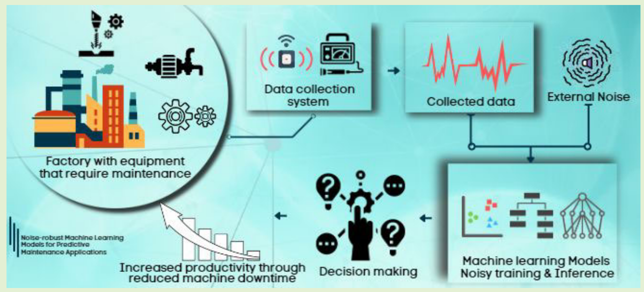
Noise-Robust Machine Learning Models for Predictive Maintenance Applications

Priscile Fogou Suawa, Anja Halbinger, Marcel Jongmanns, and Marc Reichenbach, *Member, IEEE*

***Abstract*—Predictive maintenance of equipment requires a set of data collected through sensors, from which models will learn behaviors that will allow the automatic detection or prediction of these behaviors. The objective is to anticipate unexpected situations such as sudden equipment stoppages. Industries are noisy environments due to production lines that involve a series of components. As a result, the data will always be obstructed by noise. Noise-robust predictive maintenance models, which include ensemble and deep learning models with and without data fusion, are proposed to enhance the monitoring of industrial equipment. The work reported in this article is based on two components, a milling tool, and a motor, with sound, vibration, and ultrasound data collected in real experiments. Four main tasks were performed, namely the construction of the datasets, the training of the monitoring models without adding artificial noise to the data, the evaluation of the robustness of the previously trained models by injecting several levels of noise into the test data, and the optimization of the models by a proposed noisy training approach. The results show that the models maintain their performances at over 95% accuracy despite adding noise in the test phase. These performances decrease by only 2% at a considerable noise level of 15-dB signal-to-noise ratio (SNR). The noisy training method proved to be an optimal solution for improving the noise robustness and accuracy of convolutional deep learning models, whose performance regression of 2% went from a noise level of 28 to 15 dB like the other models.**



***Index Terms*— Accelerometer, deep learning, ensemble learning, machine learning (ML), microphone, noise robustness, noisy training, predictive maintenance, ultrasound, white Gaussian noise.**

I. Introduction

Sensor data are often superimposed by noise. This can arise from the sensor itself or the electronic components, which process and amplify the signal. Additionally, this can also be background noise like humming from, e.g., ventilators or noise from other machines in the same room. When a machine learning (ML) model is trained to predict the state of a machine or tool, this noise must be considered. When the same machine is monitored in a different workshop, the background noise may give rise to wrong predictions of the ML model. Hence the importance of implementing noise-robust models. Although noise-resistant methods have been developed to improve the performance of various tasks, there are very few noise-robust models for predictive maintenance tasks. An approach based on noisy training incorporating a Gaussian layer with a decreasing noise level is proposed for the learning of such models. Unique datasets were employed in the evaluation of the proposed approaches, setting this work apart from existing research on noise-robust methods.

The work carried out in this article consisted of studying two pieces of equipment, a milling machine and a motor on which loads are mounted. Three types of data were collected during the operation of the equipment, namely sound, vibration, and ultrasound. These data were used to build models to monitor the condition of the equipment. Ensemble and deep learning methods were used to build these models. The main contributions are the following.

1) The construction of two new datasets, one composed of sound and vibration for milling machine monitoring and the other composed of ultrasound for motor monitoring.
2) The proposal of a noisy convolutional deep learning method based on the insertion of a noisy Gaussian layer with a descending noise level, which optimizes performance in terms of accuracy and robustness to noise.

Manuscript received 26 March 2023; accepted 26 April 2023. Date of publication 10 May 2023; date of current version 29 June 2023. This work was supported by the Federal Ministry of Education and Research of Germany (BMBF) within the iCampus Cottbus project, Grant Number 16ES1128K. The authors are responsible for the content of this publication. The associate editor coordinating the review of this article and approving it for publication was Dr. Lin Wang. *(Corresponding author: Priscile Fogou Suawa.)*

Priscile Fogou Suawa and Marc Reichenbach are with the Department of Computer Engineering, Brandenburg University of Technology Cottbus–Senftenberg, 03046 Cottbus, Germany (e-mail: suawapri@b-tu.de; marc.reichenbach@b-tu.de).

Anja Halbinger is with the Department of Mechanical Engineering, Friedrich-Alexander-University Erlangen-Nuremberg, 91054 Erlangen, Germany (e-mail: anja.halbinger@fau.de).

Marcel Jongmanns is with the Fraunhofer Institute for Photonic Microsystems, 01109 Dresden, Germany (e-mail: marcel.jongmanns@ipms.fraunhofer.de).

Digital Object Identifier 10.1109/JSEN.2023.3273458

3) The implementation of three noise-robust predictive maintenance models: a deep neural network (DNN), a deep convolutional neural network (DCNN), and an ensemble model based on a random forest classifier (RFC), a k-nearest neighbor classifier (KNN) and a support vector classifier (SVC).

The results obtained show that the various implemented models achieve an accuracy of at least 95%. The type of noise studied here is white Gaussian noise, which represents any internal or external noise on the sensor signals. The models proved to be robust to this type of noise as their accuracy did not decrease despite the addition of noise with signal-to-noise ratio (SNR) up to 15 dB. The proposed noisy training approach increased the performance of the models in terms of accuracy and robustness to noise.

The remainder of this article presents related work on noisy learning in Section II. Section III describes the workflow of the work presented in this article. The description of the data, the presentation of the proposed approach, and the implementation details of the different models designed are given in Section IV. Sections V and VI present, respectively, the results obtained and a discussion of the results. Finally, Section VII presents the conclusion.

II. Related Works

Authors in the literature frequently seek ways to improve the robustness of their ML models, with noisy training being a popular solution. This can be achieved by adding noise to various components, such as input data, weights, features, or labels.

Song et al. [1] published a study describing, from a supervised learning perspective, the problem of learning with noise at the label level and reviewing 62 state-of-the-art robust learning methods, grouped into five categories, according to their different methodologies.

Audhkhasi et al. [2] presents the noisy CNN (NCNN) algorithm for accelerating back-propagation (BP) training of CNNs. The NCNN algorithm is based on two theoretical results. The first one is that the BP algorithm is a special case of the generalized expectation-maximization (EM) algorithm for iteratively maximizing a likelihood. The second is that carefully chosen and injected noise accelerates the convergence of the EM algorithm on average. The NCNN algorithm was evaluated on standard MNIST test images for image recognition. Noise was added to the output neurons, the results showed a substantial reduction in the cross entropy of the training set and the classification error rate compared to the BP algorithm without noise.

Meng et al. [3] proposed a noisy training approach for speech recognition based on DNNs. Analysis and experiments with wall street journal databases have shown that by randomly selecting some noises and injecting them to corrupt the input speech during DNN training, noise patterns can be learned efficiently and generalization can be improved. Reference [4] is an extended version of [3] where the authors contributed by examining the behavior of noise injection in DNN training and studied a mixture of multiple noises at different levels of SNR. This version presents a full discussion of the technique and reports extensive experiments.

Goldberger and Ben-Reuven [5] studied the problem of training neural networks that are robust to label noise. They proposed a neural network approach that optimizes the same likelihood function as that optimized by the EM algorithm. Noise is explicitly modeled by an additional softmax layer that links correct labels to noisy labels. Thus, the algorithm can be easily combined with any existing deep learning implementation by simply adding another softmax output layer. Experimental results on two public datasets, MNIST and CIFAR-100, have shown that they can learn the noise distribution reliably and that their approach outperforms previous methods which assumed that the noisy label is independent of the feature vector.

A detailed analysis of the effects of different types of label noise on learning is provided in [6] and a generic label corruption algorithm with feature-dependent synthetic label noise is provided. The proposed methodology uses the distillation technique to create a dispersed distribution of data in the learned feature domain and thus emphasizes the similarities between the data samples by dispersing them in the feature domain. The authors have shown that using this approach, the samples most likely to be mislabeled are detected from their softmax probabilities, and their labels are returned to the relevant class.

