1.6

Real-Time Predictive Maintenance – Artificial Neural Network Based Diagnosis

Petr Blaha[1], Matus Kozovsky[1], Zdenek Havranek[1], Martin Dosedel[1], Franz Wotawa[2], David Kaufmann[2], Adil Amukhtar[2], Iulia Nica[2], Florian Klück[3] and Hermann Felbinger[3]

[1]Brno University of Technology CEITEC, Czech Republic
[2]Graz University of Technology, Austria
[3]AVL List GmbH, Austria

Abstract

In this article, we discuss the use of artificial neural networks for monitoring and diagnosis to be used in the context of real-time predictive maintenance. There are two use cases analysed here. As a first one, we discuss the motor model used for diagnosis in detail. In particular, we introduce a detailed acausal six-phase e-motor model to be used for different stator and inverter faults simulations. The inter-turn short circuit fault is targeted here. Simulation data and data measured on a real custom-made six-phase motor with the ability to emulate this fault are pre-processed based on the mathematical analysis of the fault. Such data are then used for modular neural network training. The trained modular neural network is optimized and deployed into the NVIDIA Jetson platform. The second ANN presented in this article is designed for bearing fault detection based on vibration measurements. The vibration data taken from publicly available datasets are transformed into suitable condition indicators which are analysed by the multilayer perceptron network running on a PC in MATLAB with the possibility to implement the resulting network into a small edge device. As such, two use cases are shown how artificial neural networks can be used on edge devices. Obtained results show that the approaches can be used in real setups.

Keywords: AI based diagnosis, acausal model, artificial neural network, computing-at-the-edge, modular neural network, multilayer perceptron, multiphase PMS motor, vibration diagnosis, inter-turn short circuit fault, reliability, validation.

1.6.1 Introduction and Background

This article focuses on the demonstration of Artificial Neural Network (ANN) based monitoring and diagnosis of e-motors and mechatronic systems implemented on the edge directly in embedded devices. It reveals a theoretical analysis of existing methods provided in the article "Foundations of Real Time Predictive Maintenance" and it can be viewed as its two use-cases.

1.6.1.1 AI-based Diagnosis of E-motors

There are many recent papers dealing with the AI-based diagnosis of e-motors [1], [2], [3], [4], [5] and others. The authors describe the design of the ANN and provide the success rate of the network evaluation, still they either do not deal with on the edge implementation or mention that the integration is in progress. This paper tries to reduce the complexity of the proposed networks by suitable data pre-processing to be able to classify the measured data on the edge platform represented with embedded AI hardware and tends to practical implementation and the operation in real-time. The integration of fault diagnosis and predictive maintenance algorithms as close as possible to the motor try to support this trend. This article demonstrates AI-based diagnosis and predictive maintenance for e-motor running on the edge. The diagnosis discovers the issues which are potentially dangerous for the operation if they are ignored. Diagnosis combined with the redundancy and integration of predictive maintenance tasks can substantially increase the reliability and the availability of the powertrain.

Various methods to detect faults and unexpected behaviour of cyber-physical systems were proposed [6]. These methods require a large amount of experimental data for the learning process, or well-known system behaviour described by the model. Modelling a healthy system is a relatively simple task, on the other hand, modelling the system under fault conditions can be challenging. For instance, commonly used causal modelling methods can be used to create a healthy motor model, however, modelling of fault behaviour of the electric motor using causal models is difficult [7]. For these reasons, an acausal modelling approach was selected since it brings many benefits [8].

This type of model can be created in MATLAB/Simulink using Simscape or other simulation methods and tools like Modelica.

The requirements on e-motor safety integrity levels are continuously increasing. It holds for the motor for fully or hybrid electric vehicles as well as for common industrial motors. For the e-motor, it is demanded by the braking capability of the e-motor which is good for the energy recuperation, and by the progression towards autonomous cars. In industrial applications, it is required due to a higher level of automation and precise production planning.

1.6.1.2 Artificial Intelligence in Vibration Diagnosis

Nowadays, the Artificial Intelligence (AI) approach to vibration diagnosis is growing significantly and machine learning as well as deep learning algorithms, including neural networks (NNs), are becoming a part of vibro-diagnosis [9]. Both approaches are used in practice – simple statistic-based machine learning algorithms as well as complicated NN structures. Examples of such methods can support vector machines, decision trees, Bayesian classifier, Mahalanobis-Taguchi system etc., as representants of the machine learning algorithms, and convolutional NN, recurrent NN, shallow dense NN, etc., as representants of the deep learning techniques. The functionality of the algorithms is mainly demonstrated on the publicly available datasets or on real captured data on minor occasions. Success rate of the classification is relatively high and reaches values over 98 %. Because of the lack of real data, even describing many failures of the concrete machine, transferred learning algorithms are in the scope of view of the scientific community in the last few years. This procedure allows the algorithm to be learned using one type of data captured on one machine, transfer the knowledge and classify the faults on the second machine without prior training using data of such machine.

1.6.2 Artificial Neural Network for e-Motor Diagnosis

This section provides the first use case of ANN for the inter-turn short circuit fault detection in a six-phase motor. It is composed of two subsections. The first one outlines acausal e-motor model, which is used to prepare training datasets with the fault, which are either not realisable on a real customised motor or prepared faster and complement datasets from the measurements on a real motor. The second one presents the steps from the selection of suitable condition indicators, through the data pre-processing, MNN design, training, validation, towards MNN deployment on NVIDIA Jetson Xavier platform.

