POLITECNICO
MILANO 1863

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING
Master of Science – Automation and Control Engineering

M.Sc. Thesis

Machine Learning-Based Monitoring and Diagnosis of Industrial Devices through Current Measurements

Advisor
Prof. Lorenzo Mario Fagiano

Co-advisor
Dr. Marco Lauricella

Candidate
Laura Manzo – 919913

Academic Year 2020–2021

Acknowledgement

I would like to thank Professor Fagiano and Marco Lauricella for the opportunity they gave me to work on this project and for the constant help and guide.
I also want to thank my parents and sister who fully supported me during these years of university studies.
Finally, I would like to thank all my friends and colleagues for the pleasant and funny moments we have spent together.

Sommario

Questa tesi si occupa della creazione di un algoritmo per rilevare se un dispositivo presenta anomalie o guasti. L'algoritmo riceve in ingresso la misura di corrente assorbita dal dispositivo e l'uscita è lo stato del dispositivo stesso. Esso è ottenuto tramite tecniche di machine learning, in particolare si utilizzano support vector machines e reti neurali feedforward. L'algoritmo si ottiene dopo una fase di training in cui vengono elaborati i dati raccolti sia da dispositivi guasti che integri. Le features sono estratte dalle misure di corrente, sia nel dominio del tempo che in quello della frequenza, e sono proposte anche features che riguardano la correlazione tra misure di corrente e tensione.

Il procedimento descritto è applicato a due casi di studio. Il primo riguarda due diversi modelli di motoriduttori in corrente continua a magneti permanenti e consiste nell'identificazione dei motoriduttori rumorosi. I risultati, ottenuti sia attraverso support vector machines che reti neurali, mostrano che l'algoritmo è in grado di identificare i motoriduttori particolarmente rumorosi, utilizzando solo features nel dominio del tempo.

Il secondo caso di studio riguarda due lavastoviglie. L'obiettivo è quello di identificare i componenti attivi, in particolare se le pale sono in movimento e, nel caso, quale sta ruotando. L'algoritmo proposto a tal fine è ottenuto applicando multi-class support vector machines tramite error-correcting output codes. Per una lavastoviglie l'obiettivo è raggiunto, mentre per l'altra l'algoritmo non è in grado di identificare quale pala si stia muovendo, ma può rilevare solo se ve n'è una attiva.

Abstract

This thesis deals with the creation of an algorithm to detect anomalies or faults in a device. The algorithm input is current measurement and the output is the status of the device. The algorithm is obtained through machine learning techniques, in particular, support vector machines and feedforward neural networks are applied. The algorithm is obtained after a training phase in which data, collected from both faulty and healthy devices, are processed. Features are extracted from current measurement both in time and frequency domain. Also features that deal with the correlation between current and voltage measurements are proposed.
The discussed procedure is applied to two case studies. The first one consists in the identification of noisy DC permanent-magnets gearmotors of two different models. Results, obtained both through support vector machines and neural networks, show that the algorithm is able to detect particularly noisy gearmotors, by using only time domain features.
The second case study is about two dishwashers. The objective is to identify active components, in particular, if spry arms are moving and, in case, which one. The proposed algorithm for this task is obtained applying multiclass support vector machines through error-correcting output codes. The objective is achieved for one dishwasher while, for the other, the algorithm is not able to identify which spry arm is moving but can only detect if one of them is active.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations and goals

The continuous increase in automation in many fields, like manufacturing processes and automotive, involves an increase in electrical devices used as actuators. Due to this trend, not only more sophisticated control systems have been developed, but also diagnosis and monitoring techniques. Faults can occur in any part of a production process or in any component of a more complex device. Their investigation in early stages is particularly useful in order to intervene before the occurrence of more severe faults, which would lead to production delay, economic losses, and sometimes safety issues. The objective of constant monitoring is to allow the stakeholders to carry out well-timed interventions, focused on the diagnosed faults.

To diagnose a fault, an algorithm is needed that receives a measured quantity as input, and outputs the status of the considered device. Machine learning techniques, applied directly to the measured signal, are particularly suitable for this task, when physical considerations do not suffice. Machine learning techniques use data samples and past experience to define an algorithm and optimize its parameters in order to maximise fault diagnosis. The advantage is that a mathematical model of the considered device is not needed. On the contrary, a greater amount of data have to be collected for the optimization task. For instance, current measurements of many motors, both healthy and faulty, have to be processed in order to train a reliable classifier whose

output is the motor status. Afterwards, this classifier is used to predict motor conditions from new current measurements.
The objective of this work is to provide a complete process that allows one to define an algorithm for diagnosis purpose through machine learning techniques. In particular, algorithm input is current measurement while its output is the status of the considered device. This process has been tested on two case studies: identification of noisy DC permanent-magnets gearmotors, and identification of dishwashers active components, which can be the first step for the diagnosis of faulty components.

1.2 State of the art

1.2.1 Gearmotors

This paragraph is focused on gearmotors and their main components: gearbox and motor. In particular, fault types and quantities used to detect them are presented. At the end, algorithms used to detect faults both in motors and in gearboxes are discussed.
Many studies have been conducted on gearboxes to detect mechanical faults. With reference to figure 1.1, the most common faults are [2]:

- cracked and/or worn gear tooth;
- eccentric gear;
- pitting bearing;
- misalignments between gears, rotor and shaft, and between shafts in couplings;
- unbalance element.

Among these, bearing faults are particularly studied since their distribution varies from 40% to 90% from large to small machines [3].
For fault diagnosis, vibration analysis has been studied and it has become a well-established tool for mechanical fault diagnosis, which also allows one to identify the type of fault and its severity [2].
On the other hand, vibration analysis has some drawbacks [4, 5]:

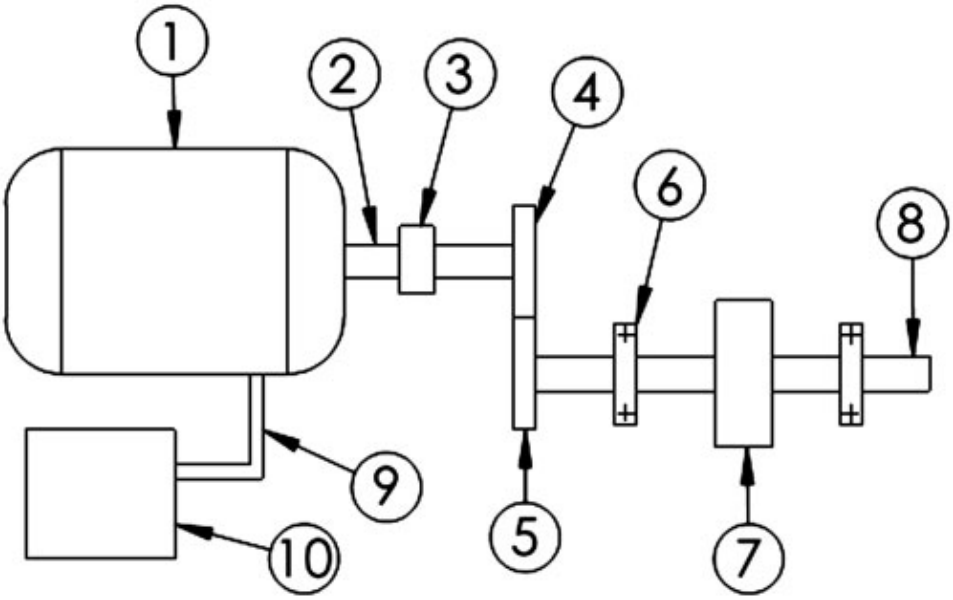


Figure 1.1: Gearmotor model: 1. Motor, 2. Rotor shaft, 3. Coupling, 4. Gear, 5. Gear, 6. Bearings, 7. Mechanical loading system, 8. Driven shaft, 9. Cable, 10. Power supply [1]

- sensitivity to sensor position and external noise;
- technical difficulties of access to the machine, in some cases;
- high cost of accurate sensors.

Therefore, Motor Current Signature Analysis (MCSA) has been introduced to detect gearbox faults by extracting information from motor current measurements. Current is acquired through non-invasive sensors which are often already present for safety reasons or control purpose [3].
Many researchers have focused on MCSA and, in particular, on the case of gearbox connected to induction motor, since they are widely used in industries. This type of diagnosis is usually performed by measuring the stator current at constant speed and load [6], but in the last years, also non-stationary conditions have been studied. This is particularly useful for online monitoring of electrical machines operating at transient speed and load, but it is a difficult task since faults themselves cause load and speed fluctuations. In [7], fault detection through MCSA under transient speed

and no load condition is achieved, whereas in [8], MCSA applied to a scraper conveyor gear under time-varying load and low speed condition has given good results. Finally, in [9], MCSA and vibration analysis have been combined while studying transient load condition.
For what concerns MCSA to detect faults in gearboxes driven by other types of motors, fewer experiments have been carried out. An example is present in [10], in which a hidden Markov model is used in order to predict gear faults in an automotive starter DC motor.
MCSA is widely applied also in electrical machines fault detection where the most common faults are [5, 11–13]:

- static rotor eccentricity, which occurs when the centre of rotation does not coincide with the centre of geometry;
- dynamic rotor eccentricity (also known as bent shaft), which happens when the centre of rotation does not coincide with the centre of mass;
- opening, shortening or abnormal connection of stator windings;
- opening, shortening or abnormal connection of rotor windings;
- broken rotor bar or cracked rotor end-rings in induction motors with squirrel-cage rotor;
- short circuit of commutator bars or their disconnection from coils in brushed DC motors;
- wearing of brushes or insufficient brushes pressure in brushed DC motors;
- demagnetization of permanent magnets, where present;
- misalignments between rotor and shaft;
- bearing fault.

These last two faults are common to studies on gearboxes and on motors.
Also in this case, induction motors are the most studied: for instance, in [14], broken rotor bar, rotor end-rings, and static eccentricity faults are considered, while in [15], stator inter-turn short circuit, cracked rotor bar, and

bearings faults are taken into account. Whereas, a shunt DC motor with rotor eccentricity fault has been considered in [16] and a DC motor is studied in [17] to diagnose bearing faults, misalignment between rotor and shaft, and rotor inter-turn short circuit. In [13], a review of different faults and diagnosis methods is provided for permanent magnet synchronous motors. Brushless DC motors with eccentricity faults and permanent-magnet brushless DC motors with stator inter-turn short circuit are considered in [12] and [18], respectively. Finally, permanent-magnet DC motors with faults in rotor windings, commutator bars, and brushes are diagnosed in [5].

In addition to current, other measured quantities have been proposed for electrical machines diagnosis purpose: acoustic signal [19], temperature [20], and vibrations [21, 22], which are particularly studied. Nevertheless, the analysis of these quantities has the same drawbacks of the ones seen for vibration analysis of gearboxes.

Most of the algorithms used to detect faults both in motors and in gearboxes are based only on signal analysis and do not need to estimate the dynamic model of the motor. Model-based techniques exhibit higher performance, especially in case of continuously varying load and speed, but have higher complexity and usually need also speed measurements in addition to current and voltage ones [5, 23].

One of the first step of any signal-based technique consists in extracting features, which are values derived from measured data that will become the input of the machine learning algorithm. Features can be computed in time domain, frequency domain, or can be obtained through convolutional neural networks [6, 17]. For what concerns learning algorithms, some of the ones used are:

- statistical tests [22];
- Bayes minimum error classifier [16, 19];
- k-nearest neighbour classifier [19];
- k-means clustering [24];
- Support Vector Machines (SVMs) [17, 25];

- decision tree [25];
- random forest [25];
- AdaBoost [25];
- Neural Networks (NNs) [15, 17, 26].

Among these, Bayes minimum error classifier, SVMs, and NNs are the most used. Regarding NNs, different types have been applied: simple feedforward NNs [15], probabilistic NNs [26], and recurrent NNs [17].
Gearmotors noise prediction is much less studied and model-based techniques are adopted. A simulation procedure is presented to predict acoustic noise in a permanent-magnet synchronous motor in [27], and in an axial flux motor in [28]. In [29], noise is predicted through field reconstruction method in a permanent-magnet synchronous motor. Finally, in [30], two different approaches are compared but in both cases model parameters of the considered induction motor have to be computed.

1.2.2 Dishwashers

No studies focused on dishwashers current analysis have been found in literature. Few researches, instead, uses power measurements in order to evaluate a specific component employed during drying phase [31, 32], but further examples of this type of analysis on dishwashers components cannot be found, even though it is probably carried out inside companies.
On the contrary, Non-Intrusive Load Monitoring (NILM) is implemented for all appliances, including dishwashers [33, 34]. NILM techniques try to identify which electrical devices are being used inside a building by measuring the total current and voltage absorptions. These quantities are often used to compute active and reactive power, in order to identify the most probable combination of active devices inside the building. In addition to power, also RMS current, power factor, and voltage-current trajectory are used. Each device is often modelled through a finite state machine: devices, like lights, that can be only on or off, correspond to machines with two states. Instead, a dishwasher can be modelled with more states depending on which components (pump, heating resistance) are active. For each state of the finite

states machine, the power absorption, for instance, is computed and it is subsequently used to disaggregate power contributions of different devices (figure 1.2).
This last step is achieved through different learning algorithms, among which the most applied are:

- neural networks [35];
- support vector machines [36];
- k-nearest neighbour [37].