Qin and Vucinic [7] studied two issues of noisy computations during inference. The first one concerns how to mitigate their side effects on naturally trained deep learning systems. Using a voting method and injecting noise into all layers of the neural networks during training, the authors achieved an improvement in inference accuracy from 21.1% to 99.5% for MNIST images, from 29.9% to 89.1% for CIFAR10 and from 15.5% to 89.6% for MNIST attack sequences. The second issue concerns the use of noisy inference as a defensive architecture against adversarial black-box attacks. The idea was to inject appropriate noise into the neural networks during inference. As a result, the robustness of adversarial-trained neural networks was improved by 0.5% and 1.13% for two models trained on MNIST and CIFAR10 databases, respectively.

Matsumoto et al. [14] propose a classification method with a combined DNN/LSTM approach that provides a robust classification of charge and spin states of electrons in semiconductor quantum dots as an alternative to the recent numerical classification method in this field. Therefore, they designed a DNN and evaluated the robustness of the classifier against different noise environments. Depending on the used SNR noise value, the DNN proved to have higher accuracy than the alternative classification method of thresholding and thus proved to be more noise robust in certain applications.

Lecuyer et al. [16] and Liu et al. [17] mainly used noisy training to make their neural networks more robust to adversarial examples that normal neural networks are vulnerable to. In this article, the use of noisy training was extended and applied to train a network that is more resilient to noisy input data, especially to noisy sensor signals. Therefore, noisy training was applied as one regularization method to increase the noise-robustness of the DCNNs.

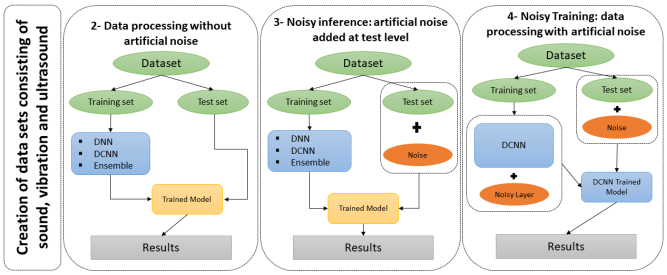


Fig. 1. Workflow of this study.

III. Overview of This Work

The work done in this article is divided into four points as shown in Fig. 1. Unlike several authors who have studied the issue of noisy training in the field of computer vision or speech recognition with benchmark datasets as presented in Section II, this work has been oriented toward the industrial domain to contribute to the predictive maintenance problem.

The first step in this work was to create datasets. Experiments were carried out on a milling machine and on laboratory equipment consisting of a motor with loads mounted on it, to obtain data to monitor these systems, in particular, to assess their health status. Section IV-A details the data acquisition process and provides a description of the data.

The second step was to use the collected data to train the different models. In this step, the data were used without the addition of external or artificial noise. The DCNN algorithm was used to train the raw data for the detection of milling machine health conditions, while the DNN and ensemble algorithms were used to train the specific features extracted from the raw data to monitor the motor. Two types of learning were implemented, namely deep learning and classical learning, to evaluate the noise resistance of each of the models resulting from these methodologies. Deep learning was evaluated with the milling machine data, as the constructed dataset was larger than that constructed from the laboratory setup. The data from the laboratory were therefore evaluated by traditional learning methods, hence the selection of features to form the input data.

The third step was to assess the robustness or noise resistance of the trained models. To do this, the pretrained models were tested with the test datasets to which artificial noises were added. For all models, the same type of noise was added and evaluated at several levels. Section IV-B describes the type of noise used and the different noise levels evaluated.

The final step, noisy training, was implemented to create a more robust monitoring model. This step consists of injecting noise during the training of the model. Noise can be introduced in the inputs, weights, inner layers, outputs, etc. In this work, two types of injections have been implemented. Section IV-B3 presents the different noise insertion processes implemented at the training stage.

IV. Experimental Details

A. Datasets Description

Three types of data were collected during the experiments, namely, sound from the Knowles SPU0410LR5H microphone with a sensitivity of −26 dBV/Pa and an SNR of 64 dB. It consists of an acoustic sensor, a low-noise input buffer, and an output amplifier. Vibrations were gathered from the AD ADXL1005 accelerometer with a sensitivity of 500 mV/g and 120 $\mu\text{g}/\sqrt{\text{Hz}}$ of noise. It offers ultralow noise density over a wide range of frequencies. Finally, an ultrasound that uses the microphone quoted above with the same settings. Only the microphone and accelerometer data were used to assess the condition of the milling machine, as the ultrasound data were extremely obstructed during the milling process. However, further experiments were carried out on a laboratory setup composed of motor-mounted loads to build up an ultrasound dataset.

1) Milling Dataset: The dataset was constructed by performing the milling process on the metal. Figs. 2 and 3 show, respectively, the milling tool used with the different types of signals recorded and the amount of wear on the tool. Indeed, the sensors were mounted on the tool, on the one hand, and, on the other hand, connected to a sensor node, a Red Pitaya board. First, a milling tool expert inspected the tool visually and with dedicated equipment to identify any visible signs of wear or damage. Next, the tool was used in a machining process to collect data on its performance, including vibration and sound. Wear measurements were also made using a microscope. Finally, the expert used his knowledge of milling tools and experience with similar tools to interpret the data and determine the health of the tool. The measured signals are transmitted via the Red Pitaya to a computer where they are recorded in text files. Five categories have been identified based on the combination of expert knowledge and wear measurement data. From the healthiest to the most degraded state of the tool, the classes are: "++," "+," "o," "−," "− −."

Several datasets have been constructed using this system by varying multiple operating conditions. Suawa and Hübner [18] presents findings from three of these datasets. For the purpose of this article, the dataset used is the one collected under the following conditions: "sample rate": 1.9 MHz, "feed rate": 360 mm/min, "rotational speed": 8000 r/min, "cutter name": k4f2, "diameter of the cutter": 8 mm, "step over": 45%, and "depth of cut": 2.5 mm. Where the "feed rate" refers to the speed at which the cutting tool moves through the material being milled, the "rotational speed" is the speed at which the spindle of the milling machine rotates, the "step over" refers to the distance that the cutting tool moves laterally between each pass as it mills a workpiece and the "depth of cut" is the distance the cutting tool moves vertically into the workpiece on each pass.

For each sensor, a total of 11.991 measurements were taken, each comprising 16.381 data points. Models were trained on the data from each sensor on the one hand and on the fused data from both sensors on the other. In each case, 70% of the data were used for the training phase and 30% for the test phase.

2) Ultrasound Dataset: Another measurement approach to determine the state of the tool is an ultrasound-based measurement. When measuring vibrations, it would be optimal to measure directly at the tool. Since the tool is rotating at high speeds up to 10k r/min and has a small diameter of less

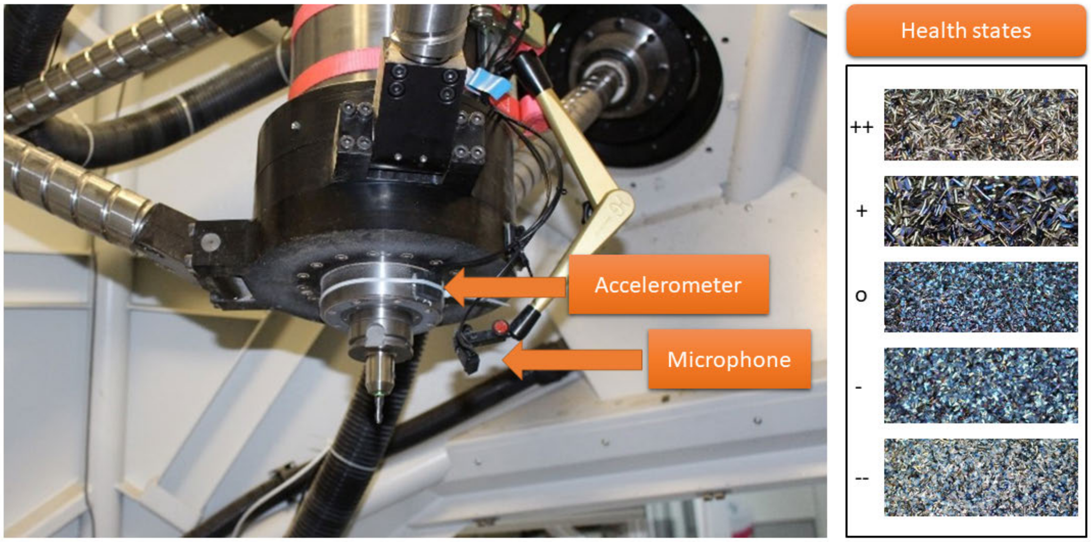


Fig. 2. Milling tool.