1.6.2.1 Acausal e-Motor Model with Faults Injection Capability

This section outlines the development of an acausal e-motor model for the six-phase motor (connected as two three-phase sub-systems) which is capable to inject several typical Permanent Magnet Synchronous Motor (PMSM) stator faults. This model was parameterized for the correspondence with the real custom-made motor equipped with many windings taps enabling to emulate these faults. They both can serve as sources of datasets for the ANN training and validation which is capable to diagnose the inter-turn short circuit fault.

The Simscape allows building physical component models in Simulink in a fast and natural way. Components and physical connections are directly integrated within block diagrams and other modelling paradigms. Individual Simscape components interact with each other. Each Simscape block is represented by a set of equations that describe the physical behaviour of components. Equations are automatically processed during the model compilation process. The motor converts electrical energy into mechanical rotating energy. The mechanical rotating components as the moment of inertia or friction block can be used to create the simple model of a motor mechanical part. The motor connection to the complex mechanical model is also possible using Simscape.

The electrical part of the dual three-phase motor model can be described by equation (1.6.1).

$$\boldsymbol{u}_{abc_{12}} = \boldsymbol{R}_{abc_{12}}\boldsymbol{i}_{abc_{12}} + \frac{\mathrm{d}\boldsymbol{L}_{abc_{12}}\boldsymbol{i}_{abc_{12}}}{\mathrm{d}t} + \boldsymbol{e}_{abc_{12}} \tag{1.6.1}$$

Conversion of electrical energy into mechanical torque can be characterized using the equation (1.6.2).

$$T_e = Pp\left(\frac{1}{2}\boldsymbol{i}^{\mathbf{T}}_{abc_{12}}\frac{\mathrm{d}\boldsymbol{L}_{abc_{12}}}{\mathrm{d}\theta}\boldsymbol{i}_{abc_{12}} + \frac{\boldsymbol{i}^{\mathbf{T}}_{abc_{12}}\boldsymbol{e}_{abc_{12}}}{\omega_{\mathrm{e}}}\right) \tag{1.6.2}$$

The mentioned equation can be used to emulate healthy motor model behaviour. This model can be extended and the equation for some coils are split into the serial connection of two coils with mutual inductances. The serial connection of the coils has the same behaviour as the original one. The voltage potential of any place of the original coil can be subsequently used to simulate electrical fault. This approach is demonstrated in Figure 1.6.1.

The variable $\boldsymbol{M}$ represents mutual inductance between the coil $\boldsymbol{L}$ and other motor windings. $\boldsymbol{R}$ represents windings resistance. Variable $\boldsymbol{e}$ denotes the influence of back-EMF voltage in windings. Parameter σ represents a division

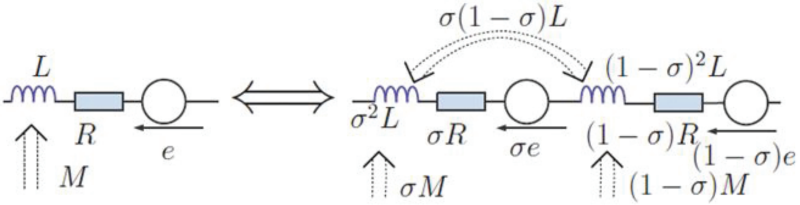

Figure 1.6.1 Winding equivalent for extended motor model.

ratio. The position of fault occurrence can be specified using this parameter. The coil splitting process is describable by equations (1.6.3).

$$u = Ri + \frac{\mathrm{d}Li}{\mathrm{d}t} + e \tag{1.6.3}$$

$$\begin{bmatrix} u_{1\_1} \\ u_{1\_2} \end{bmatrix} = \begin{bmatrix} R_{1\_1} & 0 \\ 0 & R_{1\_1} \end{bmatrix} \begin{bmatrix} i_{1\_1} \\ i_{1\_2} \end{bmatrix} + \frac{\mathrm{d}\begin{bmatrix} \sigma^2 L & \sigma(1-\sigma)L \\ \sigma(1-\sigma)L & (1-\sigma)^2 L \end{bmatrix}\begin{bmatrix} i_{1\_1} \\ i_{1\_2} \end{bmatrix}}{\mathrm{d}t} + \begin{bmatrix} e_{1\_1} \\ e_{1\_2} \end{bmatrix}$$

This description was used to create an acausal model of the dual three-phase machine able to emulate various internal motor faults. Internal short-circuits as well as disconnections in phases can be simply simulated. Figure 1.6.2 demonstrates various motor faults which can be simulated as well as emulated in the real motor. The model is used to generate important data sets for both healthy and faulty motors and for the transients from healthy to faulty states. These data sets can be used to train ANNs and for their validation.

1.6.2.2 Artificial Neural Network for Inter-turn Short Circuit Detection

This section shows the design of the DNN for inter-turn short circuit fault detection of PMSM. It starts with real experiments which were performed using the experimental motor with multiple windings taps which are capable to emulate this type of motor fault. The experiments helped with the selection of suitable condition indicator for fault detection. Further subsections describe data pre-processing, preparation of datasets and the process of training, validating and final deployment of DNN on embedded hardware.