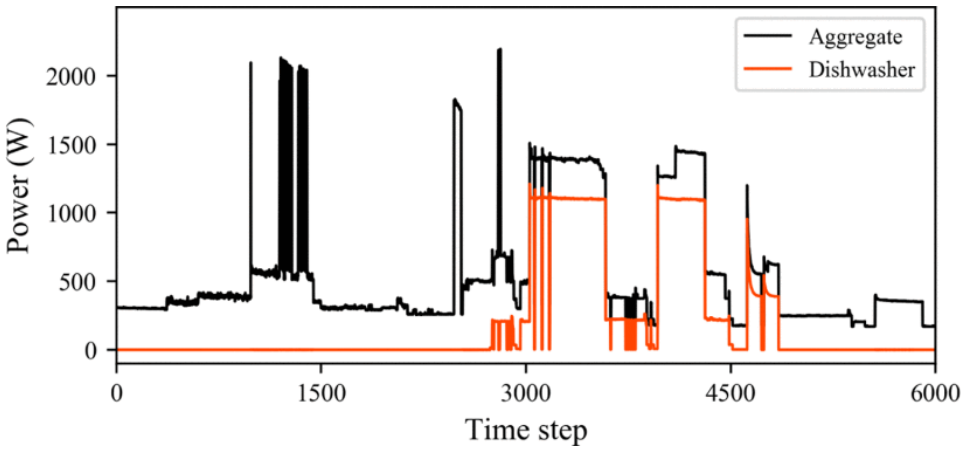


Figure 1.2: Power consumption signal of a dishwasher and the aggregate consumption profile of more devices [38]

1.3 Contributions of this thesis

In this thesis, MCSA is applied to DC permanent-magnets gearmotors in order to identify noisy devices through machine learning algorithms, and a similar procedure is used to identify dishwashers phases, which correspond to different active components.
The main contributions of this thesis are:

- an experimental campaign to collect voltage and current data from gearmotors and dishwashers, and acceleration and acoustic noise measurements from gearmotors only (chapter 2);

- the preprocessing of current measurements, in some cases, in combination with voltage ones, in order to be employed in the training and validation process (chapter 3);

- application of the learning algorithms, in particular support vector machines and feedforward neural networks, in order to identify noisy gearmotors and dishwashers phases. The basics of the employed algorithms are reviewed as well (chapter 4). Moreover, other algorithms have been tested, such as Hotelling's T^2 distance, but they have been eventually discarded since their accuracy was not satisfactory;

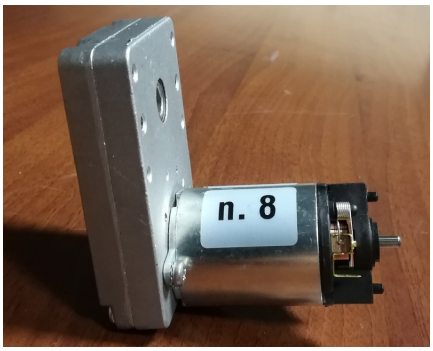
- detailed discussion on all carried out tests and results analysis with assessments of the different approaches (chapter 5).

Chapter 2

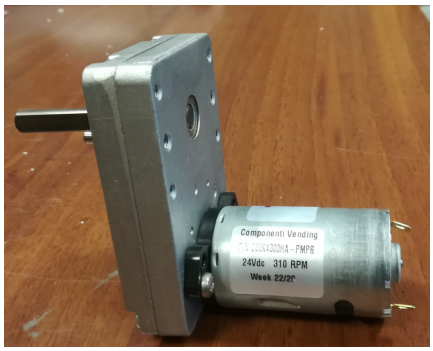
Experimental campaign

2.1 Gearmotors

Objectives of gearmotors analysis are to determine if a device is noisy, considering current measurements, and, possibly, to compute an indicator that gives an idea of the level of noise of a motor compared to the others.
Two models of gearmotors, shown in figure 2.1, have been tested, in particular, model 2 has an additional output shaft. They are both composed of a fixed-axis gearbox and a DC brushed permanent-magnet motor, which operates at nominal voltage of 24 V. Eight motors of the first model and fifteen of the second one have been provided by a medium-sized company located near Milano.



(a) Model 1



(b) Model 2

Figure 2.1: DC gearmotors

	Model 1	Model 2
Nominal voltage V_n	24 V	24 V
Average current at V_n	0.3 A	0.3 A
Speed at V_n	290 RPM	310 RPM

Table 2.1: Gearmotors specifications

During tests, gearmotors are suspended through a twine fixed to a beam, as shown in figure 2.2, and are powered at nominal voltage. They are suspended in order to possibly reproduce the dynamic behaviour of the unconstrained object.

For each motor, three tests have been conducted, in which current, voltage, and acceleration are acquired simultaneously for 2 minutes with sampling frequency of 25600 Hz. Therefore, about 212 million of samples have been collected for each measured quantity. The outputs of current and voltage transducers and of the accelerometer are converted through a multichannel analog-to-digital converter that includes anti-aliasing filter, and it is finally stored on a computer, in a Matlab file, through LabVIEW. Two power supplies are used: one for the motor and the other for the current and voltage transducers, which requires ±15 V.

Figure 2.2: Model 2 gearmotor suspended for testing

Figure 2.3: Noise acquisition through a sound level meter

Acceleration has been acquired for validation purpose: indeed, in a previous work on similar DC permanent-magnets gearmotors, it has been shown that acceleration measurements allow one to identify noisy gearmotors.
Acoustic noise is acquired through a sound level meter. It has been positioned at about 1 meter from the gearmotor and at the same height, as shown in figure 2.3, and only the average value of produced noise is stored.
Each model 1 gearmotor produces the same level of noise every time is powered but this is not valid for model 2 gearmotors. In particular, three of the latter exhibit a particularly noisy behaviour only sometimes: noise average

Motor	Noise (dB)
1	52.5
2	52.5
3	55.5
4	55.5
5	59.0
6	60.5
7	61.0
8	67.0

(a) Model 1 gearmotors

Motor	Noise (dB)
1	56.5
2	59.0
3	60.0
4	60.2
5	60.5
6	60.5
7	61.0
8	61.3
9	61.5
10	62.4
11	63.0
12	63.2
13	63.5
14	63.5
15	64.5
16 (12)	76.3
17 (9)	77.0
18 (7)	78.2

(b) Model 2 gearmotors. Last three rows contain measurements of motors 12, 9, and 7, respectively, when they show a noisy behaviour

Table 2.2: Gearmotors acoustic measurements

is about 62 dB in one case and about 77 dB in the other. This cannot be related to any change in the experimental setup, which remained the same during the experimental campaign. On the other hand, this difference in noise level is related to a difference in current measurements. Therefore, in the following analysis, each of these three motors have been treated as if they were two different motors, one noisy and one non-noisy. Table 2.2 shows acoustic noise measurements for both gearmotor models.

2.2 Dishwashers

The objective of dishwashers analysis is to detect which components are active, through current measurements. In particular, to monitor when either the upper spray arm or the lower one is moving. In this analysis, the heating resistance is not considered, since short cycles, which use only cold water, are taken into account.

Two dishwashers, shown in figure 2.4, are studied. Brand 1 dishwasher has three spray arms: the upper and the middle ones move simultaneously and they alternate with the lower one. On the contrary, brand 2 dishwasher has only two spray arms that alternate.

Current and voltage are acquired through the same devices used for gear-

(a) Brand 1 **(b)** Brand 2

Figure 2.4: Dishwashers

motors with sampling frequency of 5000 Hz. Seven measurements have been acquired from brand 1 dishwasher and eight from brand 2. The considered washing cycles last about 15 minutes, therefore, about two hours of data are collected for each dishwasher, which correspond with about 40 million of samples for each device.

In order to validate the analysis, a camera and a torch are used to film the inside of the dishwashers. In this way, it is possible to know when spray arms activate.

Figure 2.5: Experimental setup to measure dishwashers current and voltage

Chapter 3

Data preprocessing, training, and validation

The overall process of designing a diagnostic algorithm with machine learning techniques can be divided into different steps:

- experimental campaign for data acquisition;
- outliers removal;
- feature extraction;
- dataset partitioning;
- standardization;
- training of a predictive algorithm through a machine learning techniques;
- validation of the entire procedure.

The first step, the experimental campaign, has been described in the previous chapter and strongly depends on the application. All the other steps are described in next paragraphs, and some learning algorithms for the training phase are further analysed in chapter 4.

3.1 Outliers removal

Outliers are data that markedly deviate from the others and are not representative of the considered quantity, measurement errors fall under this statement. There are many definitions of outliers [39], and their removal can also be unnecessary. An outlier, for instance, is a value whose distance from the dataset mean is greater than three times dataset standard deviation. For slowly varying quantities, it is possible to use the same definition but to compute mean and standard deviation on a moving window, instead of the entire dataset.

In the considered test cases, this step has been omitted, since it is possible to observe through visual inspection that no outliers are present in most of the measurements. Only few motor current measurements make an exception, and one of them is shown in figure 3.1. Also in this case, no outliers removal is applied since they affect only few sequences of the entire dataset.

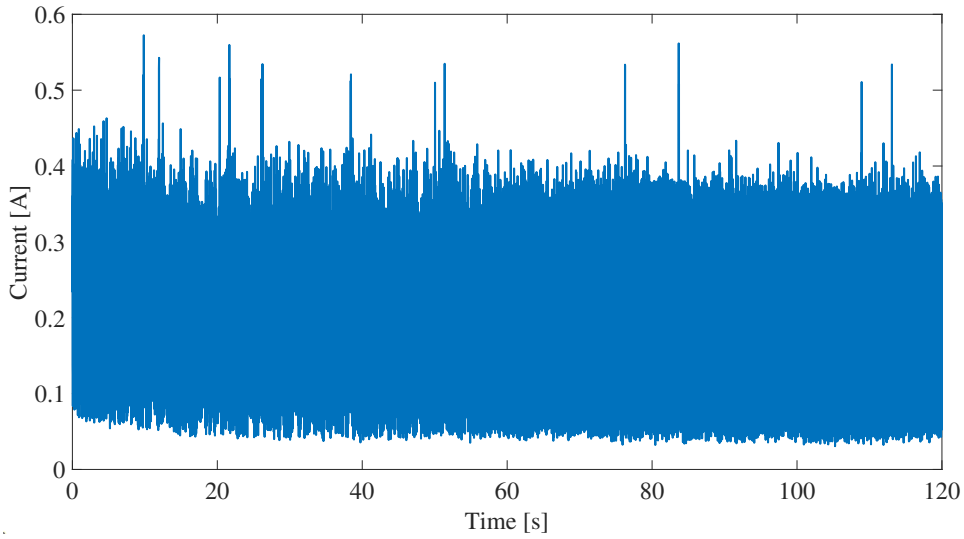


Figure 3.1: Gearmotor current measurement with outliers

On the other hand, some motor current measurements show an initial transient whose maximum duration is about 1 minute, as shown in figure 3.2. Therefore, it has been decided to consider only the second minute of all motor measurements, i. e. when current has reached a steady-state.

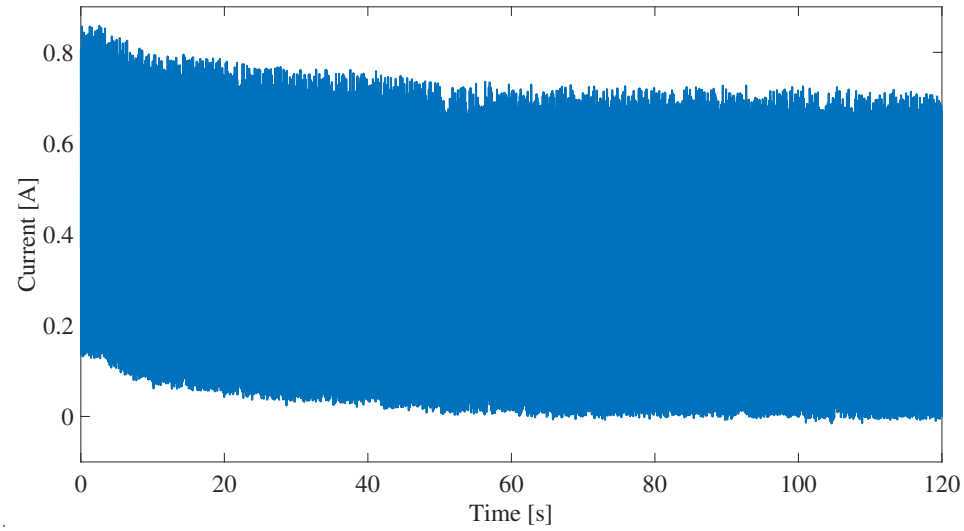


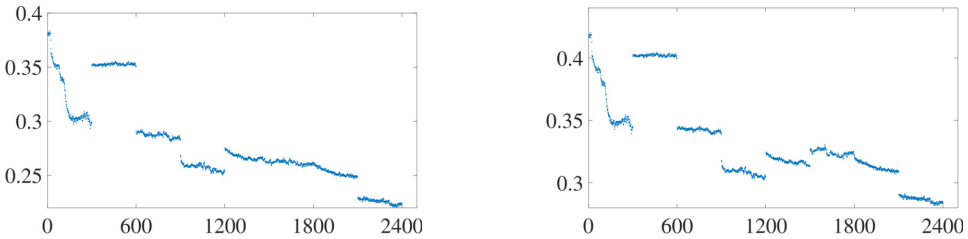
Figure 3.2: Gearmotor current measurement with slow transient

3.2 Feature extraction

Features are values informative of the dataset, obtained by some processing of the raw data. They can be computed directly from the dataset, in time domain, or from its spectral analysis, in frequency domain. They can also represent the relationship between two measured signals, like current and voltage. Finally, there exist automatic techniques to extract features through convolutional neural networks, which is not discussed [17]. The next sections describe the features that have been considered in this work.