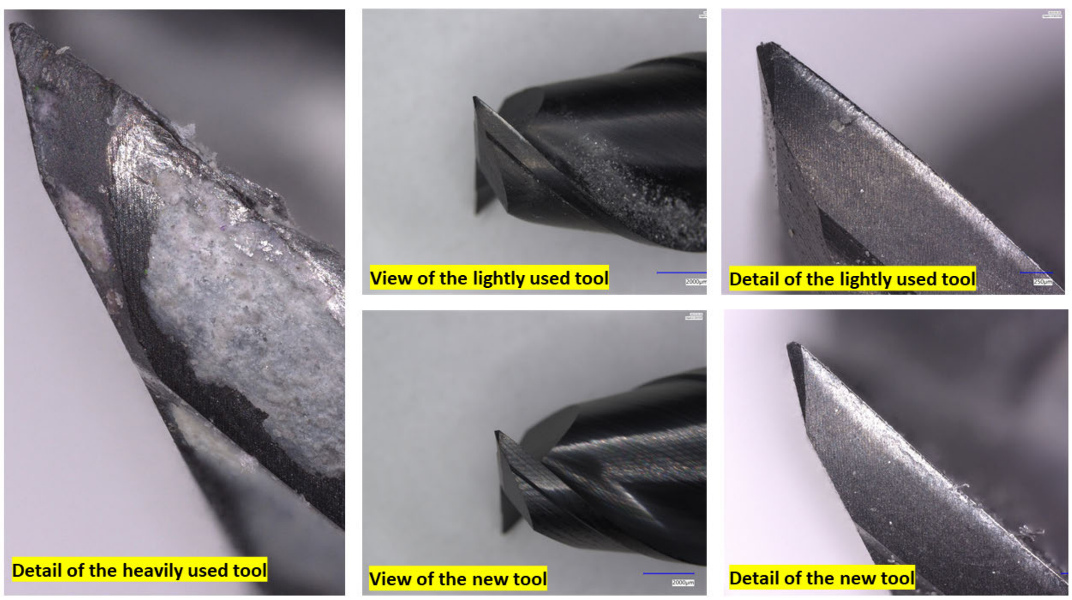


Fig. 3. Images light microscope of the cutting edge at 100× magnification.

than 10 mm, it is not easy to attach a sensor to it. Hence the idea of using an ultrasound measurement. As the tool rotates, it causes an airflow around the tool. In addition, it could slightly vibrate. Both effects affect ultrasound waves, which are directed toward and reflected from the tool. This way, the results of the prediction could be improved.

However, even after applying digital filtering, the signal was too noisy from other machine noises. Whenever the tool was spinning in the air, the measurements were as expected. With the tool in the air and increasing rotational speed, it was also possible to see a changing pattern in the ultrasound measurements. The time-of-flight would change more erratically as the rotational speed increases. When the tool touched the workpiece, this signal became less homogeneous and clipping occurred often at seemingly random events. This encouraged measurements in the lab, to get more insight into the required SNR, at which the ML models still produce reliable results.

The data from the ultrasound measurements are, consequently, not based on the measurements in the milling machine, but on a laboratory setup. A motor was, in different measurement series, equipped with a balanced or unbalanced load. The ultrasound transmitter was aimed at the load and the reflected signal was acquired by a wideband MEMS microphone and filtered using a digital bandpass filter for the relevant frequencies. The measurement system is the same as is applied in the milling machine measurements. A series of pulse-echo measurements were performed with 20 pulses per second. From the changes of the received echoes over time, 28 parameters are calculated: maximum, minimum, absolute maximum, sum, median, mode, mean, absolute mean, standard deviation, energy quantification related, root-mean-square (rms), rms error, peak value, peak-to-peak value, peak-to-rms value, skewness, kurtosis, crest factor, impulse factor, and shape factor from the time domain, and mean, variance,

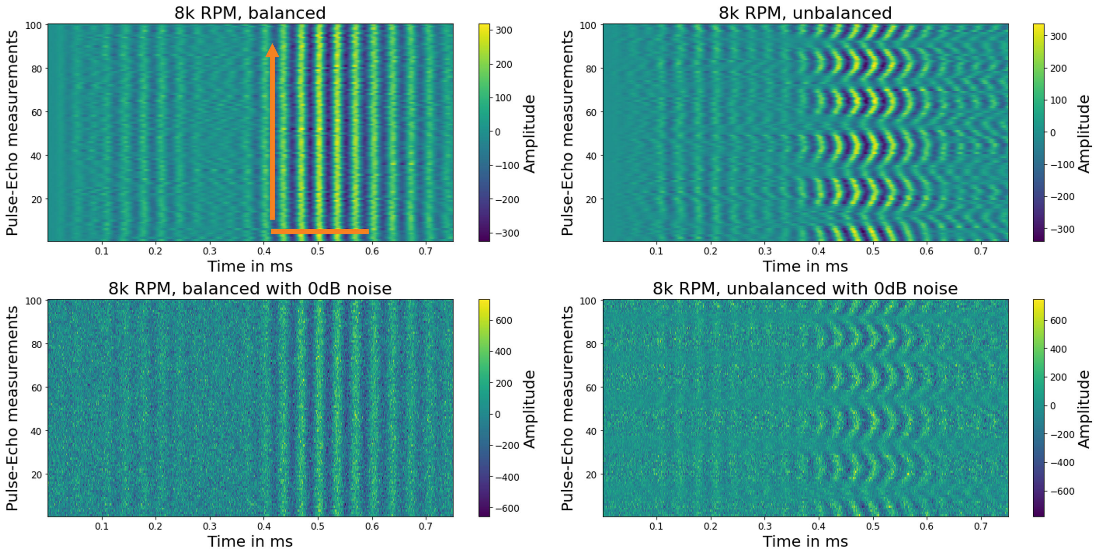


Fig. 4. Measured signal of the ultrasonic pulse-echo measurements. The echo is received between 0.4 and 0.6 ms after sending the pulse. 20 measurements are done per second. The evaluation is done between the pulses as annotated by the arrow. Although 0-dB noise has been added to the data, the vibration pattern is still visible for the unbalanced load.

rms, frequency center, root variance frequency, mean energy, and center frequency from the frequency domain. They have been adapted from Hamadache et al. [22]. This was done for the balanced load at rotational speeds of 5k, 6k, 7k, 8k, 9k, and 10k r/min, and for the unbalanced load of rotational speeds up to 8k r/min.

An example of the generated data is given in Fig. 4. The echo can be received about 0.4 ms after the pulse is transmitted. The received echo is oriented along the x-axis. 20 pulses are transmitted per second, which can be seen on the y-axis. For the balanced load, feint changes in the signal can be seen. The vibrations caused by the unbalanced load give a specific pattern, which can even be seen in the noisy data. Since the noise is calculated per echo, i.e., along the x-axis, it simulates noise added by the sensor itself or the amplifier circuit. The evaluation is done along several pulses to evaluate the changes over time, i.e., the vibration. In terms of comparability of all data presented in this article, the balanced load is closer to the milling dataset in the way the noise affects the measurements.