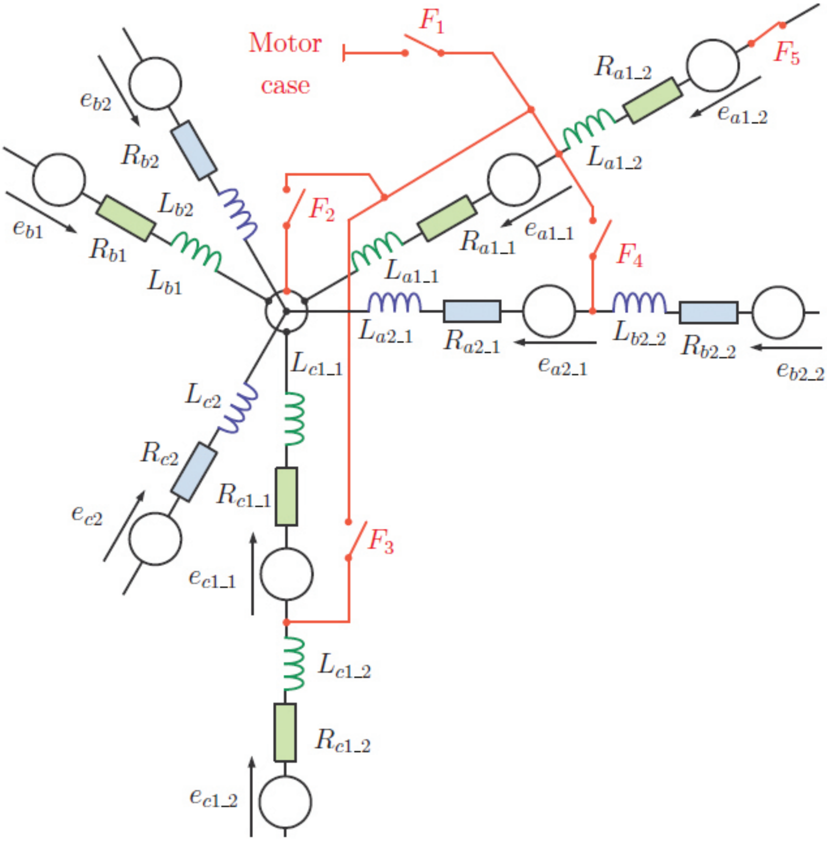

Figure 1.6.2 Simulated/Emulated faults in extended motor model / experimental motor.

1.6.2.2.1 Selection of suitable condition indicator for fault detection

Figure 1.6.3 demonstrates phase currents of both healthy and damaged motor sub-systems (only currents in the damaged sub-system are shown in this figure). Figure 1.6.4 shows phase currents transformed into *dq* coordinates. In this case, currents of both sub-systems are visible. As it can be observed, currents of damaged sub-system contain significant noise and distortion in a form of a significant second harmonic component. This is in accordance with the mathematical analysis of this fault as it is described e.g., in [8] and [11].

Phase currents or phase currents transformed into *dq* coordinates could be used as inputs to recurrent NN. This type of NN can filter the noise and consider not only the actual measurements but also previous ones.

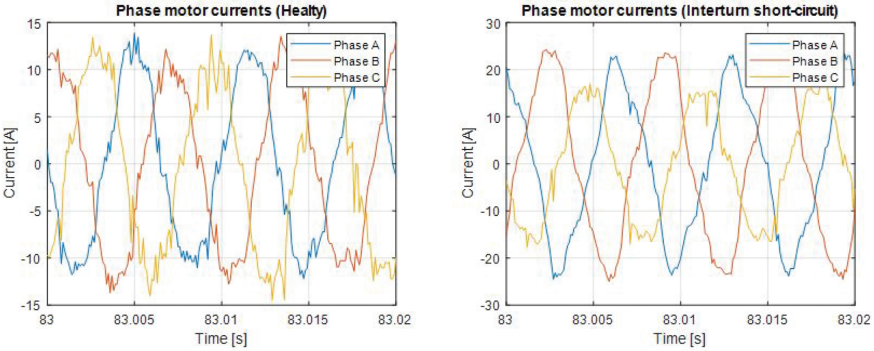

Figure 1.6.3 Phase motor currents (healthy/with fault).

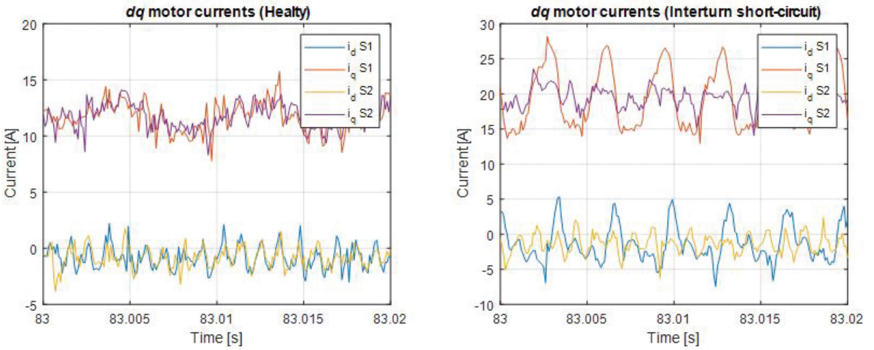

Figure 1.6.4 Motor currents in *dq* coordinates (healthy/with fault).