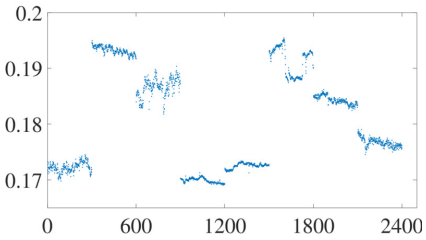
3.2.1 Time domain features

Before computing features in time domain, the dataset is divided into n sequences, and each of them is used to compute all chosen features. At the end, results are rearranged in a features matrix with n rows and m columns, where m is the number of selected features.
Given a sequence x containing N samples x_i, a list of some of the most used features is presented.

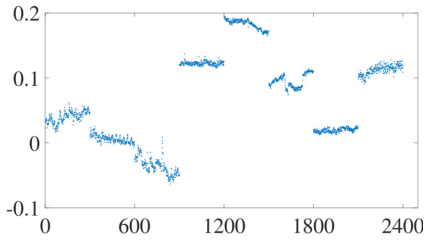


(a) Mean

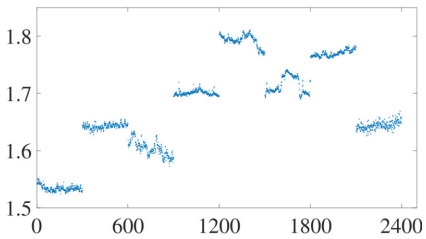
(b) RMS



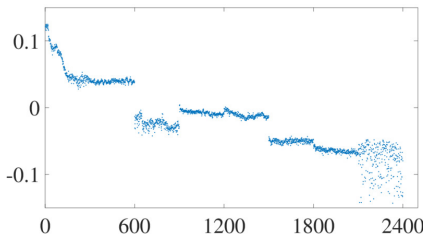
(c) Standard deviation



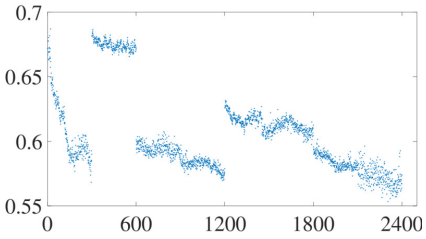
(d) Skewness



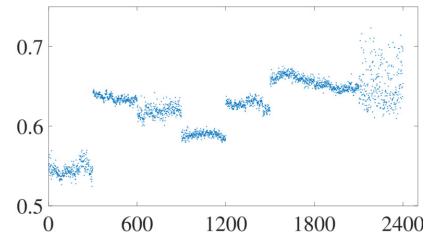
(e) Kurtosis



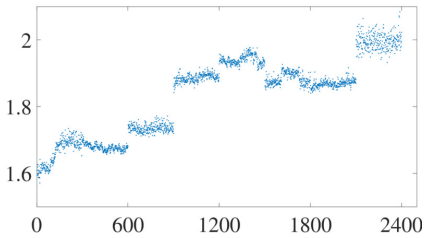
(f) Minimum



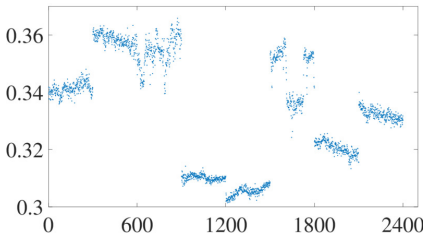
(g) Maximum



(h) Range



(i) Crest factor



(j) Interquartile range

Figure 3.3: Time domain features for model 1 gearmotors, using the first measurements of current as raw data. Data from 1 to 300 refer to motor 1; 301-600: motor 2; 601-900: motor 3; 901-1200: motor 4; 1201-1500: motor 5; 1501-1800: motor 6; 1801-2100: motor 7; 2101-2400: motor 8.

Mean value: $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$

Root mean square: $RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$

Standard deviation: $\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N-1} \left(x_i^2 - \mu\right)}$

Skewness: $s = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^3}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}}$

Kurtosis: $k = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^4}{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2\right)^2}$

Minimum: $min = \min_{\forall x_i \in x} x_i$

Maximum: $max = \max_{\forall x_i \in x} x_i$

Range: $range = max - min$

Crest factor: $C = \frac{max}{RMS}$

Interquartile range: $IQR = Q_3 - Q_1$

Where Q_1 is the first quartile, which is the 25^{th} percentile of the distribution of x, and Q_3 is the third quartile, which is the 75^{th} percentile.

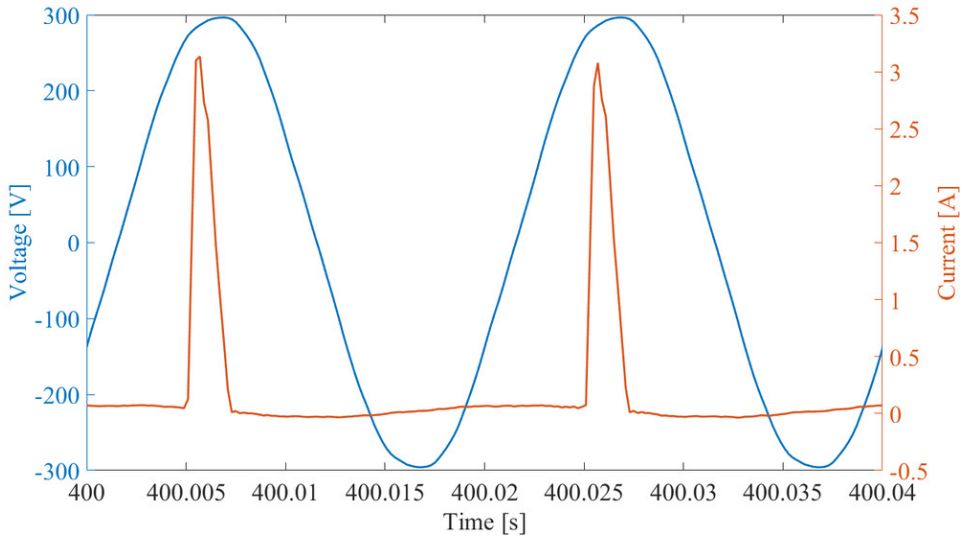
Gearmotors features

In figure 3.3, all listed features have been computed for model 1 motors. For each motor of both types, the second minute of current measurement has been divided into 300 sequences. Therefore, each of them lasts 0.2 s and contains 5120 samples. This value has been chosen after some trials: a higher one would give no benefit, while a lower one would increase the effect of noise.

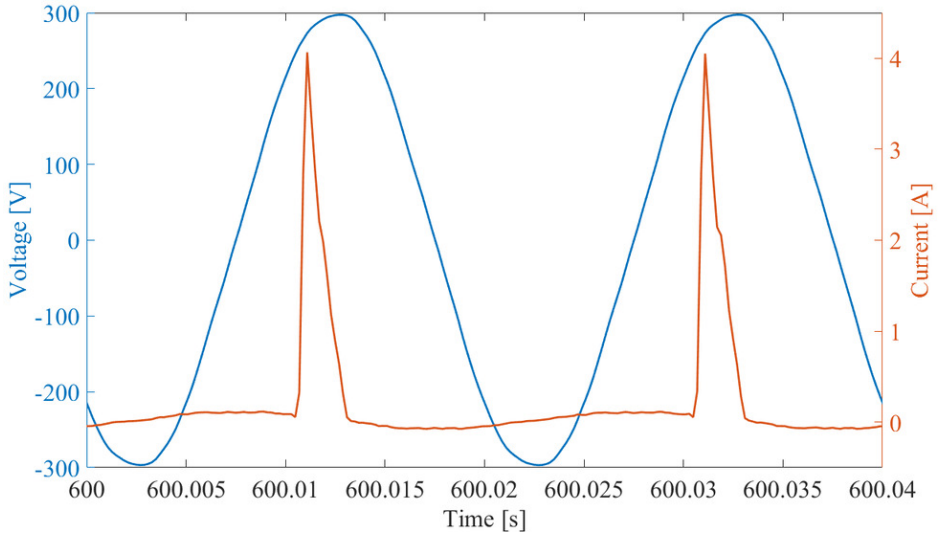
In figure 3.3, it is also possible to notice that mean, minimum, and crest factor show an ascending or descending trend that corresponds to noise level.

Dishwashers features

In dishwashers analysis, it is not convenient to extract features directly from the current signal since it is alternating (figure 3.4).



(a) Brand 1 dishwasher



(b) Brand 2 dishwasher

Figure 3.4: Dishwashers current and voltage measurements when bottom spry arm is moving

For this reason, the RMS current is computed on a moving window and I_{RMS} is used both to automatically separate dishwashers phases and to compute features. Given a window of length w and a vector of current measurement

I of length L, I_{RMS} is computed as

$$I_{RMS}(t) = \sqrt{\frac{1}{w} \sum_{i=t}^{t+w-1} I_i^2} \quad \text{for } 1 < t < L - w + 1$$

It has been chosen w equal to the number of samples in one period of the current. Since the current frequency is $f = 50$ Hz, w can be obtained as

$$w = \frac{f_{sample}}{f} = \frac{5000}{50} = 100$$

Then, features are computed on sequences of I_{RMS}. The length of these sequences is the same of the gearmotors ones: a sequence consists of 5120 samples of I_{RMS}.

3.2.2 Frequency domain features

Features in frequency domain are computed after a transform has been applied.

The most common is discrete Fourier transform, which can be computed through a fast Fourier transform algorithm. Like for features in time domain, it is possible to divide the signal into sequences, but it is worth to consider that the interval between two samples in frequency domain is equal to the inverse of the duration of the sequence. Once the spectrum of the signal is obtained, it is possible to consider specific frequency bands.

In some applications, it is possible to compute the frequencies of interest considering the geometry of tested device, like in [40] where faults in gears are studied. In that case, knowing the number of teeth of each gear and the number of poles of the induction motor connected to the gearbox, the frequency of each gear has been computed. This one depends on the fundamental frequency of the stator current that can be estimated. In this way it is possible to detect faults in specific gear motor assemblies.

For other applications instead, the entire spectrum is divided into frequency bands and, on each of them, a feature is computed. In this way, the number of columns of the features matrix is equal to the number of frequency bands considered. It is possible to compute the same features presented for time domain or to use different ones like the peak-to-mean ratio

$$k_{pm} = \frac{max}{\mu}$$

In gearmotors case study, different features and bandwidths have been tested but no one showed good results, differently from features in time domain. In particular, tested features have been maximum value, mean, and their ratio. They have been computed on different bandwidths and also on a bandwidth, which allows one to have one peak for each band (figure 3.5).

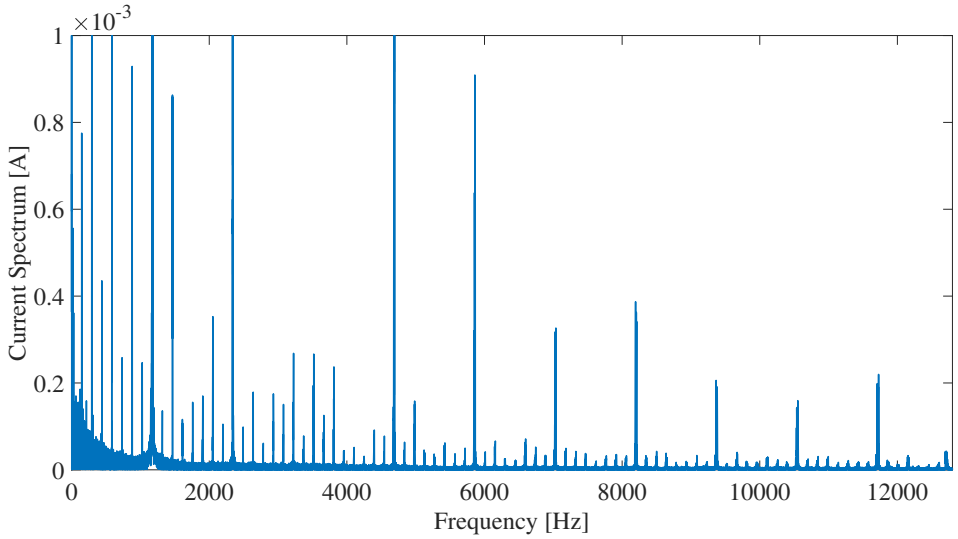


Figure 3.5: Current spectrum of a model 1 motor at steady state obtained through fast Fourier transform.

In addition to discrete Fourier transform, other transforms can be used, like wavelet transforms. Among them discrete wavelet transforms are widely used. In this case, convolutions are applied between the signal and wavelet functions $\psi(t)$. Different wavelet functions have been proposed and scaling factors, a and b, are introduced in order to decompose the signal into more functions with different frequency bands and resolutions [8].