B. Machine Learning Models With Noise Injection

This section describes the type of noise used in this study, the different noise levels assessed, and the different methods implemented.

1) Noise Description: According to [20] the term "noise" describes every type of disturbance that is superimposed on a signal. The signals captured by the microphone, accelerometer, and ultrasound in this study are significantly affected by external influences. Hence, the purpose of this article is to explore how these external factors impact the accuracy of ML models. As examining external influences in detail would necessitate making new adjustments depending on the frequency of external noise for every sensor environment change, white random noise was chosen to represent these influences in the existing models.

White noise describes a signal with a constant noise power density and independence of the frequency domain and is distributed as $\mathcal{N}(0, \sigma^2)$ with the variance σ^2 as a measure for the power of the noise [20], [21]. Gaussian white noise is a special type of white noise with Gaussian-distributed random variables [23].

For the description of the noise level, it is very common to use the parameter of SNR which describes the ratio of useful signal to interference signal. The SNR is mostly given in dB because the size of the signal and the size of the noise might differ greatly. For voltage, the SNR is calculated as (1), whereas A is the amplitude of the voltage [21]

$$\mathrm{SNR} = 20 \log \left(\frac{A_{\mathrm{signal}}}{A_{\mathrm{noise}}} \right) \ \mathrm{dB}. \tag{1}$$

2) Noise at Test Level: The noise may introduce inaccuracies or uncertainties in the test data, making it challenging for the model to make accurate predictions. Dealing with noise at the testing level is an important aspect of evaluating and improving machine learning models.

1) *For the Ultrasound Data:* The ML training and tests for the ultrasound data have been done in Python using scikit-learn and Tensorflow. The goal of the ML model is the prediction of the rotational speed based on the parameters calculated from the ultrasound data. The balanced and unbalanced loads were examined in different models. The training was performed with nonnoisy data only.

   During the evaluation, artificial noise derived from a random number generator using a Gaussian distribution function was added with different levels to the signals of the test set which is different from the training set. The level ranges from 40 dB, where the amplitude of the noise is 1/100th of the amplitude of the ultrasound signal, to −6 dB, where the amplitude of the noise is double the amplitude of the measured signal. The noise was added before the extraction of the features

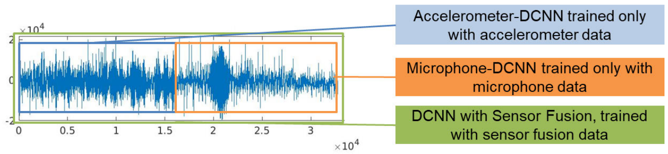


Fig. 5. Different DCNNs trained with the sensor signals of the milling machine.

that were introduced in the ML models. Two different models are compared. A DNN trained with Tensorflow and an ensemble of different approaches in scikit-learn, both done in Python.

For the evaluation based on the scikit-learn toolbox, three models are trained. An RFC, KNN, and SVC were individually trained. The hyperparameters were optimized by the grid search functionality of scikit-learn. For the final evaluation, these three models were combined into an ensemble. In a second approach, a DNN is trained using Tensorflow. It consists of three fully connected layers with 300, 200, and 100 neurons, respectively, using a rectified linear unit (ReLU) activation. The final layer is a softmax-activated layer with a number of neurons equal to the number of categories, i.e., the rotational speed.

2) *For the Milling Machine Data:* As a starting point for the different investigations, three DCNNs models were trained with nonnoisy data from the microphone, the accelerometer, and data from the two sensors combined, as shown in Fig. 5. These models were implemented using MATLAB according to the structure given in Table I without layers 2 and 5, and with the following parameters: Optimizer: Adam, Learning Rate: 0.001, Number of epochs: 130, Minibatch Size: 2000.

To assess the robustness to noise of the different DCNNs, they were tested with several sets of noisy test data. Qin and Vucinic [7] and Matsumoto et al. [14] present the benefits of injecting noise into the test data to examine the robustness of the trained network. Noisy test datasets were generated by adding white noise described in Section IV-B1 with a Gaussian normal probability distribution $\mathcal{N}(0, \sigma^2)$ of varying intensity to the matrix of the test dataset. To check which intensity of noise has which effect on the performance of the neural network, several noisy test datasets with varying noise levels were created. The additional white noise was modeled based on different SNR levels, using noise levels from SNR of −4–32 dB, whereas an SNR of 32 dB describes a low SNR and thus few noise addition to the sensor signal. An example of an original sensor signal of the microphone and the corresponding sensor signal with the addition of white Gaussian noise with an SNR of 14 dB can be seen in Fig. 6.

For testing the DCNN with sensor fusion, it was also an objective to find out which sensor is more resilient to noise and how sensor fusion can help to achieve better results in accuracy. To test the noise resilience dependent on the accelerometer and the microphone, the respective noise was added in three steps. First, the noise was applied only to the accelerometer signal. Then the noise was applied to the microphone signal, and finally, the noise was added to both sensor signals. The noisy datasets were fed into the neural network and the results were evaluated with a confusion matrix and the corresponding accuracies.

TABLE I
LAYER STRUCTURE OF THE DCNNS WITH TWO GAUSSIAN NOISE LAYERS, EXAMPLE FOR A COMBINED APPROACH ON THE DCNNS FOR ACCELEROMETER AND MICROPHONE (16 381 DATAPOINTS INPUT)

No.	Layer	Number and Size of Filters
1	Input	1x16381x1
2	Gaussian Noise	1x16381x1
3	Convolution	1x16282x20
4	Batch Normalization	1x16282x20
5	Gaussian Noise	1x16282x20
6	ReLU	1x16282x20
7	Max Pooling	1x8141x20
8	Convolution	1x8042x30
9	Batch Normalization	1x8042x30
10	ReLU	1x8042x30
11	Max Pooling	1x4021x30
12	Convolution	1x3922x40
13	Batch Normalization	1x3922x40
14	ReLU	1x3922x40
15	Max Pooling	1x1961x40
16	Convolution	1x1862x50
17	Batch Normalization	1x1862x50
18	ReLU	1x1862x50
19	Max Pooling	1x931x50
20	Convolution	1x832x60
21	Batch Normalization	1x832x60
22	ReLU	1x832x60
23	Max Pooling	1x416x60
24	Fully Connected	1x1x5
25	Softmax	1x1x5
26	Classification Output	1x1x5

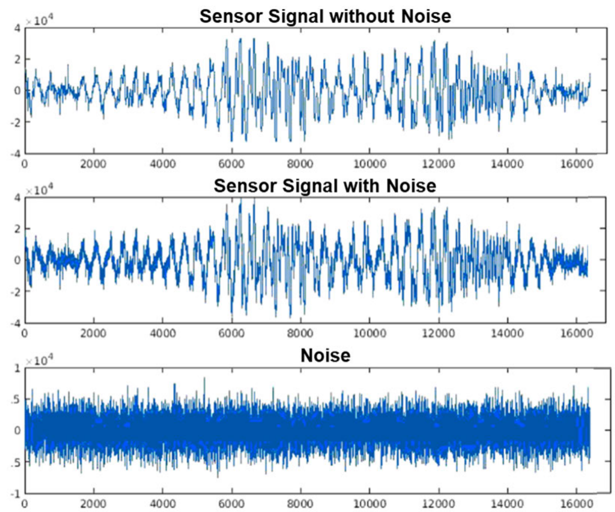


Fig. 6. Exemplary microphone sensor signal with the addition of white Gaussian noise (SNR = 14 dB).

3) Noise at Training Level: In literature, noisy training in terms of adding noise during the training phase is commonly known as a method to increase generalization and thus the robustness of the network, and to reduce overfitting of the trained DCNN [11], [12], [13], [15]. Because the goal was

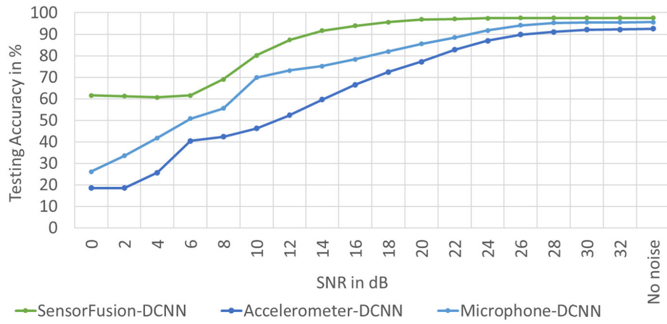


Fig. 7. Results of noise injection to the three DCNNs trained with data from the milling machine.