The computational complexity of such a NN would be high. On the other hand, linear NN would not be able to detect this fault properly using actual measurements as inputs due to high measurement noise. This problem can be overcome by suitable data pre-processing using the filtration method which is described later in this article.

Figure 1.6.5 shows low pass filtered motor current magnitudes in $\alpha\beta$ coordinates in both sub-systems. The magnitudes should be constant and independent of the motor electrical angle for the ideal motor and power inverter operating in steady-state; however, the sixth harmonic component is visible in both healthy and faulty waveforms. The sixth harmonics component is generated especially by the dead-time effect. Analysed inter-turn fault causes a significant increase of the second harmonic component which is nicely visible in $\alpha\beta$ current magnitude waveform.

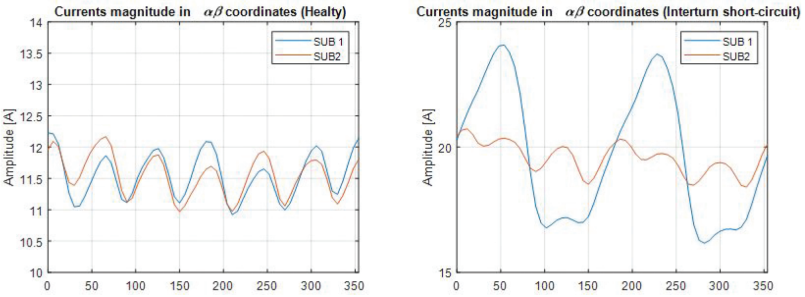

Figure 1.6.5 Filtered data in $\alpha\beta$ coordinates (healthy/with fault).

From the analysis above it is evident that the second harmonic component in filtered currents during one electrical period is a good condition indicator for the inter-turn short circuit fault.

The slight drawback is the fact that the number of measured data during one electrical period depends on motor speed. To suppress this drawback, the whole current waveform is converted to 60 data points per electrical period per sub-system. This fixes the length of the data buffer for its easier processing with ANN.

1.6.2.2.2 Network structure selection

The designed ANN is composed of several ANN modules, and as such, they form a Modular Neural Network (MNN). Filtered magnitudes of current waveforms in both sub-systems are used as inputs into the MNN. To increase fault classification precision, also filtered magnitudes of voltage waveforms are used as inputs. Currents represent the motor torque, while voltages carry the information about the rotational speed. Figure 1.6.6 shows the proposed structure of the MNN. The symmetry of the motor is reflected in the symmetry of data processing in MNN.

1.6.2.2.3 Data pre-processing

Inputs into the MNN consist of four buffers. Each buffer has 60 elements. The buffers are created from actual measured current/voltage magnitudes in both sub-systems.

Used filtration method is based on sixty IIR filters per MNN input. Only one filter with index i is active at a time depending on the actual motor position $\varphi_{e_{(k)}}$ in degrees.

$$i = \text{floor}(\varphi_{e_{(k)}}/60) \qquad (1.6.4)$$

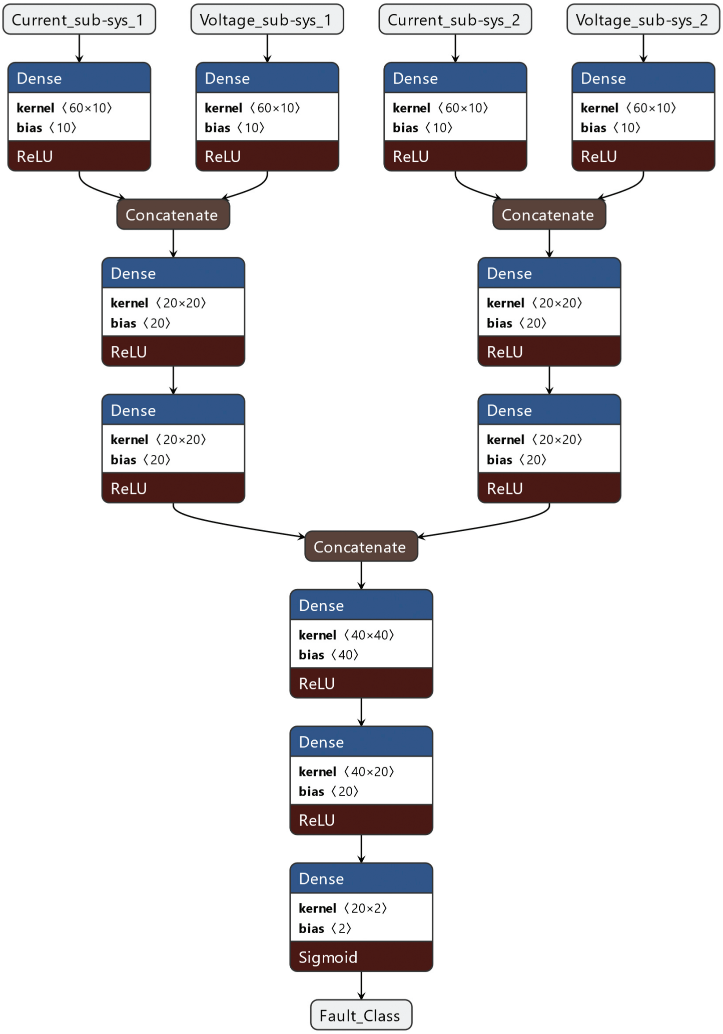

Figure 1.6.6 MNN structure used for inter-turn short circuit detection.