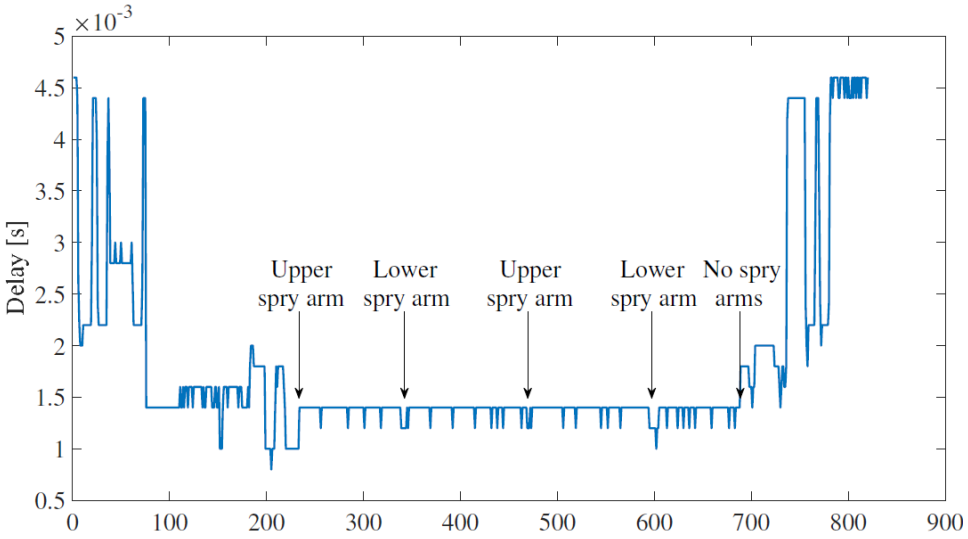
$$[W_\psi f](a,b) = \int_{-\infty}^{\infty} f(t)\,|a|^{-1/2}\,\overline{\psi}\left(\frac{t-b}{a}\right)dt$$

Therefore, wavelet transform preserves information also in time domain and allows one to study frequency domain with a higher resolution, for specific frequency bands [41].

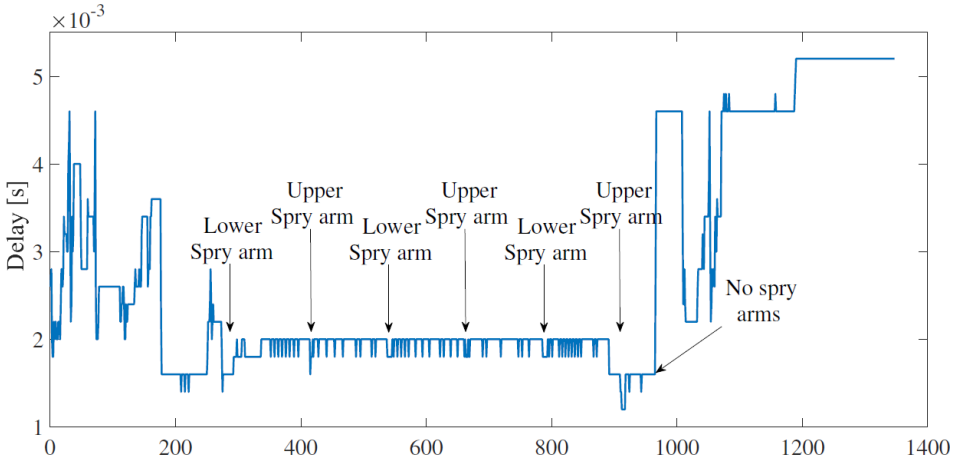
3.2.3 Voltage-current features

If both current and voltage measurements are available, then it is possible to compute the delay between the two signals, and use it as a feature [42]. To obtain the delay, cross-correlation between current I and voltage V is computed as

$$R_{VI\tau} = \sum_{i=1}^{N-\tau} V_i I_{i+\tau} \quad 0 < \tau < T$$



(a) Brand 1 dishwasher



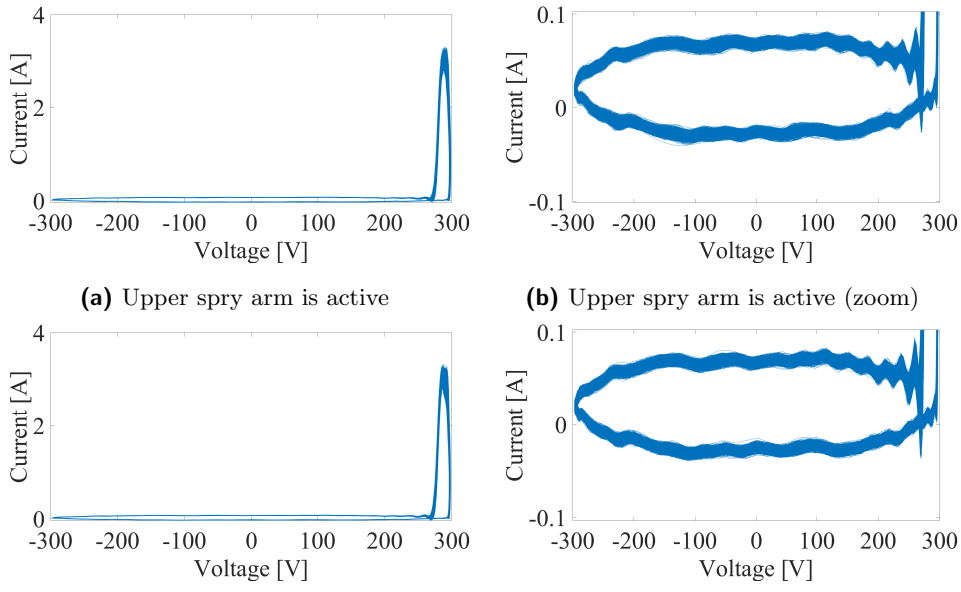
(b) Brand 2 dishwasher

Figure 3.6: Delay between dishwashers current and voltage

where N is the length of vectors I and V, and T is the period of the signals. The delay D is the value of τ that corresponds to the maximum value of cross-correlation. To have it in seconds, it has to be multiplied for the sampling frequency:

$$D = f_{samp} \cdot \arg\max_{\tau} R_{VI\tau}$$

This feature has been computed for dishwashers analysis, dividing current and voltage measurements into sequences of length 5120 samples as for time domain features. Figure 3.4 shows that there is a small difference in phase shift between the two dishwashers and this is coherent with the computed delay (figure 3.6). Anyway, this feature is not used to identify dishwashers phases since there is no change in phase shift when a spry arm activates.
In [43], many features are proposed, which can be extracted from the voltage-current trajectory. They are asymmetry, looping direction, area, curvature of mean line, self-intersection, slope of middle segment, area of left and right segments and peak of middle segment. These features have been tested on brand 1 dishwashers, since for brand 2 time domain features were enough to achieve the phase identification task. However, also these features are



(a) Upper spry arm is active

(b) Upper spry arm is active (zoom)

(c) Lower spry arm is active

(d) Lower spry arm is active (zoom)

Figure 3.7: Brand 1 dishwasher voltage-current trajectories

not used in the following analysis, since there is no correlation between changes in the voltage-current plot and the activation of a dishwasher spry arm (figure 3.7).
In the appendix (figure 6.4), voltage-current trajectories are reported for model 1 gearmotors. They are not used in feature extraction process, since through visual inspection, they do not seem to carry any new information.

3.3 Dataset partitioning

After features have been extracted, dataset have to be partitioned into sets in order to derive the diagnostic algorithm and to test it. The most common choice consists in dividing it into two sets: training set and validation set. Sometimes also a third one, test set, is added. The objective of the partitioning is presented.

- Training set is used by learning algorithm in order to fit parameters for the desired task.

- Validation set is introduced to test the validity of the entire learning procedure and to tune specific parameters called hyperparameters, which will be discussed afterwards. In particular, some metrics can be computed both on the training and on the validation set in order to assess the algorithm.

- Test set is used only if the validation step succeeds in order to evaluate the procedure through an unbiased test, which simulates a real use of the algorithm with a new dataset. Test set is introduced only if the entire dataset is sufficiently large, and usually it has the same dimension of the validation set.

This partitioning is achieved by dividing the features matrix into two or three matrices by extracting specific rows, which can be chosen randomly or with other criteria.
It is worth to notice that, in this analysis, data collected from one motor are used either for training or for validation. Otherwise, if a measurement is split into training and validation sets, metrics computed on the training

set are always equal to the ones computed on the validation set. The same has been done for dishwashers where each measurement is used either for validation or training.
In gearmotors analysis, different choices of dataset partitioning have been tested and they are discussed afterwards, while, for dishwashers, cross-validation is used: it is possible to repeat the dataset-partitioning step more times and, every time, to do again the remaining steps and evaluate the overall procedure. In particular, if leave-one-out cross-validation is adopted, one measurement is used for validation and all the others for training, and this is repeated for all available measurements. This method has been used for dishwashers analysis, where each measurement contains samples from all the possible dishwasher phases. In this way, dataset partitioning does not influence results. This usually allows one to have a relatively large training set and to be able to detect overfitting even if the dataset is small.

3.4 Standardization

Standardization allows one to obtain a dataset with zero mean and unitary standard deviation. Considering a features matrix X, each element x_{ij} becomes:

$$x_{ij}^{(new)} = \frac{x_{ij}^{(old)} - \mu_j}{\sigma_j}$$

Standardization is applied to all datasets, while mean μ_j and standard deviation σ_j are computed considering only the training set, column by column. Standardization is particularly useful if there is a wide difference among ranges of different features. There are two reasons for this: it avoids that a feature unnecessarily gains more importance than the others in the learning phase and, depending on the optimization technique, it can speed up the training phase.

3.5 Training

During training, a predictive algorithm is defined and its parameters are computed. The algorithm receives a features vector as input, while the

output depends on the considered task and learning algorithm.

- In a classification (or clustering) problem, the output is the class to which the input sample belongs, if a classification algorithm is used. Otherwise, if a regression algorithm is applied, the output is a vector containing probabilities of the sample belonging to each class. It is also possible to consider two or more classes, defining a binary classification problem or a multi-class one, respectively.

- In a regression problem, the output is a continuous quantity and a regression algorithm has to be used.

Many learning algorithms have been proposed in literature. They can also be classified into unsupervised learning algorithms and supervised ones, where only in the latter, data have to be previously labelled with the expected output. In addition, the difference between classification and clustering is that the first one uses a supervised algorithm while the second an unsupervised one.
In this thesis, the only unsupervised learning algorithm taken into consideration is k-means clustering, while supervised ones are support vector machines and feedforward neural networks. Also a statistical test, Hotelling's T^2 distance, is used.
Some machine learning algorithms are explained in detail in chapter 4.

3.6 Validation

After the algorithm has been trained, data from both training and validation sets are passed to the algorithm.
In supervised learning algorithms, by comparing the obtained output with the expected one, it is possible to evaluate the procedure. Some metrics are introduced in order to achieve this task by looking at one or few values.
For binary classification with a so called positive class and a negative one, it is possible to compute the number of:

- true positives (TP): samples correctly predicted as belonging to the positive class;

- true negatives (TN): samples correctly predicted as belonging to the negative class;

- false positives (FP): samples actually belonging to the negative class but misclassified;

- false negatives (FN): samples actually belonging to the positive class but misclassified.

		Predicted	
		Positive	Negative
Expected	Positive	TP	FN
	Negative	FP	TN

Table 3.1: Confusion matrix

Afterwards, metrics can be computed, and the objective is to maximise one of them.

- Accuracy: used if both classes have the same importance, and therefore FP and FN have the same weight.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision: used if FP are more undesirable with respect to FN.

$$precision = \frac{TP}{TP + FP}$$

- Recall or sensitivity: complementary to precision.

$$recall = \frac{TP}{TP + FN}$$

- F_β score: precision and recall are combined. If $F_\beta < 1$ then precision is weighted more than recall, if $F_\beta > 1$ then the opposite occurs.

$$F_\beta = \left(1 + \beta^2\right) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}$$

On the other hand, these metrics cannot be used for regression algorithms but others can be computed. Given a dataset with n samples, the output of the algorithm, $\hat{y}_i$, is computed for each sample i. $\hat{y}_i$ is compared with the expected output y_i, through some metrics and the object is to minimize one of them.

Mean squared error: $$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Root mean squared error: $$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Mean absolute error: $$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Finally, for unsupervised learning methods other metrics can be introduced that are not presented here. They usually take into account the distances among samples belonging to the same class and the ones among data from different classes. One of them is the mean of silhouette coefficients [44].
All these metrics have to be computed both for the training and the validation set. By comparing these two values with the expected one, different strategies can be adopted in order to improve the algorithm. For instance, if the chosen metric is accuracy, the following situations can occur:

- training accuracy is higher or about equal to the expected one but validation accuracy is much lower. Therefore, there is an overfitting problem that can be solved by decreasing the algorithm complexity or enlarging the training set;

- training and validation accuracy are about equal but they are lower than the expected one. Underfitting problem occurs, and can be solved by increasing the algorithm complexity;

- training accuracy is lower than the expected one and it is higher than validation. In this case, more data should be collected;

- training, validation, and expected accuracy are about equal, then the objective of the analysis has been reached and, if a test set is available, its accuracy can be computed:

- ◇ if test accuracy is comparable with the other values of accuracy, then the algorithm is complete;
- ◇ otherwise, if test accuracy is much lower, then overfitting occurs and the validation set should be enlarged. This happens because the validation set is used to tune some values, called hyperparameters, which determine algorithm complexity.