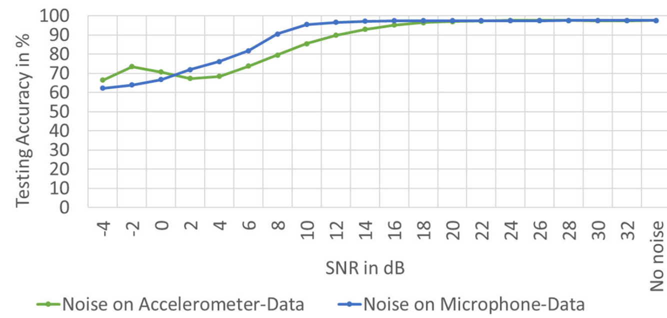


Fig. 8. Results of noise injection to one-half of the test data to the DCNNs with sensor fusion.

TABLE II
PARAMETERS OF DIFFERENT NOISY TRAINED NETWORKS FOR MICROPHONE AND ACCELEROMETER DATA

DCNN-Name	Position of Gaussian Noise Layer	Sigma -Max	Step-Size
DCNN1	After Input Layer	1	-0.01
DCNN2	After Input Layer	5	-0.01
DCNN3	Before RELU-Layer	1	-0.01
DCNN4	Before RELU-Layer	0.4	-0.005
DCNN5	After Input-Layer and before RELU-Layer	0.5 0.1	-0.004 -0.002

not only to improve the generalization but also to improve the robustness of the DCNN to noisy input, this article focuses on a variety of noise injections close to the input data of the DCNN.

Within noisy training, the goal was to add Gaussian noise distributed with $\mathcal{N}(0, \sigma^2)$ to the data by integrating a Gaussian noise layer during the training of the DCNNs. Inspired by Zhou et al. [11] and Neelakantan et al. [19] a descending noise level was used within the Gaussian noise layer. This means that the used noise level decreases with every epoch until it finally equals zero during the last epochs and the Gaussian noise layer adds no noise to the input data. This approach optimizes the training in terms of avoiding overfitting the neural network to noise influence or especially noisy input data.

The execution of the descending noise was implemented by decreasing the value for the standard deviation σ as the input parameter for the Gaussian noise layer and thus the noise intensity decreased with every epoch. The descending σ level was chosen with a linear decrease in a range from $\sigma_{\max}(\text{epoch}_1)$ to $\sigma_{\min}(\text{epoch}_n) = 0$ whereas n is the number of epochs. The step size in Table II describes the linear decrease of the standard deviation σ with every epoch.

To observe which positions of the Gaussian noise layer and which σ-values lead to the most robust networks, different positions, and different σ-values were used to train the corresponding neural networks. Aside from the addition of Gaussian noise layers, the structure and all other parameters of the DCNN were left unchanged. The layer structure of the DCNNs with the two used positions for the Gaussian noise layer is shown in Table I. Table II shows the different combinations of the Gaussian noise layer's position and parameters for the noisy training of different DCNNs.

The noisy trained networks were tested with the original test dataset as well as the different noise-injected datasets to evaluate which network shows improved robustness, i.e., an increased accuracy toward noisy sensor signals.

V. RESULTS

The models developed in this study were evaluated using separate test data with balanced labels, and the accuracy metric was used to assess their performance. For the ultrasound data, the ensembles were trained in approximately 15 min, which included a grid search to determine hyperparameters and cross-validation. The DNN, using calculated parameters, took around 5 min per run, and the models were trained multiple times with the same hyperparameters to compare their accuracies, which differed by less than 1%. The DCNN took about 3 h per cycle for the milling data, and like the previous methods, the models were trained and tested several times with the same parameters to compare their accuracies, which also differed by less than 1%. It is important to note that the learning time varied based on the machine used, and in this study, a two-CPU server with 512-GB RAM was utilized.

A. Results of Noise Injection to the Milling-Machine DCNNs

During noise injection, the original DCNNs without noisy training have been tested with test data that has been injected with Gaussian white noise. According to the findings presented in Fig. 7, the accuracy of all networks decreased as the SNR value decreased, indicating an increase in the level of white noise added to the original test dataset. Especially the original DCNN with sensor fusion shows very high robustness to noise with an accuracy decrease ($\leq$2%) at less than 18 dB. The DCNN of the accelerometer shows a strong accuracy decrease after only 28 dB and the microphone DCNN for less than 26 dB.

The results of noise injection to the DCNN trained with the data for sensor fusion show that noise on the second half of the signal (i.e., on the microphone data) proved to be less problematic than when noise was added to the first half of the signal (i.e., the accelerometer data) (see Fig. 8). The noise level of the accelerometer has then a greater impact on the accuracy of the original DCNN with sensor fusion than the microphone noise. If noise is only added to the microphone data, it does not strongly ($\leq$2%) influence the accuracy of the overall network until the noise level is smaller than 10 dB. These experiments show that the original DCNN with sensor

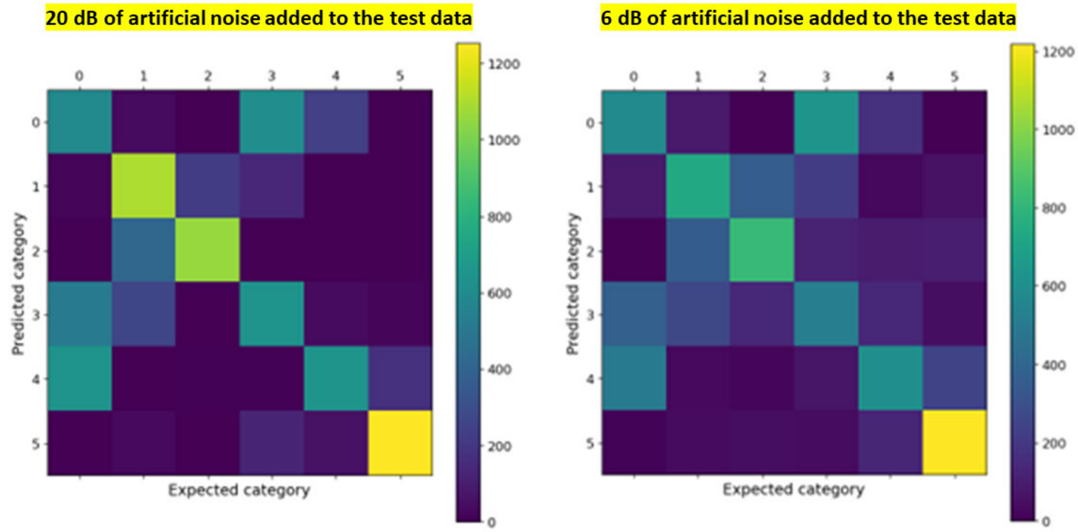


Fig. 9. Confusion matrix of the ensemble trained model with 20 and 6 dB of artificial noise added to the test data.

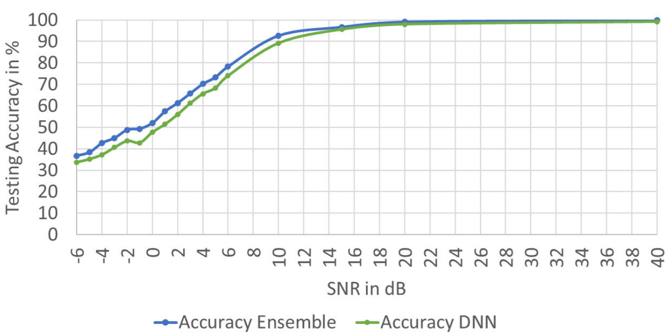


Fig. 10. Results of noise injection, balanced load.