Input data updating occurs once per motor control period which is set to 100 micro us. Filters are described by the following formula:

$$y_{i_{(k)}} = K u_{i_{(k)}} + (1 - K)\, y_{i_{(k-1)}} \tag{1.6.5}$$

where the filtering constant K is set to 0.01, k is step related with the control period of the data generation and $u_{i_{(k)}}$ denotes one of the inputs.

Outputs of the filters are grouped into the buffer. This buffer is used as an input to MNN and it contains filtered signal along one electrical period.

1.6.2.2.4 Preparation of datasets

Fault symptoms that can appear in PMSMs depend not only on the emulated fault type but also on the motor operating point (motor speed, load torque, problematic stator phase). For this reason, a large amount of training data is required to cover all possible fault states in all operating conditions.

Datasets measured on the real motor were obtained under various motor speeds and torques. Randomly generated transients between randomly selected electrical speeds in the range from 200 to 3000 rad/s and with different torques in the range from 0 to 10 Nm (the breaking torques were not used for learning and not for validation) were used to generate training datasets. Simulated data using the motor model were used to generate complementary datasets under the fault condition because these experiments are time-consuming on a real motor. The fault current is high and causes fast local overheating of the motor and it is always necessary to let the motor cool down after such experiments. The experimental motor also does not enable to make the short-circuiting between an arbitrary couple of turns of the coil. And this is the second reason why simulated data are used to prepare missing datasets advantageously.

The motor symmetry was employed to extend datasets with the faults in different motor phases. Phase currents and voltages were re-grouped in different orders to prepare the training and validation data for the six-phase (two times three-phase) motor. This solution helped to prepare additional datasets for learning/testing without the necessity to simulate/experiment with each phase and each sub-system separately. This approach significantly reduced the time needed for the dataset preparation.

1.6.2.2.5 MNN training

The network from Figure 1.6.6 was trained from the mixture of real measurements and the data coming from the simulations using the acausal motor model on a workstation PC in the environment of MATLAB using pre-processed datasets as described in the previous two sections.

1.6.2.2.6 MNN validation

The capability of MNN to diagnose the inter-turn short circuit fault was validated using data from the real motor only. Three turns of the stator winding coil were short-circuited. It represents 3/7 of the stator coil in one slot. Validation datasets were measured on the real motor in a similar way as the ones for training. Data used for training were not used for the validation at the same time. The fault was successfully classified with the probability of 99.92 %. When the fault depth was lower, the fault detectability was slightly reduced.

The fault detection below 200 rad/s is significantly less precise, but the severity of the fault is also lower, and it is usually not harmful for the motor.

1.6.2.2.7 MNN deployment

The designed and trained network was implemented in NVIDIA Jetson Xavier platform using GPU coder in MATLAB. This NVIDIA platform was connected with the inverter controller using Ethernet. The controller sends required voltages and currents. The data pre-processing can run in both, in the controller or in the NVIDIA platform.

After the fault injection into the model simulation, it requires only 1.1 ms for classification with a success ratio of 99.92%. The latency of the Linux running on the NVIDIA platform spans up to 100 μs with the provided JetPack software. Other operating systems designed for hard real-time like RedHawk Linux exist and significantly improve the latency issue.

1.6.3 Artificial Neural Network based Vibration Diagnosis

This section is devoted to the design of the second use case, which is ANN for vibration diagnosis for the bearing state of health monitoring. The first subsection deals in general with the vibration diagnosis of rotating machines. The second subsection analysis AI approaches in vibration diagnosis. The third section presents the developed MLP network.

1.6.3.1 Vibration Diagnosis of Rotating Machines

Vibration diagnosis of rotating machines is a commonly used technique in technical diagnosis and faults identification. Not only typical mechanical failures, such as unbalance, misalignment, gears, and bearings problems can be advantageously diagnosed, but also electrically caused failures may be simply found. Nowadays, electrical faults are diagnosed mainly using electrical quantities measurement, however, mechanical vibrations measurement can be very helpful in the detection of electrically hardly detected faults. On the other hand, vibration-based diagnosis is capable to reveal the faults undetectable by measurement of only electrical quantities. There can be two reasons for this fact:

- Manifestation of a fault in the electrical signal domain is quite weak (e.g., in the initial phase of the fault) and cannot be correctly measured due to small signal to noise ratio, while the vibration signal provides successful information for sufficient detection of the fault.
- Given fault does not have an image in the electrical domain, thus the measurement of mechanically generated signals is helpful for successful fault diagnosis.

Thanks to the aforementioned aspects, vibration diagnosis is a widely used part of the diagnosis and predictive maintenance of rotating machines, including e-machines.