An analogous process can be followed for other metrics and tasks.

Chapter 4

Machine learning algorithms

As shown in paragraph 3.5, machine learning algorithms can be chosen depending on:

- availability of algorithm expected outputs;
- desired outputs form;
- task complexity.

In this chapter only Support Vector Machines (SVMs) and Neural Networks (NNs) are discussed, even though other algorithms like k-means and Hotelling T^2 distance have been used. K-means is an unsupervised learning algorithm [45] that has been useful at the beginning to have a better understanding of available data and possible clusterings. Instead, Hotelling T^2 distance is a multivariate statistical test [46], but results obtained through this algorithm are worse than through SVMs and NNs and therefore it is not reported in this chapter.

4.1 Support vector machines

SVMs have been introduced in order to solve binary classification problems, but a similar algorithm, which will be presented in 4.2, has been proposed in order to solve multi-class problems. There exist also a clustering version [47] and a regression one [48] that will not be treated.

4.1.1 Algorithm

SVMs classify data by finding the best hyperplane that separates samples belonging to one class from those belonging to the other. The hyperplane is chosen in order to maximise the margin between the two classes, as shown in figure 4.1. The algorithm is initially presented for binary classification with linearly separable data and then it is extended to non-linearly separable ones.

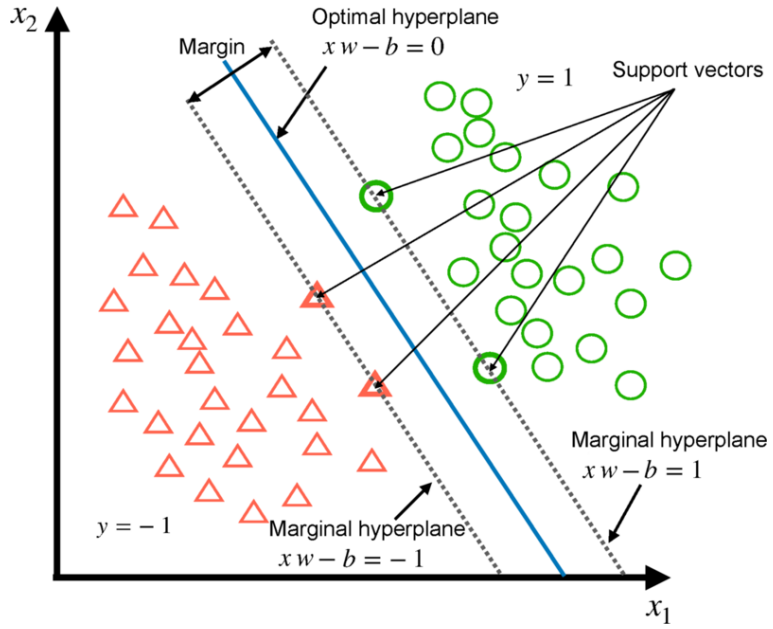


Figure 4.1: Support vector machine with linearly separable data [49]

For each sample i in the dataset, $x_i \in \mathbb{R}^m$ is its features vector and $y_i = \pm 1$ is the label corresponding to its expected class. Considering parameters $w \in \mathbb{R}^m$ and $b \in \mathbb{R}$, where w is a column vector, the separating hyperplane is defined as

$$xw - b = 0$$

The equations of the two parallel hyperplanes defining the margin are

$$P_1 : xw - b = 1$$

$$P_2 : xw - b = -1$$

The following constraints are added in order to have all data, belonging to the positive class, on or above P_1 and the remaining data on or below P_2. Data positioned exactly on one of these hyperplanes are called vector machines.

$$x_i w - b \geq 1 \quad \text{if } y_i = 1$$

$$x_i w - b \leq -1 \quad \text{if } y_i = -1$$

can be rewritten as

$$y_i(x_i w - b) \geq 1 \quad \forall i \tag{4.1}$$

The object of the problem is to maximise the margin, and therefore to maximise the distance between the two hyperplanes P_1 and P_2. This can be computed as the distance d between a point x_0 belonging to P_1 and the hyperplane P_2:

$$d = \frac{|x_0 w - b + 1|}{||w||}$$

By substituting $x_0 w - b = 1$, it is possible to obtain

$$d = \frac{2}{||w||}$$

Therefore, the distance is maximised when $||w||$ is minimised and the entire problem can be rewritten as a quadratic optimization problem with constraints obtained in 4.1 and objective function

$$\min_{w,b} \frac{1}{2} w^T w$$

This problem is extended to the case of non-linearly separable data by adding slack variables ξ_i in order to soften the constraints in 4.1, as shown in figure 4.2.

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} w^T w + C \sum_i \xi_i \\ \text{subject to} \quad & y_i(x_i w - b) \geq 1 - \xi_i \quad \forall i \\ & \xi_i \geq 0 \quad \forall i \end{aligned} \tag{4.2}$$

C is a parameter and, by decreasing it, the weight given to misclassification is reduced. In the following analysis, C is not tuned and it is set equal to 1.

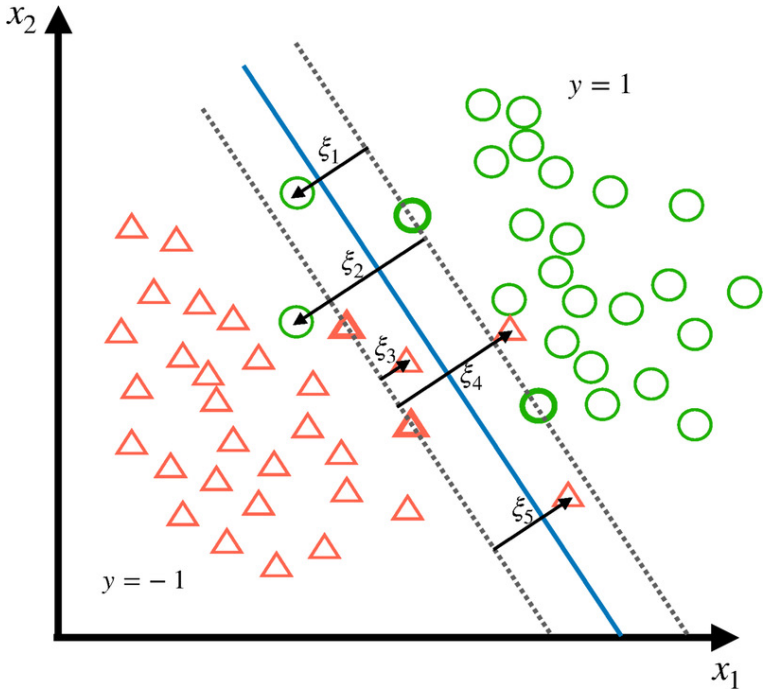


Figure 4.2: Support vector machine with slack variables ξ_i [49]

4.2 is a quadratic optimization problem that can be solved in the dual space introducing Lagrange multipliers α_i, as shown in [50]. The problem becomes

$$\begin{aligned} \max_{\alpha} \quad & \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j^T \\ \text{subject to} \quad & \sum_i y_i \alpha_i = 0 \quad \forall i \\ & 0 \leq \alpha_i \leq C \quad \forall i \end{aligned} \tag{4.3}$$

Another solution, for non-linearly separable data, is obtained by observing that data can be divided by a geometrical object different from a hyperplane. Therefore, it is possible to map data into a different (usually higher-dimensional) space through a non-linear function, as shown in figure 4.3, and then compute the hyperplane that separates data, which brings back to the previous formulation of the problem.

In practice, instead of computing data coordinates in the new space, the kernel trick is adopted: considering a function $\varphi(x_i)$ that maps x_i into an

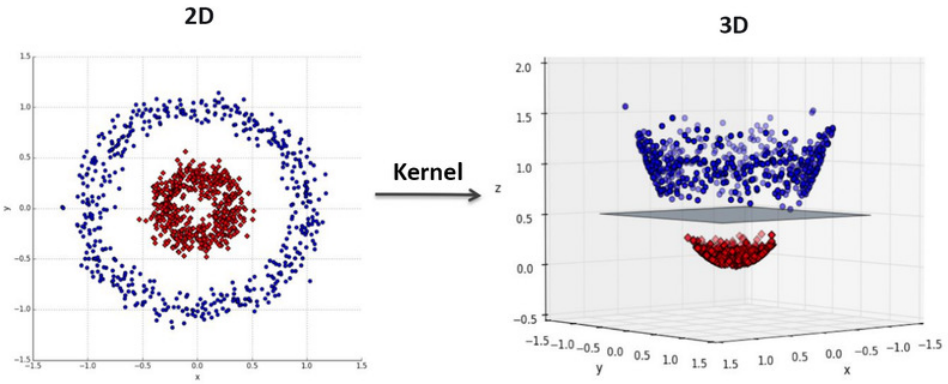


Figure 4.3: Mapping of 2D non-linearly separable data into 3D space through a radial basis function, making them linearly separable [51]

higher dimensional space, and given a kernel function $k()$ which satisfies

$$k\left(x_i, x_j\right) = \varphi(x_i) \cdot \varphi(x_j)$$

it is possible to substitute the scalar product $\varphi(x_i) \cdot \varphi(x_j)$ without computing $\varphi(x_i)$. Therefore, equation 4.3 becomes

$$\max_{\alpha} \quad \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j k\left(x_i, x_j\right) \tag{4.4}$$

The most common kernel functions are:

Polynomial of degree d:	$k\left(x_i, x_j\right) = \left(x_i \cdot x_j + 1\right)^d$
Radial basis function:	$k\left(x_i, x_j\right) = e^{-\gamma \lvert\lvert x_i - x_j \rvert\rvert^2}$
Hyperbolic tangent:	$k\left(x_i, x_j\right) = \tanh(p_1 x_i \cdot x_j + p_2)$
	for some $p_1 > 0$ and $p_2 < 0$

Not all values for p_1 and p_2 are allowed and it has been shown that hyperbolic tangent usually does not perform better than radial basis functions [52], therefore hyperbolic tangent is not used as kernel function in this analysis. In general these two solutions for the problem of non-linearly separable data are combined: data are mapped to a new space through kernel functions and a slack variable is introduced while solving the optimization problem.

It is worth to notice that both solutions introduce parameters that determines the algorithm complexity. These parameters are called hyperparameters and include the value C, the kernel choice, and its parameters like the polynomial degree d. They are tuned by comparing metrics computed on the training set and on the validation one.

4.1.2 Experimental application

SVMs have been applied to classify gearmotors into noisy and non-noisy ones. Tests have been conducted for different features choices, training sets, and kernels.

As shown in figure 3.3, it is convenient to choose mean, minimum, and crest factor as features, at least for linear kernel. Therefore, for each motor a features matrix with 300 rows, one for each sample, and 3 columns is computed. Since no threshold has been given in order to define the two classes, a first test is presented in which only the most noisy motor and the less noisy one have been included in the training set, while all the others belong to the validation set. Having no threshold, no expected classification is present and therefore no metrics can be computed in this specific case. Results are

Motor	Noise (dB)	Predicted class	Scores		
			Average	Minimum	Maximum
1	52.5	1	1.08	1.00	1.17
2	52.5	1	1.04	0.52	2.00
3	55.5	-1	-0.38	-0.53	-0.13
4	55.5	1	0.12	-0.03	0.27
5	59.0	-1	-0.46	-0.59	-0.31
6	60.5	-1	-0.57	-0.70	-0.45
7	61.0	-1	-0.67	-0.87	-0.50
8	67.0	-1	-1.25	-1.70	-1.00

Table 4.1: SVM results for the first measurement of model 1 gearmotors, where the positive class corresponds to non-noisy motors and the negative class to noisy ones. Features, computed in time domain, are: mean value, minimum, and crest factor. Training set: motor 1 (non-noisy), motor 8 (noisy). Kernel is linear.

Motor	Noise (dB)	Predicted class	Scores		
			Average	Minimum	Maximum
1	56.5	1	1.16	1.00	1.33
2	59.0	1	0.35	0.16	0.71
3	60.0	1	1.15	0.86	1.41
4	60.2	1	0.77	0.60	0.95
5	60.5	1	0.94	0.77	1.64
6	60.5	1	0.81	0.65	0.93
7	61.0	1	1.26	0.86	1.66
8	61.3	1	0.71	0.49	1.42
9	61.5	1	1.37	1.10	1.65
10	62.4	1	0.86	0.68	1.14
11	63.0	1	1.03	0.46	1.77
12	63.2	1	1.64	1.40	2.01
13	63.5	1	0.96	0.67	1.43
14	63.5	1	0.66	0.44	0.95
15	64.5	1	0.20	0.08	0.50
16	76.3	-1	-0.51	-0.84	-0.08
17	77.0	-1	-0.42	-0.67	-0.06
18	78.2	-1	-1.40	-1.74	-1.00

Table 4.2: SVM results for the first measurement of model 2 gearmotors, where the positive class corresponds to non-noisy motors and the negative class to noisy ones. Features, computed in time domain, are: mean value, minimum, and crest factor. Training set: motor 1 (non-noisy), motor 18 (noisy). Kernel is linear.

shown in tables 4.1 and 4.2. Predicted class is the mode of class prediction for all samples from the same motor. In order to have a better understanding of the results, score value has been introduced, whose absolute value is the distance between each sample and the separating hyperplane. Score is positive if the sample is located in the space corresponding to the positive class and vice versa.