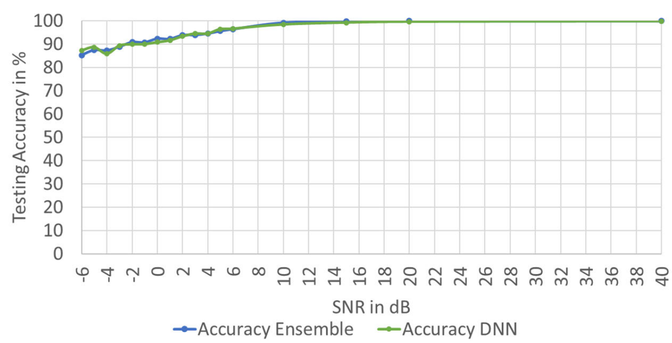


Fig. 11. Results of noise injection, unbalanced load.

fusion is a more stable and noise-robust network in comparison to the same DCNN that is only trained with one of the two sensor datasets.

B. Results of Noise Injection to the Ultrasound Data

Fig. 9 shows confusion matrices of the ensemble method, trained to predict the rotational speed of a balanced load. No noise was added to the training data. The categories 0–5 correspond to a rotational speed of 4k, 5k, 6k, 7k, 8k, and 10k r/min, respectively. The color bar shows the number of samples. Detection of the higher rotational speeds works quite well under all circumstances. The higher the speed, the higher the vibrations and subsequently, the better the distinction between the cases. For lower rotational speeds, the more easily the different cases are confused with each other. Figs. 10 and 11 present the results of the experiments performed on the ultrasound data. For the balanced load, one can see that the DNN and the ensemble have a similar performance with minimal losses in accuracy ($\leq$2%) until an SNR of 15 dB. When the level of the noise increases more, the accuracy starts to decline nearly linearly with the noise level. Overall, the ensemble of RFC, KNN, and SVC, offers a slightly better noise resistance (about 5%) than the DNN model. For the unbalanced load, this trend is similar.

TABLE III
TEST ACCURACY (%) OF ENSEMBLE AND DNN MODELS

Noise	Balanced Load		Unbalanced Load	
	Ensemble	DNN	Ensemble	DNN
-6	36.70	33.70	85.20	87.30
-5	38.30	35.10	87.40	88.70
-4	42.60	37.20	87.20	85.90
-3	45.00	40.60	88.80	89.40
-2	48.80	43.60	90.80	89.90
-1	49.30	42.80	90.70	90.00
0	52.00	47.70	92.30	90.80
1	57.40	51.40	92.20	91.60
2	61.30	56.00	93.80	93.30
3	65.80	61.20	93.90	94.50
4	70.20	65.60	94.50	94.50
5	73.30	68.30	95.60	96.40
6	78.30	74.00	96.40	96.60
10	92.70	89.20	99.20	98.40
15	96.70	95.60	99.80	99.20
20	99.10	98.00	99.90	99.60
40	99.80	99.20	99.90	99.80

Compared to the balanced load, the cut-off is at 10 dB and the decline is less steep (see the cells in blue in Table III). Keep in mind that the noise was added to the received echo, and the evaluation is done along several echos. The changes between the echos, from which the parameters are extracted, are not as influenced by the noise as the received echo itself (refer to Fig. 4).

TABLE IV
TEST ACCURACY (%) OF DCNN MODELS TRAINED ON DATA FROM A RANGE OF ACCELEROMETER DATA WITH AND WITHOUT ADDED NOISE

SNR (dB)	Initial DCNN-ACC	DCNN1-ACC	DCNN2-ACC	DCNN3-ACC	DCNN4-ACC	DCNN5-ACC
-4	18,49	18,49	18,49	18,49	22,69	18,49
-2	18,49	18,49	18,49	18,49	22,69	18,49
0	18,49	18,49	18,49	18,49	22,69	18,51
2	18,49	18,49	18,49	18,49	22,69	18,79
4	25,69	18,49	18,49	18,49	22,74	33,00
6	40,42	18,49	18,49	19,29	23,35	41,20
8	42,31	19,71	19,24	43,48	25,27	41,92
10	46,26	35,17	26,52	48,46	27,94	42,73
12	52,38	50,18	47,90	55,91	34,84	48,82
14	59,66	63,25	68,00	64,28	51,54	60,38
16	66,53	71,50	75,51	70,50	68,09	69,89
18	72,48	75,37	78,84	75,29	75,15	76,12
20	77,26	78,32	83,04	78,57	78,62	79,93
22	82,79	83,21	87,57	81,51	82,71	85,38
24	87,02	87,27	90,83	84,71	86,99	89,32
26	89,85	89,21	92,63	87,10	89,99	92,16
28	91,10	90,66	93,33	88,52	91,24	93,22
30	92,05	91,19	93,94	89,05	92,13	93,75
32	92,22	91,47	94,16	89,49	92,47	94,13
No noise	92,52	91,94	93,91	89,21	92,55	94,16

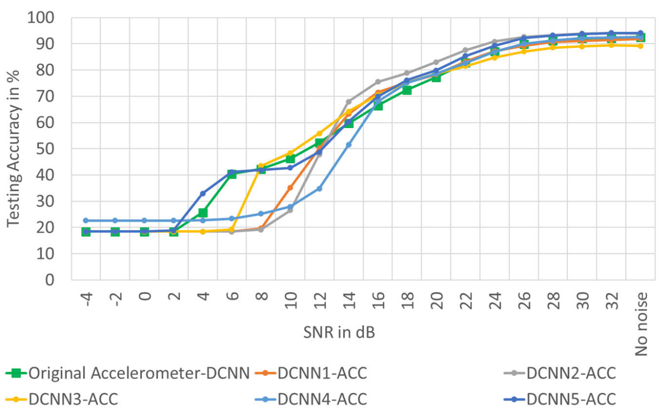


Fig. 12. Test results of the noisy trained accelerometer-DCNNs with accelerometer data (ACC used as an abbreviation for accelerometer.)

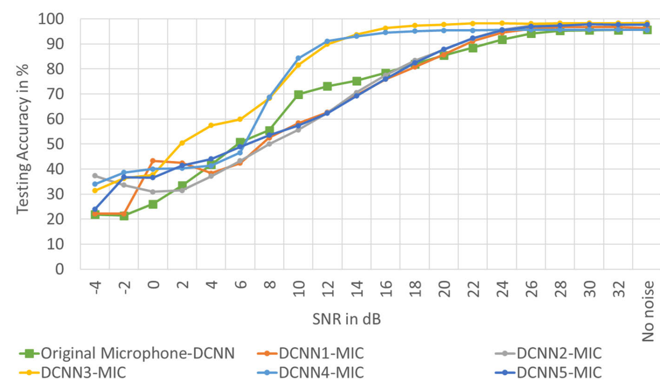


Fig. 13. Test results of the noisy trained microphone-DCNNs with noisy microphone data (MIC used as an abbreviation for microphone.)

The unbalanced load is basically a vibration motor. It is designed to cause vibrations, while the balanced load rotates smoothly without creating a lot of vibrations. Due to the increased influence on the ultrasound data by the increased amplitude of the vibrations, the data generated by the unbalanced load are more noise resistant. For the balanced load, the deviations in the ultrasound signal are mostly caused by the turbulent flow around the load.

C. Results of Noisy Training With the Milling-Machine DCNNs

Since the results of noise injection to the test data show, that the DCNNs without sensor fusion are not very noise-robust, different methods of noisy training have been applied for the DCNNs of the accelerometer and the microphone. To further improve the noise robustness of the DCNNs, noisy training was executed by adding a Gaussian noise layer with different levels of noise and different positions for noise injection inside the DCNN and subsequently training of the DCNN. The results show that most used combinations lead to an improvement in testing accuracy with noise-injected test data.