1.6.3.2 AI Approaches in Vibration Diagnosis

The algorithms, statistical procedures, and NNs are usually created, learned, and finally inferred on computers, both standard personal computers, and advanced powerful multicore computers with the support of dedicated graphic cards with multicore graphic processors. Also, specialized dedicated hardware such as the NVIDIA Jetson platform is commonly used thanks to relatively small dimensions and high computational performance compared to standard computers. A typical application area for this kind of hardware is Computing-at-the-Edge (CatE) nodes. Insignificant limitations in available memory and performance, rather typical for small size CatE sensors, shall also be taken into consideration. Finally, implementation of the NNs into small sensors or CatE nodes is a relatively challenging process because of limited resources, mainly available memory, computational performance, power consumption as well as the speed of inference of the algorithm. It is very common, that NN creation, learning, and validation process is done using a powerful computer, NN structure and parameters are exported from the IDE and imported into this small performance device as a functional and successfully learned algorithm. A NN is then executed on the target hardware with no need to learn the overall structure of the network. It is good to mention, that by the small performance device is understood a simple microprocessor with several kilobytes of read/write memory, max. a megabyte of program memory, core frequency of about several hundreds of MHz and typical performance of around 100 DMIPS (Dhrystone Million Instructions Per Second). For comparison, typical Jetson NANO hardware has 128 core GPU, 4GB of internal RAM, and is capable of processing around 1600 FLOPS.

Because the performance of the system, as previously mentioned, can be somehow limited, it is good to reduce the amount of input data by pre-processing procedures. Not only the NN algorithm itself, but also other necessary code needs to be executed inside the processor to ensure the basic functionality of the system (e.g., communication with sensing elements, drawing graphics on display to communicate with the user, peripheral service routine, etc.). Signal pre-processing leads to reduction of the input data and in fact to the reduction of the size and execution time of the AI algorithm. In the vibration diagnosis, two types of extracted features are usually used:

- Time domain features – features calculated from the time signal, mainly statistic parameters like RMS value, standard deviation, kurtosis etc.
- Translated domain features – features calculated from translated domain. Frequency transform, Hilbert transform, Gabor transform, Z-transform, etc., are the most used transforms in the vibration diagnosis. It is good to mention, that not the whole e.g. frequency spectrum is used as an input for the algorithm, but only some particular frequency lines representing possible faults are led to the input of the NN. This brings a significant reduction of the input data and computational complexity of pre-processing algorithms.

1.6.3.3 MLP implementable in device at the edge

As an example of a simple and powerful NN algorithm for bearing faults classification, Multilayer Perceptron (MLP) can be considered. Simple shallow dense NN of a MLP type can be seen in Figure 1.6.7.

The network has one hidden layer and three layers in total (including input and output layer). The number of input neurons is equal to eight, representing eight input time-domain features. The number of output neurons is equal to five, representing five output classes. Therefore, the network is trained to distinguish between five faulty states of the input signal. Training dataset used for this network is represented by publicly available CWRU bearing data centre data. As this dataset is used by many scientists for evaluation of their bearing faults detection algorithms capabilities, accuracy of different neural networks can be found in the literature, e.g. [10], where maximal accuracy of 99,92 % can be found. Dataset, containing data of healthy bearing state and four degrees of bearing outer ring faults, was pre-processed and eight

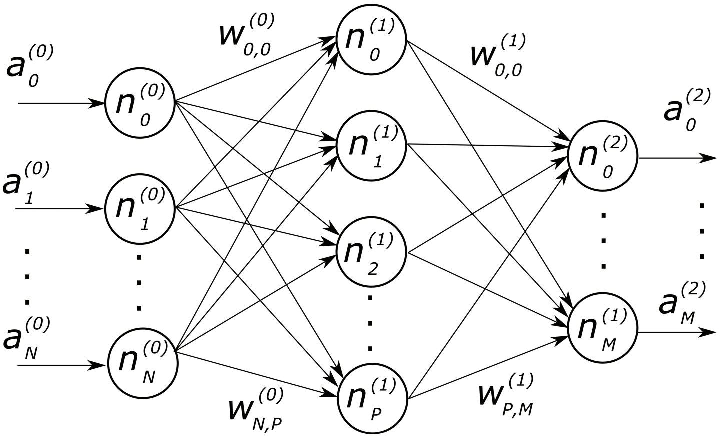

Figure 1.6.7 Shallow dense NN.

signal features have been extracted, namely RMS value, kurtosis, skewness, variance, standard deviation, mean value and min and max value.

Inference algorithm of the network, as well as weights modification procedure using back propagation method, have been implemented in MATLAB environment. The output value of each neuron can be calculated using equation (1.6.6).

$$y = f\left(\sum_{i=1}^{N} w_i x_i\right) \tag{1.6.6}$$

Where $\boldsymbol{w}_i$ is the vector of the individual weights $w_1, w_2, ..., w_N$, $\boldsymbol{x}_i$ is the vector of individual inputs $x_1, x_2, ..., x_N$ of the perceptron, and f(·) is an activation function. In this case, sigmoid activation has been used.