From results of SVM on model 1 gearmotors, it is possible to see that motors are generally well classified and scores average gives knowledge on motors

noise level.
On the contrary, scores average for model 2 motors does not give much information on noise level. Through this SVM, it is possible to identify motors with the particularly noisy behaviour discussed in chapter 2.

4.2 Multi-class SVMs through error-correcting output codes

In order to extend SVMs to a multi-class problem, it is possible to reduce this one to multiple binary classification problems. This can be achieved through directed acyclic graphs where each node corresponds to a binary SVM [53], as shown in figure 4.4. Otherwise, it is possible to use an Error-Correcting Output Codes (ECOC) algorithm, which is discussed in this chapter. Finally, another option is to define a direct method for training multi-class predictors without introducing multiple binary classification problems: this strategy requires a generalized definition of margin [54].

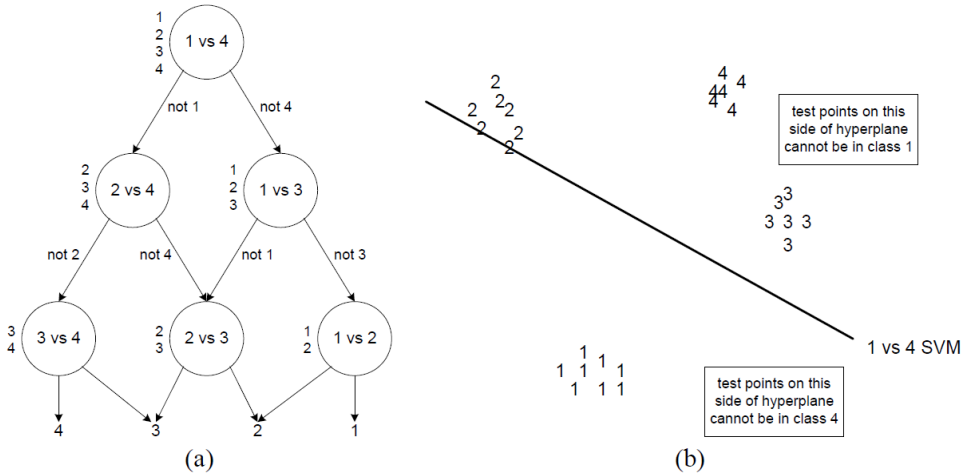


Figure 4.4: (a) The decision graph for finding the best class out of four classes. (b) SVM corresponding to the root node of the graph [53]

4.2.1 Algorithm

In order to use an ECOC algorithm, it is necessary to define a coding design matrix M, which determines the classes that the SVMs train on. For

instance, if the number of classes is $K = 3$, it is possible to define

$$M = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{bmatrix}$$

where each row k represents a class and each column l a SVM, in which the class labelled with 1 is the positive class, label -1 means negative class, and all the others are labelled with 0 and are not considered for that SVM. This coding design is called *one vs one*, since each class is trained against one class at a time and, therefore, columns of M contain only one positive and one negative class. In this way, $M \in \mathbb{R}^{K \times L}$ with $L = \frac{1}{2}K(K-1)$.

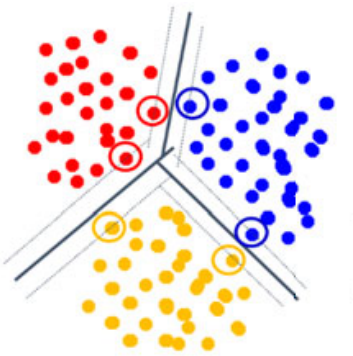


Figure 4.5: Three classes SVM obtained through ECOC and *one vs one* coding design matrix [55]

After all SVMs have been trained, it is possible to assign a new sample to a class through the following steps. First, for each SVM l, the score s_l is computed. Then, a loss function is defined as

$$g(m_{kl}, s_l) = \frac{1}{2} max\{0, 1 - m_{kl} s_l\}$$

where m_{kl} is an element of M. The loss function is

- $g(m_{kl}, s_l) = 0.5$ if $m_{kl} = 0$;
- $g(m_{kl}, s_l) > 0.5$ if m_{kl} and s_l have opposite sign: $m_{k,l} = 1$ but the class predicted by SVM is negative, and vice versa;
- $0 < g(m_{kl}, s_l) < 0.5$ if m_{kl} and s_l have the same sign.

Finally, the predicted class $\hat{k}$ of the overall problem is computed as

$$\hat{k} = \arg\min_k \sum_{l=1}^{L} |m_{kl}| g(m_{kl}, s_l)$$

4.2.2 Experimental application

ECOC has been used to classify dishwashers phases. Three classes have been defined:

- class 1: no spry arm is active;
- class 2: the middle and, if present, the upper spry arms are active;
- class 3: only the lower spry arm is active.

The metric used to evaluate the algorithm is recall. At the beginning, all discussed features in time domain have been used, but recall remarkably increases when only mean value is used. Therefore, the following results are obtained using only the mean value as feature.
As discussed in chapter 3.3, leave-one-out cross-validation is adopted for dishwasher analysis. Results are presented in table 4.3 for brand 1, and in table 4.4 for brand 2.
It is possible to notice that the algorithm works well for brand 2 dishwasher measurements but not for the brand 1 ones.

Validation set	Training recall			Validation recall		
	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
Measure 1	1	0.66	0.79	0.99	0.94	0.91
Measure 2	1	0.77	0.85	0.99	1	0.01
Measure 3	1	0.68	0.81	1	0.97	0.63
Measure 4	1	0.72	0.74	1	1	0.87
Measure 5	1	0.71	0.78	1	0.54	1
Measure 6	1	0.73	0.8	1	0.01	1
Measure 7	1	0.73	0.78	1	0.32	1
Average	1	0.71	0.79	1	0.68	0.77

Table 4.3: ECOC results for brand 1 dishwasher with leave-one-out cross-validation. Each row corresponds to a test with a different validation set. All measurements, but the one used for validation, belong to the training set. Only the mean value is used as feature. Kernels are linear.

Validation set	Training recall			Validation recall		
	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
Measure 1	0.95	1	1	0.97	1	1
Measure 2	0.95	1	1	0.94	1	1
Measure 3	0.95	1	1	0.92	1	1
Measure 4	0.95	1	1	0.96	1	1
Measure 5	0.95	1	1	0.94	1	1
Measure 6	0.95	1	1	0.93	1	1
Measure 7	0.94	1	1	0.99	1	1
Measure 8	0.95	1	1	0.92	1	1
Average	0.95	1	1	0.95	1	1

Table 4.4: ECOC results for brand 2 dishwasher with leave-one-out cross-validation. Each row corresponds to a test with a different validation set. All measurements, but the one used for validation, belong to the training set. Only the mean value is used as feature. Kernels are linear.

4.3 Neural networks

NNs are widely used in many applications and, among the others, they are particularly adopted in image recognition field. Many types of NNs algorithms have been proposed, and in this chapter only feedforward NNs are discussed.

4.3.1 Algorithm

NNs are modelled through directed weighted graphs. In the case of feed-forward NNs, graphs are acyclic: information moves in only one direction from input nodes to output ones (figure 4.6). In addition, hidden layers are present between the inputs and the output layer. All outputs of a layer are inputs for the next layer only.

Considering the layer l, which can be either a hidden layer or the output one, its input is called $x^{[l-1]}$, the output is $x^{[l]}$, and it contains $n^{[l]}$ nodes. Then, $x^{[l]}$ is defined as

$$x^{[l]} = g^{[l]}\left(z^{[l]}\right) = g^{[l]}\left(W^{[l]}x^{[l-1]} + b^{[l]}\right) \tag{4.5}$$

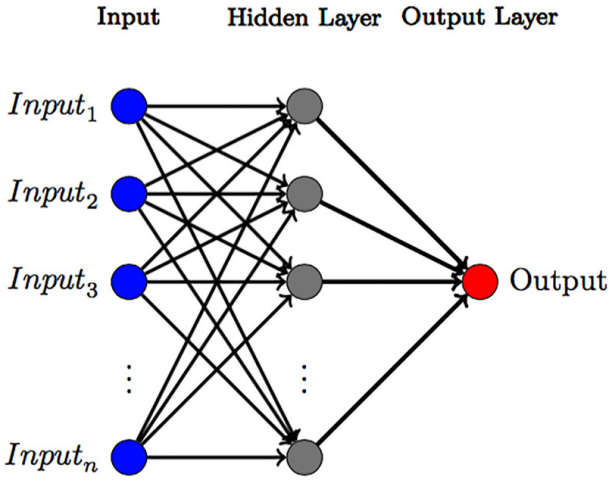


Figure 4.6: Feedforward NN with n inputs, one hidden layer, and one output node [56]

where, for layer l,

- $g^{[l]} : \mathbb{R}^{n^{[l]}} \to \mathbb{R}^{n^{[l]}}$ is the activation function;
- $W^{[l]} \in \mathbb{R}^{n^{[l]} \times n^{[l-1]}}$ is the weight matrix;
- $b^{[l]} \in \mathbb{R}^{n^{[l]}}$ is the bias vector.

The input $x^{[0]}$ of the first layer is one row of the features matrix X, therefore $n^{[0]}$ is equal to the number of features, which is the number of columns of X.
Some activation functions for hidden layers are:

hyperbolic tangent:	$g(z) = tanh(z)$
rectified Linear Unit (ReLU):	$g(z) = \max\{0, z\}$

In this analysis, only ReLU is used since it is more efficient. There exist other activation functions obtained as evolutions of ReLU that can be chosen by considering training and validation metrics [57].
For what concerns activation functions for the output layer, they have to be chosen depending on the considered problem. ReLU can be used also for

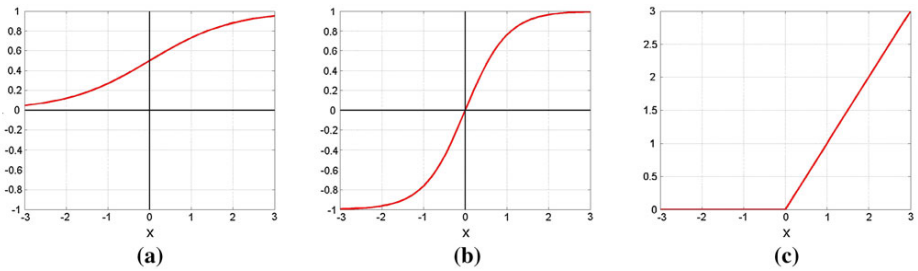


Figure 4.7: Activation functions: (a) sigmoid, (b) hyperbolic tangent, (c) ReLU [58]

the output layer, and other examples of activation functions are:

sigmoid function: $g(z) = \dfrac{1}{1+e^{-z}}$

linear function: $g(z) = z$

softmax function: $g(z) = \dfrac{e^{z}}{\sum_{k=1}^{K} e^{z_k}}$ where z_k is an element of $z \in \mathbb{R}^K$

Sigmoid function is used for binary classification. Linear function and ReLU are chosen for regression problems, where the latter constraints the output to be positive. Finally, softmax function is used in multi-class problems, to output the probabilities of a sample belonging to each of the K classes.
It is worth to notice that the output of this last layer is used to compute the cost function, as explained afterwards, but it is not always equal to the predicted output $\hat{y}$ of the algorithm. Indeed, in binary classification, for instance, it is necessary to introduce a threshold to have $\hat{y} = \{0, 1\}$:

$$\hat{y} = \begin{cases} 0 & \text{if } x^{[L]} < 0.5 \\ 1 & \text{if } x^{[L]} > 0.5 \end{cases}$$

where $x^{[L]}$ is the output of the last layer with sigmoid activation function. Analogously, for multi-class problems, it is possible to have $\hat{y}$ corresponding to the label of the most probable class.
In order to compute weight matrices $W^{[l]}$ and bias vectors $b^{[l]}$, it is necessary to define a cost function to be minimized. Given the expected output y, the most used cost functions are:

- binary cross entropy for binary classification problems

$$C(y, x^{[L]}) = -y \log x^{[L]} - (1-y) \log \left(1 - x^{[L]}\right)$$

which, for instance, tends to infinite if $y = 1$ and $x^{[L]}$ tends to 0.

- mean squared error for regression problems

$$C(y, x^{[L]}) = \left(y - x^{[L]}\right)^2$$

- categorical cross entropy for multi-class problems

$$C(y, x^{[L]}) = -\sum_{k=1}^{K} y_k \log x_k^{[L]}$$

where y_k is an element of the vector y with all zeros entries except a unitary one corresponding to the expected class.