For noisy training with the accelerometer data, all of the noisy trained networks resulted in an increase in test accuracy between 16 and 22 dB (see Fig. 12). Especially DCNN2 with a Gaussian noise layer of $\sigma_{\text{Max}} = 5$ achieved the biggest improvement with up to 12% better accuracy (at SNR = 14 dB) than the original DCNN without noisy training (see the yellow cell in Table IV).

Especially for the microphone data, the noisy trained networks DCNN3 and DCNN4 where the Gaussian noise level was implemented between the first convolution layer and the first RELU-layer showed a very high improvement in accuracy for an SNR of more than 6 dB (see Fig. 13). Those two networks show a drop in accuracy ($\leq$2%) at less

TABLE V
TEST ACCURACY (%) OF DCNN MODELS TRAINED ON DATA FROM A RANGE OF MICROPHONE DATA WITH AND WITHOUT ADDED NOISE

SNR (dB)	Initial DCNN-MIC	DCNN1-MIC	DCNN2-MIC	DCNN3-MIC	DCNN4-MIC	DCNN5-MIC
-4	21,85	22,13	37,31	31,36	34,03	23,91
-2	21,41	22,19	33,58	36,34	38,62	36,81
0	26,08	43,29	30,92	37,70	40,09	36,56
2	33,47	42,51	31,44	50,49	40,34	41,45
4	41,81	38,34	37,14	57,49	41,31	44,04
6	50,79	42,37	43,31	59,94	46,57	48,87
8	55,55	52,57	50,01	68,31	68,67	53,55
10	69,86	58,44	55,71	81,60	84,29	57,38
12	73,17	62,69	62,52	89,88	91,13	62,33
14	75,23	69,53	70,61	93,72	93,08	69,20
16	78,40	75,90	77,54	96,33	94,55	76,06
18	82,01	80,79	83,35	97,36	95,14	82,40
20	85,46	85,71	87,68	97,69	95,41	87,82
22	88,49	91,13	92,22	98,19	95,39	92,36
24	91,77	94,50	95,69	98,33	95,58	95,47
26	94,19	95,91	97,25	98,14	95,66	96,97
28	95,30	96,66	97,94	98,30	95,66	97,33
30	95,47	96,80	98,28	98,25	95,69	97,78
32	95,55	96,75	98,28	98,33	95,64	97,72
No noise	95,66	96,39	98,44	98,28	95,58	97,75

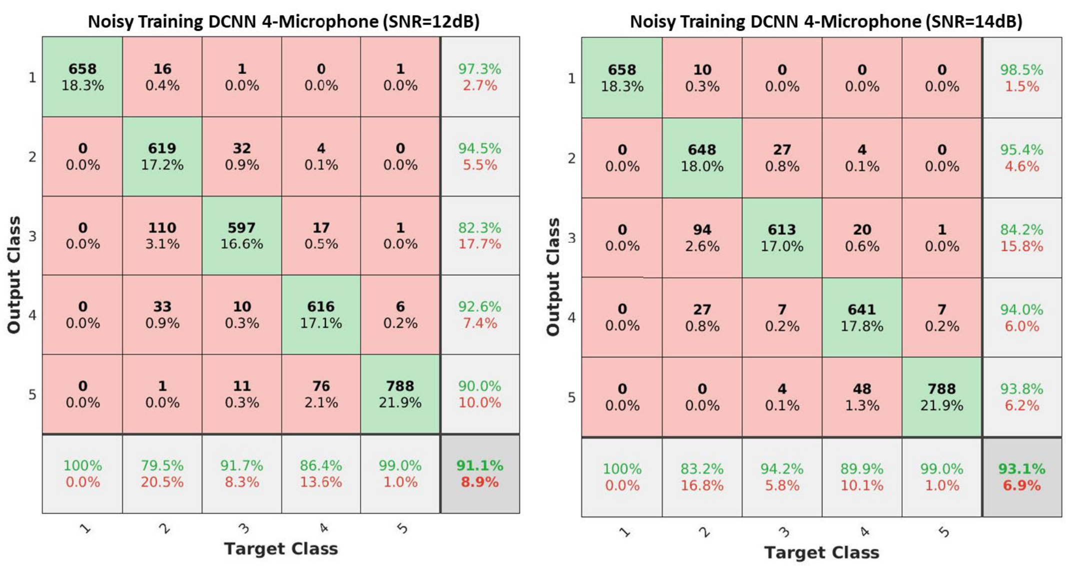


Fig. 14. Confusion matrices of the results for microphone DCNN4-MIC at SNR = 12 and 14 dB.

than 14 dB, way later than the original microphone-DCNN. Table V gives the overall statistics of the different models and the blue part represents the networks with the greatest improvement in accuracy. Confusion matrices in Fig. 14 are examples of matrices from the DCNN4-MIC where the numbers 1–5 indicate the different classes "++," "+," "o," "−," and "− −," respectively. Noisy training also improved the general test accuracy of the noisy trained DCNN2, DCNN3, and DCNN5 by more than 2% when tested with normal nonnoisy data, which shows that also a better generalization was achieved using noisy training as a method of regularization. Especially for the DCNN of the microphone, the noise robustness could be increased with noisy training with a Gaussian noise layer before the first RELU layer.

VI. DISCUSSION AND OUTLOOK

For the ultrasound dataset, classical learning algorithms were selected because the features were extracted manually to reduce the volume of data, as the overall goal is to create an application running on a microcontroller installed directly on the equipment. For the milling dataset where no manual feature extraction step was implemented, the deep convolutional learning algorithm was selected for its ability to automatically extract features through multiple convolution layers and pooling operations. The first approach is attractive when a knowledge expert is present, as traditional methods often require manual feature engineering and threshold adjustment, although this can be time-consuming and labor-intensive. The second method automatically extracts features and learns patterns from raw sensor data, reducing the need for human intervention.

Although two completely different datasets have been examined using different ML approaches, a similar trend can be found. The ML algorithms based on feature extraction show, depending on the used data size, no significant ($\leq 2\%$) decrease in accuracy until an SNR level of 15 dB (the amplitude of the noise is 1/5th of the signal's amplitude). At higher noise levels, the accuracy decreases nearly linearly. The DCNN was already showing a significant decrease in accuracy at SNR levels of 28 dB (noise amplitude is 1/32nd of the signal's amplitude) for the model trained on single sensor data, i.e., the accelerometer and the microphone. By implementing sensor data fusion and noisy training, this threshold was increased to a noise level of 15-dB SNR.

With the approach of using noisy training not only as a method of regularization and generalization improvement but to train the neural network to be robust against noise in the sensor signals, this article proposes a new way of training robust neural networks for use in predictive maintenance applications without any further financial effort. As the results show, a noisy training approach with an additional Gaussian noise layer close to the input and a descending noise level to avoid overfitting led to a significant improvement in the noise robustness of the DCNNs. A DCNN that is robust toward white noise is hereby characterized by high accuracies even with a high degree of injected noise in comparison to the original neural network without noisy training as well as a slow decrease in accuracy with a rising level of noise injection (i.e., decreasing SNR-value).

As a continuation of this research, further explorations are planned to evaluate the feasibility of utilizing ultrasound sensors on the milling machine. In addition, there is interest in incorporating noisy training into the traditional machine learning algorithms used in this study and evaluating the proposed noisy training method and models on various machines using transfer learning to develop a comprehensive framework for monitoring milling machines.

VII. Conclusion

This study aimed to improve industrial equipment monitoring by implementing noise-robust ML models. Two datasets were created from experiments on a milling machine and a motor, including vibration, sound, and ultrasound data. Three methods were used to build the models: DNN, ensemble, and DCNN. The latter was applied to raw signals, while the former two used extracted features.

The same type of noise was used and evaluated at several levels for both datasets. This is white Gaussian noise, used as a means to summarize all internal and external noise on the sensor signals. This noise was added to the test data used as input to different ML models to study the robustness of the systems to noisy input signals. Even though two different datasets were used, the same degree of robustness to noise of the different proposed models was observed. With an accuracy of at least 95%, these models maintain their performance up to a level of 15 dB of noise added to the signals. It should be noted that the DCNN models derived from training on single sensor data did not initially achieve the same performance as the other models. However, the robustness of these models was improved by the proposed noisy training, which consists of using additional Gaussian noise layers near the input with a descending noise level.