It is necessary to adjust the initial weights values of the NN during the learning procedure. The commonly used approach is based on back propagation algorithm (gradient descend method). The goal is to adjust the weights according to equation (1.6.7)

$$w_j^0(t+1) = w_j^0(t) + \Delta w_j^0 \tag{1.6.7}$$

with the effort to minimize the output error, defined by subtraction between desired (d_j) and real (${a_j}^{(2)}$) output values of the network (1.6.8).

$$E = \frac{1}{2}\sum_j \left(a_j^{(2)} - d_j\right)^2 \tag{1.6.8}$$

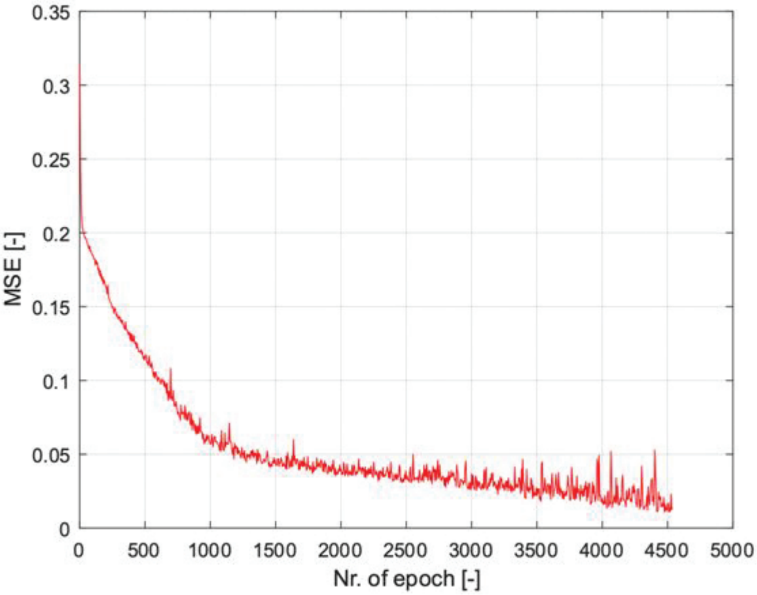

Figure 1.6.8 Mean square error of MLP during training phase.

The equation for final weights modification after partial derivatives of the aforementioned equations and using mathematical operations can be written:

$$\begin{aligned}\Delta w_j^0 = \frac{\partial E}{\partial w_j^0} &= \frac{\partial E_C}{\partial a_j^{(2)}} \cdot \frac{\partial a_j^{(2)}}{\partial z_k} \cdot \frac{\partial z_k}{\partial w_j^0} = \cdots \\ &= \left(a_j^{(2)} - d_j\right) \cdot a_j^{(2)} \left(1 - a_j^{(2)}\right) \cdot a_j^{(0)} \cdot \alpha\end{aligned} \tag{1.6.9}$$

where $\boldsymbol{a}_j^{(2)}$ is the output of the network, $\boldsymbol{a}_j^{(0)}$ is the input of the network, d_j is the desired output and α is the learning rate of the back propagation algorithm.

The final MLP ANN has been implemented in MATLAB and its classification accuracy has been evaluated using Confusion Matrices (CM). Mean square error (MSE) calculated according to Equation (1.6.8) during the learning phase can be observed in Figure 1.6.8.

As it can be seen, MSE reaches very low values (below 2 %) at the end of the learning phase, which has been confirmed by the CM obtained from the output acquired during the testing process. Mentioned CM can be seen in Figure 1.6.9.

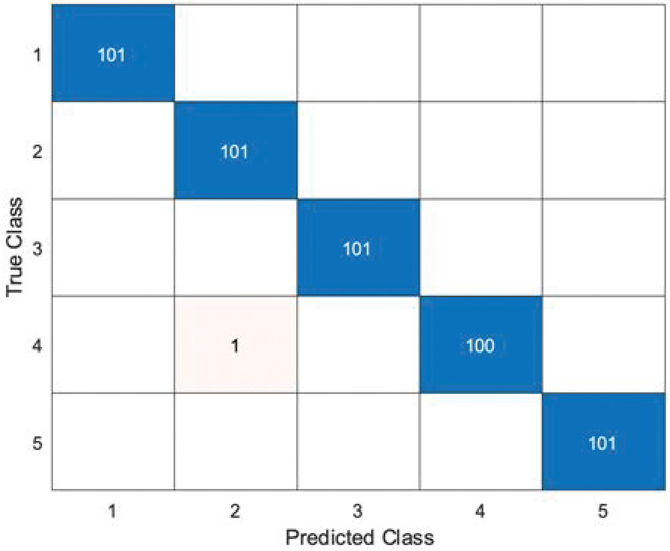

Figure 1.6.9 CM of the testing process of MLP.

Since the MLP was successfully implemented in MATLAB, there is a strong intention to import the network in the low-performance CatE device. Such a device can be represented by a small electronic sensor including a sensing element and microcontroller suitable for MLP inference (e.g. ARM based STM32 microcontroller). Once the structure of the network is created and the weights of the network are established by the training process, a file describing the structure of the network and weights values can be exported from MATLAB and imported by STM Cube. AI application directly into a microcontroller. Validation of the network is done on a PC within testing phase, while validation of the resulting network implementable into STM device is done within Cube.AI software. Afterwards, MLP will fully run inside the target STM device.

To fulfil the requirements of the limited resources of the microcontroller, a simple evaluation of the occupied memory has been done and it is listed in Table 1.6.1 (considering implementation of *float* data type using four bytes).