Once a cost function has been chosen, the objective is to find $W^{[l]}$ and $b^{[l]}$ that minimize it. To do so, $\frac{\partial C}{\partial W^{[l]}}$ and $\frac{\partial C}{\partial b^{[l]}}$ have to be computed.
For simplicity, the equation 4.5 is rewritten as

$$x_i^{[l]} = g^{[l]}\left(z_i^{[l]}\right) = g^{[l]}\left(\sum_j w_{ij}^{[l]} x_j^{[l-1]} + b_i^{[l]}\right) \tag{4.6}$$

Considering the equation 4.6, and one of the cost functions already defined, it is possible to compute the derivative of the cost function with respect to $w_{ij}^{[l]}$ $\forall i, j, l$ as

$$\frac{\partial C}{\partial w_{ij}^{[l]}} = \frac{\partial C}{\partial x_i^{[l]}} \frac{\partial x_i^{[l]}}{\partial z_i^{[l]}} \frac{\partial z_i^{[l]}}{\partial w_{ij}^{[l]}} \tag{4.7}$$

Introducing the quantity $\delta_i^{[l]}$

$$\delta_i^{[l]} = \frac{\partial C}{\partial x_i^{[l]}} \frac{\partial x_i^{[l]}}{\partial z_i^{[l]}} \tag{4.8}$$

the equation 4.7 can be rewritten as

$$\frac{\partial C}{\partial w_{ij}^{[l]}} = \delta_i^{[l]} x_j^{[l-1]} \tag{4.9}$$

since

$$\frac{\partial z_i^{[l]}}{\partial w_{ij}^{[l]}} = x_j^{[l-1]}$$

Analogously, the derivative of the cost function with respect to $b_i^{[l]}$ is

$$\frac{\partial C}{\partial b_i^{[l]}} = \delta_i^{[l]} \tag{4.10}$$

since

$$\frac{\partial z_i^{[l]}}{\partial b_i^{[l]}} = 1$$

Then, it is necessary to compute $\delta_i^{[l]}$. For the last layer, it can be obtained directly from 4.8. For the previous layer, it is

$$\delta_i^{[L-1]} = \sum_{j=1}^{n^{[L]}} \frac{\partial C}{\partial x_j^{[L]}} \frac{\partial x_j^{[L]}}{\partial z_j^{[L]}} \frac{\partial z_j^{[L]}}{\partial x_i^{[L-1]}} \frac{\partial x_i^{[L-1]}}{\partial z_i^{[L-1]}}$$

since $C = C\left(y, x_1^{[L]}, x_2^{[L]}, ..., x_{n^{[L]}}^{[L]}\right)$ where y is known and fixed.
By iterating, $\delta_i^{[l]}$ is computed with respect to the following layer as

$$\delta_i^{[l]} = \sum_{j=1}^{n^{[l+1]}} \frac{\partial C}{\partial x_j^{[l+1]}} \frac{\partial x_j^{[l+1]}}{\partial z_j^{[l+1]}} \frac{\partial z_j^{[l+1]}}{\partial x_i^{[l]}} \frac{\partial x_i^{[l]}}{\partial z_i^{[l]}}$$

This equation can be rewritten as

$$\delta_i^{[l]} = \sum_j \delta_j^{[l+1]} w_{ji}^{[l+1]} \frac{\partial x_i^{[l]}}{\partial z_i^{[l]}} \tag{4.11}$$

by considering equation 4.8 and

$$\frac{\partial z_j^{[l+1]}}{\partial x_i^{[l]}} = w_{ji}^{[l+1]}$$

Summing up, the overall procedure to compute $\frac{\partial C}{\partial W^{[l]}}$ and $\frac{\partial C}{\partial b^{[l]}}$ is:

- consider a row $x^{[0]}$ of the features matrix and its expected value y;
- apply 4.5 for each layer from 1 to L and, at each step, store the values $x^{[l]}$ and $z^{[l]}$;
- considering the expected value y, and $x^{[L]}$, compute $\frac{\partial C}{\partial x^{[L]}}$;
- compute $\delta^{[l]}$ for each layer from L to 1, by applying 4.11 $\forall i$.

- compute $\frac{\partial C}{\partial W^{[l]}}$ and $\frac{\partial C}{\partial b^{[l]}}$ by applying 4.9 and 4.10 $\forall l, i, j$.

The first part of this process, until $x^{[L]}$ is computed, is called forward propagation, while the last part is referred to as backpropagation, since $\frac{\partial C}{\partial W^{[l]}}$ and $\frac{\partial C}{\partial b^{[l]}}$ are computed until layer 1.

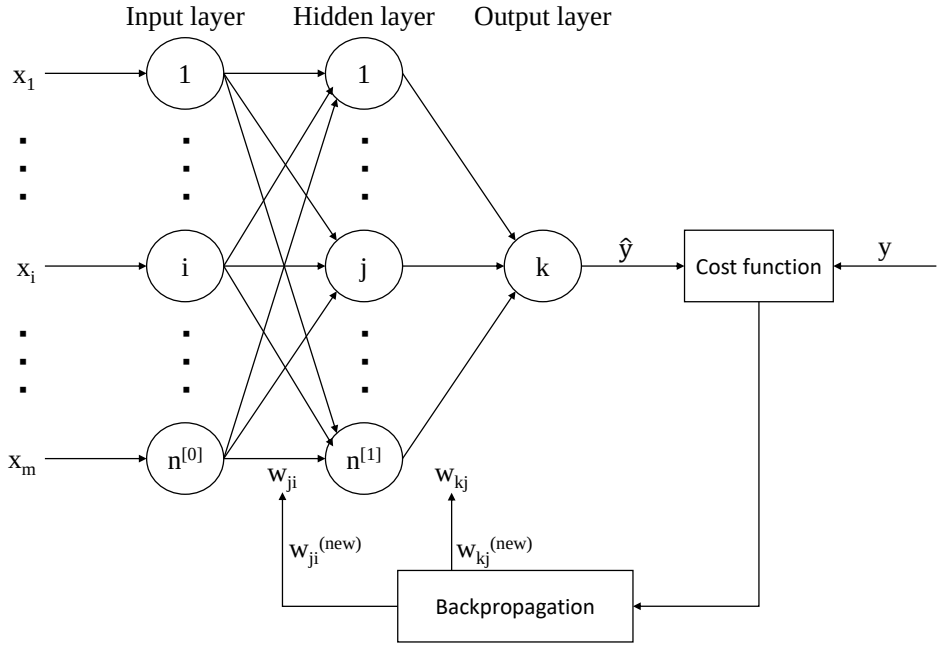


Figure 4.8: NN backpropagation [59]

Then, one possibility is to update $W^{[l]}$ and $b^{[l]}$ through gradient descent:

$$W^{[l](new)} = W^{[l](old)} - \eta \frac{\partial C}{\partial W^{[l]}}$$

$$b^{[l](new)} = b^{[l](old)} - \eta \frac{\partial C}{\partial b^{[l]}}$$

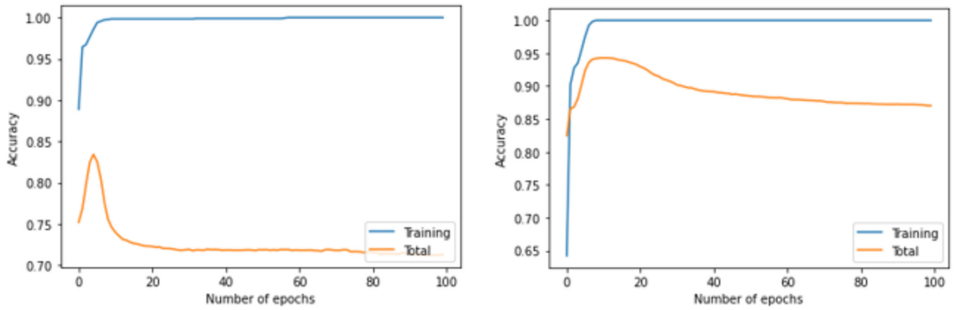
where η is the learning rate.

In this analysis, instead, adaptive moment estimation (Adam) [60] is applied, which is one of the most used optimization algorithms in NNs.

All data in the training set are used more times in order to compute weights and bias: every *epoch* all data are passed once to the NN.

To complete the procedure, it is necessary to initialize $W^{[l]}$ and $b^{[l]}$, and to define a stopping criterion for the algorithm. $W^{[l]}$ are randomly initialized,

while $b^{[l]}$ are zero vectors for all layers.
The simplest stopping criterion limits the number of epochs. Another one, which can be add to the previous criterion, consists in computing the mean of the cost function evaluated on the entire validation set. If it does not decrease for a certain number of epoch, then the algorithm stops. Otherwise, instead of the cost function, it is possible to compute one of the metrics proposed in chapter 3.6.
It is important to have a stopping criterion, since an overtrained NN is subject to overfitting, as shown in figure 4.9. There exist also other methods to avoid overtraining, which are not discussed here, like L_2 regularization and dropout regularization [61].



(a) Training set: motors 1, 4, 8 (non-noisy), and 9, 13, 18 (noisy)

(b) Training set: motors 1, 6, 12 (non-noisy), and 13, 15, 18 (noisy)

Figure 4.9: Accuracy of training set and both training and validation set, computed at the end of each epoch. It has been obtained through NNs for model 2 gearmotors with different training set

In NNs, overfitting is prevented by avoiding overtraining and also by upper binding the values of hyperparameters. These ones are the number of layers, and number of nodes for each layer.

4.3.2 Experimental application

Feedforward NNs have been applied to gearmotors classification problem. All time domain features presented in chapter 3.2.1 have been computed. Thresholds that divide motors into noisy and non-noisy ones have been defined. In particular, more NNs have been trained, which correspond to

different expected classifications.
For the training set of model 2 motors, three noisy motors and three non-noisy ones have been used: for each class, the most noisy motor, the less noisy, and one in the middle have been chosen for training. All the other motors belong to the validation set.
Two stopping criteria have been used:

- number of epochs equal to 100, or
- training accuracy greater than 0.99.

For the stopping criterion, accuracy is computed on training set and not on the validation set in order to use the less possible the validation set, since no test set is available. Figure 4.9 shows that the one proposed is a good stopping criterion.
After some trials, it has been chosen a NN with only one layer having 6 nodes: indeed, deeper NNs were subject to overfitting. Activation functions for the hidden layer are ReLU, and sigmoid for the output one. The cost function is binary cross entropy. 4.5 and 4.6.

Motor	Noise (dB)	Expected class	Predicted class	Accuracy
1	56.5	1	1	1
2	59.0	1	-1	0.48
3	60.0	1	-1	0.09
4	60.2	1	1	1
5	60.5	1	1	0.99
6	60.5	1	1	0.97
7	61.0	1	1	1
8	61.3	1	1	0.99
9	61.5	-1	-1	0.97
10	62.4	-1	-1	1
11	63.0	-1	1	0.02
12	63.2	-1	-1	1
13	63.5	-1	-1	1
14	63.5	-1	1	0.44
15	64.5	-1	-1	0.9
16	76.3	-1	-1	1
17	77.0	-1	-1	1
18	78.2	-1	-1	1

Training accuracy	Validation accuracy
0.99	0.74

Table 4.5: NN results for model 2 motors with training set composed of motors 1, 4, 8 (non-noisy), and 9, 13, 18 (noisy)

Motor	Noise (dB)	Expected class	Predicted class	Accuracy
1	56.5	1	1	1
2	59.0	1	-1	0
3	60.0	1	-1	0.01
4	60.2	1	-1	0
5	60.5	1	1	1
6	60.5	1	-1	0
7	61.0	1	1	0.92
8	61.3	1	1	1
9	61.5	1	1	1
10	62.4	1	1	1
11	63.0	-1	-1	1
12	63.2	-1	1	0
13	63.5	-1	-1	0.85
14	63.5	-1	-1	1
15	64.5	-1	-1	1
16	76.3	-1	-1	1
17	77.0	-1	-1	1
18	78.2	-1	-1	1

Training accuracy	Validation accuracy
1	0.57

Table 4.6: NN results for model 2 motors with training set composed of motors 1, 5, 10 (non-noisy), and 11, 14, 18 (noisy)

Chapter 5

Experimental results

5.1 Identification of noisy gearmotors

Gearmotors current, voltage and acceleration are acquired three times for 2 minutes with sampling frequency of 25600 Hz (figures 6.1, 6.2, 6.3). Also acoustic noise is measured for each motor and its average value is stored, as shown in table 2.2.
Some procedures to remove outliers have been tested but at the and they have not been applied. On the contrary, to have current measurements at steady state, only the second minute, of each of the three measurements, is considered.
Ten features in time domain are computed on sequences of 5120 samples (0.2 seconds). They are mean value, RMS, standard deviation, skewness, kurtosis, minimum, maximum, range, crest factor, and interquartile range. Therefore, features matrix has 10 columns and 900 rows for each of the 26 motors.
The spectrum of the current is computed through fast Fourier transform, and, in particular, three metrics are considered to extract information, which are mean value, maximum, and their ratio. The geometry of gearboxes is unknown, which does not allow one to focus the analysis on specific frequencies. Therefore features are initially computed on bandwidths of 50 Hz or its multiples until 400 Hz, then on bandwidths that allow one to have one peak for each frequency band: peaks distance is about 146 Hz for both gearmotor

models, therefore the first band is between 0 and 73 Hz while the others are between $73 + 146k$ and $73 + 146(k + 1)$ Hz.
Some tests are conducted through discrete wavelet transforms, but wavelet coefficients do not improve noisy motors identification.
Finally, voltage-current trajectories are plotted in order to find some metrics to be used as features, but it is possible to notice through visual inspection that they do not bring new information (figure 6.4).
When SVMs or NNs are applied, all features are standardized considering mean and standard deviation of the training set. Also with k-means, the dataset is standardized but mean and standard deviation are computed on the overall dataset, since the algorithm is unsupervised and therefore no training set is defined. If Hotelling's T^2 distance is used, it is not necessary to standardize the dataset.
Some of the results obtained with k-means are reported in table 6.1. This algorithm is not used as a strategy in order to obtain a good classification of gearmotor, but only to have a better understanding of computed features and how gearmotors tend to be clustered into sets without considering acoustic noise measurements.
The first algorithm used for classification is Hotelling's T^2 distance. For this test, it is necessary to define a training set, composed by samples from non-noisy motors, and a significance level α. Then, it is possible to compute T^2 distances and a threshold T^2_α: a value of T^2 higher than T^2_α corresponds to noisy motors while a lower one corresponds to non-noisy motors. Many tests are conducted through Hotelling's T^2 distance with different training sets, features, and windows for features computation. Tested features are in time and frequency domain, obtained both through Fourier and wavelets transforms. Figure 6.5 shows one of these tests but it is possible to see that it fails in classifying motors.
The second algorithm adopted for classification is SVMs. Tests have been conducted on model 1 motors with many combinations of kernels, dataset partitioning, and features choices, both in time and frequency domain. In particular, considered kernels are:

