery, Beijing, China, 17–21 September 2011; pp. 99–108. [CrossRef]

36. Wang, Y.B.; Chang, D.G.; Qin, S.R.; Fan, Y.H.; Mu, H.B.; Zhang, G.J. Separating Multi-Source Partial Discharge Signals Using Linear Prediction Analysis and Isolation Forest Algorithm. *IEEE Trans. Instrum. Meas.* 2020, 69, 2734–2742. [CrossRef]

37. Hinton, G.E.; Zemel, R. Autoencoders, Minimum Description Length and Helmholtz Free Energy. In *Advances in Neural Information Processing Systems*; Cowan, J., Tesauro, G., Alspector, J., Eds.; Morgan-Kaufmann: Burlington, MA, USA, 1993; Volume 6.

38. Sakurada, M.; Yairi, T. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In Proceedings of the MLSDA 2014—Second Workshop on Machine Learning for Sensory Data Analysis, Gold Coast, Australia, 2 December 2014; pp. 4–11. [CrossRef]

39. Gondara, L. Medical Image Denoising Using Convolutional Denoising Autoencoders. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona, Spain, 12–15 December 2016; pp. 241–246. [CrossRef]

40. Hinton, G.E.; Krizhevsky, A.; Wang, S.D. Transforming Auto-Encoders. In Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2011, Espoo, Finland, 14–17 June 2011; pp. 44–51. [CrossRef]

41. Chollet, F. Keras. 2015. Available online: https://github.com/fchollet/keras (accessed on 27 June 2022).

42. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. In Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016; pp. 265–283. [CrossRef]

43. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language Processing, Atlanta, GA, USA, 16–21 June 2013.

44. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
45. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.

46. Prechelt, L. Early Stopping—However, When? In Neural Networks: Tricks of the Trade, 2nd ed.; Montavon, G., Orr, G.B., Müller, K.R., Eds.; Springer: Berlin, Germany, 2012; pp. 53–67. [CrossRef]

47. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.

48. Deng, L.; Yu, D. Deep Learning: Methods and Applications. *Found. Trends Signal Process.* **2014**, *7*, 197–387. [CrossRef]

49. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014; pp. 818–833. [CrossRef]

50. Aggarwal, C.C. *Neural Networks and Deep Learning: A Textbook*; Springer: Berlin/Heidelberg, Germany, 2006.

51. Brodersen, K.H.; Ong, C.S.; Stephan, K.E.; Buhmann, J.M. The Balanced Accuracy and Its Posterior Distribution. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 3121–3124. [CrossRef]

52. Matthews, B. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta Protein Struct.* **1975**, *405*, 442–451. [CrossRef]

53. Kraljevski, I.; Duckhorn, F.; Tschöpe, C.; Wolff, M. Machine Learning for Anomaly Assessment in Sensor Networks for NDT in Aerospace. *IEEE Sens. J.* **2021**, *21*, 11000–11008. [CrossRef]