This amount of total occupied memory of ca. 1.2 kB can be smoothly implemented into small size memory of a microcontroller. Despite the small size of the MLP network, the classification accuracy of the network is satisfactory, as it can be seen in Figure 1.6.9 and the overall algorithm is very well suited for this simple case of bearing faults classification. It is good

Table 1.6.1 CatE device evaluation of occupied memory.

Layer	Memory
Input features	8 x 4 bytes
weights vector (between input and hidden layer)	162 x 4 bytes
weights vector (between hidden and output layer)	90 x 4 bytes
output layer	5 x 4 bytes
intermediate temporary variables	30 x 4 bytes
TOTAL	∼1.200 bytes

to mention, that the accuracy of the classification strongly depends on the learning phase given by the quality and size of the input training dataset.

1.6.4 Conclusion

Two ANNs were designed to detect unexpected behaviour of the e-motor and the bearing on the edge device to operate in real-time. For the inter-turn short circuit detection in the PMSM, MNN was utilized because of the motor symmetry. The highly detailed acausal e-motor model was used to substitute measurement in the operating points which were unreachable on a customized real motor and to reduce the number of required experiments on a real motor. A significant factor in the diagnosis of an inter-turn short circuit fault is the processing time. It was reduced with the used computational hardware to 1.1 ms which is promising and should be sufficient for real-time diagnostic of common e-motors. The second ANN prepared for abnormal vibrations analysis due to bearing faults is MLP designed in a way that the computation is prepared to be deployed directly on the vibration sensor's microcontroller. This is possible since a low-sized and efficient MLP network is applied, which delivers good results in the classification of bearing faults.

Acknowledgements

This work is conducted under the framework of the ECSEL AI4DI "Artificial Intelligence for Digitising Industry" project. The project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826060. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Austria, Czech Republic, Italy, Latvia, Belgium, Lithuania, France, Greece, Finland, Norway. The work was also supported by the infrastructure of RICAIP that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857306 and from Ministry of Education, Youth and Sports under OP RDE grant agreement No CZ.02.1.01/0.0/0.0/17_043/0010085.

References

[1] J. Han, D. Choi, S. Hong and H. Kim: Motor Fault Diagnosis Using CNN Based Deep Learning Algorithm Considering Motor Rotating Speed, 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA), 2019, pp. 440-445, doi: 10.1109/IEA.2019.8714900.

[2] Y. Luo, J. Qiu and C. Shi: Fault Detection of Permanent Magnet Synchronous Motor Based on Deep Learning Method, 2018 21st International Conference on Electrical Machines and Systems (ICEMS), 2018, pp. 699-703, doi: 10.23919/ICEMS.2018.8549129.

[3] S. Wang, J. Bao, S. Li, H. Yan, T. Tang and D. Tang: Research on Interturn Short Circuit Fault Identification Method of PMSM based on Deep Learning, 2019 22nd International Conference on Electrical Machines and Systems (ICEMS), 2019, pp. 1-4, doi: 10.1109/ICEMS.2019.8921744.

[4] J. Qi and H. Wan: A Detection Method of Phase Failure of PMSM Based on Deep Learning, 2020 Chinese Automation Congress (CAC), 2020, pp. 5591-5595, doi: 10.1109/CAC51589.2020.9326708.

[5] F. Husari and J. Seshadrinath: Sensitive Inter-Tum Fault Identifcation in Induction Motors Using Deep Learning Based Methods, 2020 IEEE International Conference on Power Electronics, Smart Grid and Renewable Energy (PESGRE2020), 2020, pp. 1-6, doi: 10.1109/PESGRE45664.2020.9070334.

[6] Y. Chen, S. Liang, W. Li, H. Liang and C. Wang: Faults and Diagnosis Methods of Permanent Magnet Synchronous Motors: A Review. *Appl. Sci.* 2019, *9*, 2116. https://doi.org/10.3390/app9102116

[7] S. Foitzik and M. Doppelbauer: Simulation of Stator Winding Faults with an Analytical Model of a PMSM, 2018 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES), 2018, pp. 1-6, doi: 10.1109/PEDES.2018.8707719.

[8] L. Otava, M. Graf, and L. Buchta: Interior Permanent Magnet Synchronous Motor Stator Winding Fault Modelling, IFAC-PapersOnLine, vol. 48, no. 4, pp. 324–329, 2015, doi: 10.1016/j.ifacol. 2015.07.055.

[9] G. Zurita, V. Sanchez and D. Cabrera: A Review Of Vibration Machine Diagnostics By Using Artificial Intelligence Method, UPB -INVESTIGACIÓN & DESARROLLO, No. 16, Vol. 1: 102 – 114 (2016).

[10] S. Zhang, S. Zhang, B. Wang, and T. G. Habetler, "Machine Learning and Deep Learning Algorithms for Bearing Fault Diagnostics – A Comprehensive Review," Jan. 2019, doi: 10.1109/ACCESS.2020.2972859.

[11] B. Mansouri, H.J. Idrissi and A. Venon: Inter-Turn Short-Circuit Failure of PMSM Indicator based on Kalman Filtering in Operational Behavior. Annual conference of the prognostics and health management society, pp. 1-7, 1 2019, doi: https://doi.org/10.36001/phmconf.2019.v11i1.831