Acknowledgment

The authors would like to thank Philipp Städter from the Chair of Automation Technology at BTU Cottbus - Senftenberg and Tenia Meisel who worked on her bachelor thesis at the Fraunhofer Institute for Photonic Microsystems IPMS for their help in collecting the data.

References

[1] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 7, 2022, doi: 10.1109/TNNLS.2022.3152527.

[2] K. Audhkhasi, O. Osoba, and B. Kosko, "Noise-enhanced convolutional neural networks," *Neural Netw.*, vol. 78, pp. 15–23, Jun. 2016, doi: 10.1016/j.neunet.2015.09.014.

[3] X. Meng, C. Liu, Z. Zhang, and D. Wang, "Noisy training for deep neural networks," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process. (ChinaSIP)*, Jul. 2014, pp. 16–20, doi: 10.1109/ChinaSIP.2014.6889193.

[4] S. Yin et al., "Noisy training for deep neural networks in speech recognition," *EURASIP J. Audio, Speech, Music Process.*, vol. 2015, no. 1, pp. 1–14, Dec. 2015, doi: 10.1186/s13636-014-0047-0.

[5] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," in *Proc. Int. Conf. Learn. Represent.*, 2017. [Online]. Available: https://openreview.net/forum?id=H12GRgcxg

[6] G. Algan and I. Ulusoy, "Label noise types and their effects on deep learning," 2020, *arXiv:2003.10471*.

[7] M. Qin and D. Vucinic, "Noisy computations during inference: Harmful or helpful?" 2018, *arXiv:1811.10649*.

[8] Z. Kowalczuk and T. Merta, "Evaluation of position estimation based on accelerometer data," in *Proc. 10th Int. Workshop Robot Motion Control (RoMoCo)*, Jul. 2015, pp. 246–251, doi: 10.1109/RoMoCo.2015.7219743.

[9] M. Kirkko-Jaakkola, J. Collin, and J. Takala, "Bias prediction for MEMS gyroscopes," *IEEE Sensors J.*, vol. 12, no. 6, pp. 2157–2163, Jun. 2012, doi: 10.1109/JSEN.2012.2185692.

[10] G. An, "The effects of adding noise during backpropagation training on a generalization performance," *Neural Comput.*, vol. 8, no. 3, pp. 643–674, Apr. 1996, doi: 10.1162/neco.1996.8.3.643.

[11] M. Zhou, T. Liu, Y. Li, D. Lin, Z. Enlu, and T. Zhao, "Towards understanding the importance of noise in training neural networks," in *Proc. 36th Int. Conf. Mach. Learn. (PMLR)*, 2019, pp. 7594–7602.

[12] H. Noh, T. You, J. Mun, and B. Han, "Regularizing deep neural networks by noise: Its interpretation and optimization," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1–10.

[13] I. Goodfellow, Y. Bengio, and A. Courville, "Regularization for deep learning," in *Deep Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 2016, ch. 7, pp. 224–270. [Online]. Available: http://www.deeplearningbook.org

[14] Y. Matsumoto, T. Fujita, A. Ludwig, A. Wieck, K. Komatani, and A. Oiwa, "Noise-robust classification of single-shot electron spin readouts using a deep neural network," *NPJ Quantum Inf.*, vol. 7, no. 1, p. 136, 2021. [Online]. Available: https://www.nature.com/articles/s41534-021-00470-7

[15] X. Liu, T. Xiao, S. Si, Q. Cao, S. Kumar, and C. Hsieh, "How does noise help robustness? Explanation and exploration under the neural SDE framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 279–287, doi: 10.1109/CVPR42600.2020.00036.

[16] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, “Certified robustness to adversarial examples with differential privacy,” in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 656–672, doi: 10.1109/SP.2019.00044.

[17] X. Liu, M. Cheng, H. Zhang, and C. J. Hsieh, *Towards Robust Neural Networks Via Random Self-Ensemble*, V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss, Eds., vol. 11211. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-030-01234-2_23.

[18] P. F. Suawa and M. Hubner, “Health monitoring of milling tools under distinct operating conditions by a deep convolutional neural network model,” in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2022, pp. 1107–1110, doi: 10.23919/DATE54114.2022.9774570.

[19] A. Neelakantan et al., “Adding gradient noise improves learning for very deep networks,” 2015, *arXiv:1511.06807*.

[20] K. Kark, “Thermisches rauschen,” in *Antennen und Strahlungsfelder*, 1st ed. Wiesbaden, Germany: Springer, 2022, ch.26, pp. 675–678, doi: 10.1007/978-3-658-38595-826.

[21] R. Heilmann, *Kenngrößen des Rauschens*, 1st ed. Wiesbaden, Germany: Springer, 2020, ch. 1, pp. 4–5, doi: 10.1007/978-3-658-29214-0.

[22] M. Hamadache, J. H. Jung, J. Park, and B. D. Youn, “A comprehensive review of artificial intelligence-based approaches for rolling element bearing PHM: Shallow and deep learning,” *JMST Adv.*, vol. 1, pp. 125–151, May 2019, doi: 10.1007/s42791-019-0016-y.

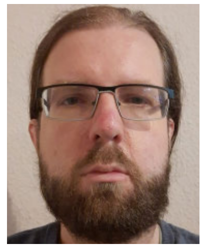
[23] O. Beuchler, “LTI-systeme und stochastische signale,” in *Signale und Systeme: Theorie, Simulation, Anwendung*, 2nd ed. Berlin, Germany: Springer, 2015, ch. 5, pp. 399–540, doi: 10.1007/978-3-662-45965-2.

Priscile Fogou Suawa received the B.Sc. degree in computer science and the M.Sc. degree from the University of Yaoundé I, Yaoundé, Cameroon, in 2014 and 2016, respectively. She is currently pursuing the Ph.D. degree with the Brandenburg University of Technology (B-TU) Cottbus–Senftenberg, Cottbus, Germany.

She was an Analyst Programmer in Computer Vision and Image processing at Camertronix SARL, Yaoundé, from 2017 to 2019. Her research interests include resilient systems with artificial intelligence, predictive maintenance, and sensor fusion.

Anja Halbinger received the B.Eng. degree in mechanical engineering and management from UAS Konstanz, Konstanz, Germany, in 2021, which she finished with her thesis in the development department of electrical drives at HILTI. She is currently pursuing the master’s degree in mechanical engineering and management with Friedrich-Alexander-University in Erlangen-Nuremberg, Erlangen, Germany.

In 2019, she worked for Voith Hydro, Heidenheim, Germany, in Global Product Management for Service and Maintenance of Hydropower Plants.



Marcel Jongmanns received the M.Sc. degree in biomedical engineering and the Ph.D. degree in engineering from the Brandenburg University of Technology Cottbus–Senftenberg, Cottbus, Germany, in 2015 and 2018, respectively.

Subsequently, he was a Research Associate at the Chair of Aerodynamics and Fluid Dynamics, Brandenburg University of Technology Cottbus–Senftenberg, Since 2019, he has been a Research Associate with the Fraunhofer Institute for Photonic Microsystems (IPMS), Dresden, Germany, with a focus on electronics development and AI-based data processing.

Marc Reichenbach (Member, IEEE) received the Diploma degree in computer science from Friedrich Schiller University Jena, Jena, Germany, in 2010, and the Ph.D. degree from Friedrich-Alexander University Erlangen-Nürnberg (FAU), Erlangen, Germany, in 2017.

From 2017 to 2021, he was a Postdoctoral Researcher with the Chair of Computer Architecture, FAU. Since 2021, he has been heading the Chair of Computer Engineering, Brandenburg University of Technology Cottbus-Senftenberg (BTU), Cottbus, Germany, as a Substitute Professor. His research interests include novel computer architectures, memristive computing, and smart sensor architectures for varying application fields.