- linear function;

- radial basis function;

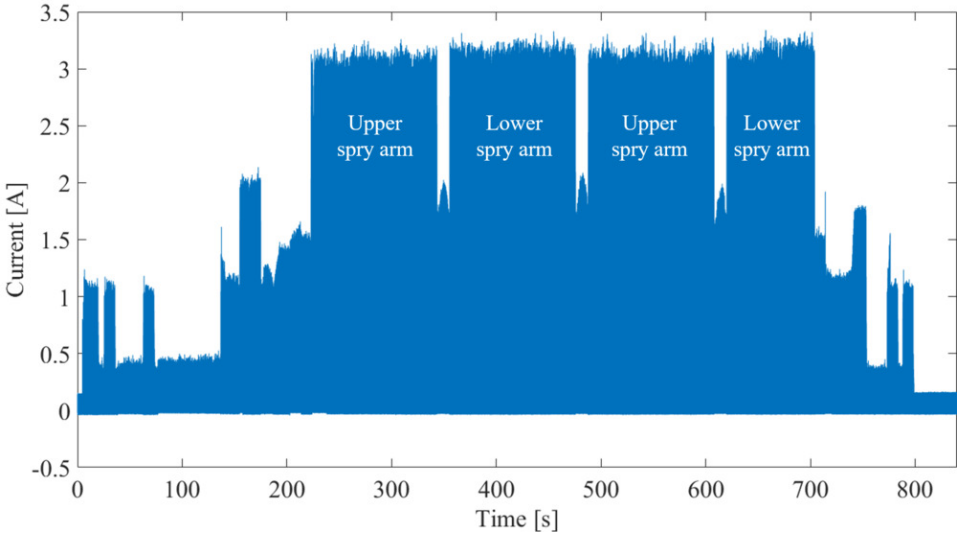
- polynomial with degree between 2 and 5.

Some of the tests on kernels are reported in table 6.2 for model 2 gearmotors. Best results, for model 1 gearmotors, are obtained with linear kernel and three features (mean value, minimum, and crest factor), and they are shown in table 6.3. To make a comparison, this table also shows results obtained through acceleration measurements with features selected in a previous work on similar gearmotors. Table 6.4 shows results for model 2 gearmotors for acceleration measurements, current measurements with three features, and current measurements with ten features. It is possible to notice that, if acceleration measurements are considered, SVMs average scores provide a good metric in order to evaluate noise level for both gearmotor models. The same stands also for model 1 current measurements but, on the contrary, model 2 current measurements allow one only to recognize particularly noisy gearmotors.
Table 6.5 shows accuracy for model 2 gearmotors with different expected classifications with a comparison between results obtained through current and acceleration measurements. Results obtained through SVMs with acceleration measurements are usually better, but anyway comparable, with the ones found with current measurements. In both cases accuracy is not particularly high probably also because there are not significant differences in acoustic noise among motors from different classes.
The other learning algorithm considered is NNs. All ten features in time domain are used, and different NNs are tested on model 1 motors but overfitting occurs also for small NNs. It has been decided to use a NN with one hidden layer with six nodes having ReLU activation function. The output node activation function is sigmoid, and the cost function is binary cross entropy.
Different stopping criteria have been tested: initially, both validation and test sets were present, and the first one was used to compute the accuracy. If both training and validation accuracy were over a predetermined threshold, then the training stopped. Problems encountered are that, depending on dataset partitioning, sometimes the stopping criterion is never reached,

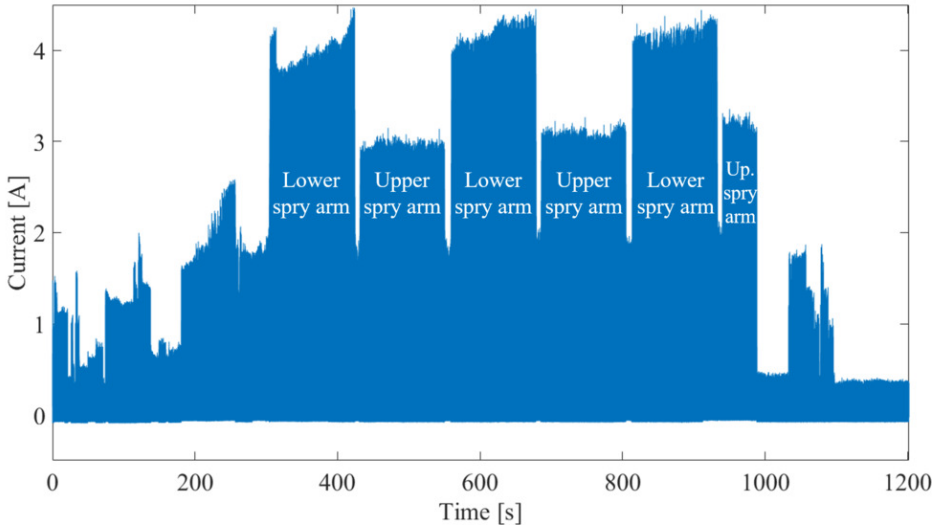
while other times it is reached almost immediately but test accuracy is very low or zero. Therefore, it has been decided to define only training and validation sets, in order to have larger datasets, and stopping criteria are the maximum number of epochs, set to 100, and training accuracy higher than 0.99 (figure 4.9).
NNs results shown in tables 6.6 and 6.7 are obtained in the same way of SVMs ones, by defining different expected classifications and moving the threshold that divides noisy motors from non-noisy ones. These results are slightly better than the ones obtained through SVMs. Instead, table 6.8 shows results for larger training sets in which 9 of the 18 model 2 motors belong to the training set. In this case, enlarging the training set does not allow one to have better performance with respect to the case of a training set composed of 6 motors.
Finally, tests are conducted in order to solve the regression problem in which the expected output is the acoustic noise of each motor. In this case, there is not the problem of defining a threshold that divides motors into noisy and non-noisy ones. For these tests, a NN similar to the one adopted for the classification problem, has been used. The only differences are the activation function of the last node of the NN that is a linear function and the cost function which is the mean squared error. Figures 6.6 and 6.7 show results for the case in which half of the motors belongs to the training set. Results are shown for each epoch and the only stopping criterion is the maximum number of epochs, which is set to 150. A second stopping criterion has not been defined yet but it is possible to see that results deeply depends on the number of epochs.
In general, current measurements allow one to identify particularly noisy gearmotors both through SVMs and NNs, but they are not enough in order to identify small differences in acoustic noise level.

5.2 Identification of dishwashers phases

In order to identify dishwashers phases, current and voltage are acquired for cold washes lasting about 15 minutes, with sampling frequency of 5 kHz. Figure 5.1 shows that phases identification can be a simple or a complex



(a) Brand 1 dishwasher



(b) Brand 2 dishwasher

Figure 5.1: Dishwashers current measurements

task depending on dishwashers model.

Current measurements are split and labelled depending on the phase. RMS current is computed with moving window of 0.02 seconds in order to compute all presented time domain features. Also delay between current and voltage has been tested as feature with no results (figure 3.6). Voltage-current trajectories are plot in order to evaluate if features can be extracted from them,

but figure 3.7 shows that they do not bring useful information. Therefore, only time domain features are used, which are computed on sequences of 5120 samples (about 1 second) of the RMS current. Afterwards dataset is repetitively split through leave-one-out cross-validation into training and validation sets and, each time, features are standardized considering mean and standard deviation of the training set. In this way, each time, about 35 million samples or raw data are used for training and 5 million for validation.

Then, 3-classes SVM is computed through ECOC with *one vs one* coding design matrix. Like for gearmotors, different kernels and features choices have been tested, but best results are obtained, for both dishwashers, using only the mean value as feature. This leads to an almost perfect classification for brand 2 dishwasher (table 4.3), but does not allow one to distinguish between the two phases with active spry arms for brand 1 dishwasher (table 4.4). Therefore, online monitoring of the brand 2 dishwasher, is easily achieved, while this does not stand for brand 1 dishwasher.

Chapter 6

Conclusions

The objective of this thesis was to provide a procedure to define, through machine learning techniques, an algorithm that outputs the status of a device, given current measurements and, in case, voltage ones. The targeted devices were DC permanent-magnets gearmotors and dishwashers. In the first case, the algorithm was able to detect particularly noisy gearmotors, while performance decreases if motors with an intermediate level of noise were considered. All motors were nominally the same, therefore, faults were probably due to dispersion in production process. This type of analysis was not present in the state of the art, where specific faults were considered. Instead, for dishwashers, results depend on the considered device, and there have been found no similar studies in literature that allow one to make a comparison.

Some aspects of this thesis can be further investigated. It is possible to better analyse gearmotors regression problem. In this case, it would be particularly convenient to have a larger sample of motors whose noise measurements are well distributed in a high range.

On the other hand, dishwashers analysis opens different directions for further developments. It is possible to repeat the analysis for washing cycles with hot water in order to understand if phases, which can be easily identified for brand 2 dishwasher, are still recognizable when the heating resistance is active. Another possibility is to repeat tests on cold washing cycles in order to identify a fault. This one can be a blocked spry arm or the water inlet

valve completely or partially closed.
Moreover, it is possible to make a different type of analysis, a non-intrusive load monitoring, in which the two dishwashers work simultaneously. Then it would be interesting to identify the contributions of each dishwashers and possibly its washing phase from the aggregate current and voltage measurements.

Appendix - additional figures and tables pertaining to the gearmotor data

Motor	Noise (dB)	First set of features		Second set of features	
		Class 1 (%)	Class 2 (%)	Class 1 (%)	Class 2 (%)
1	52.5	100	0	100	0
2	52.5	98	2	96	4
3	55.5	0	100	0	100
4	55.5	79	21	34	66
5	59	0	100	0	100
6	60.5	0	100	0	100
7	61	0	100	0	100
8	67	0	100	0	100

(a) Model 1 gearmotors

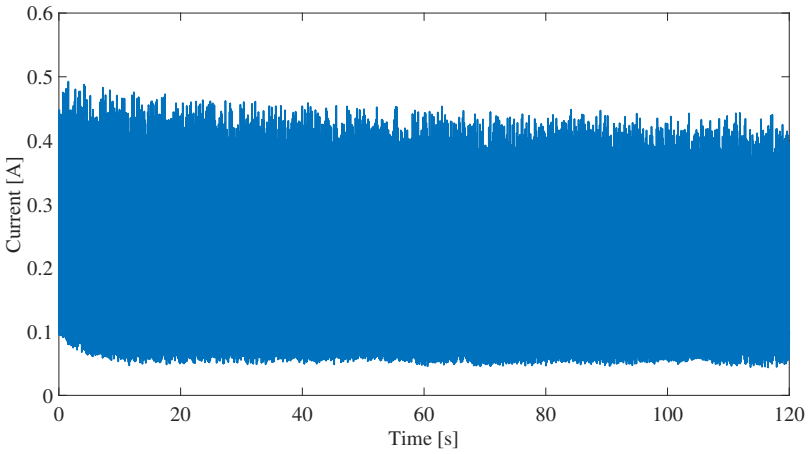
Motor	Noise (dB)	First set		Second set		Third set	
		Class 1	Class 2	Class 1	Class 2	Class 1	Class 2
1	56.5	0	100	0	100	0	100
2	59	0	100	0	100	0	100
3	60	0	100	0	100	0	100
4	60.2	0	100	0	100	0	100
5	60.5	1	99	1	99	2	98
6	60.5	0	100	0	100	0	100
7	61	0	100	0	100	0	100
8	61.3	0	100	0	100	0	100
9	61.5	0	100	0	100	0	100
10	62.4	0	100	0	100	98	2
11	63	0	100	0	100	0	100
12	63.2	0	100	0	100	50	50
13	63.5	0	100	0	100	0	100
14	63.5	0	100	0	100	0	100
15	64.5	0	100	0	100	0	100
16	76.3	100	0	100	0	100	0
17	77	100	0	100	0	100	0
18	78.2	100	0	100	0	100	0

(b) Model 2 gearmotors

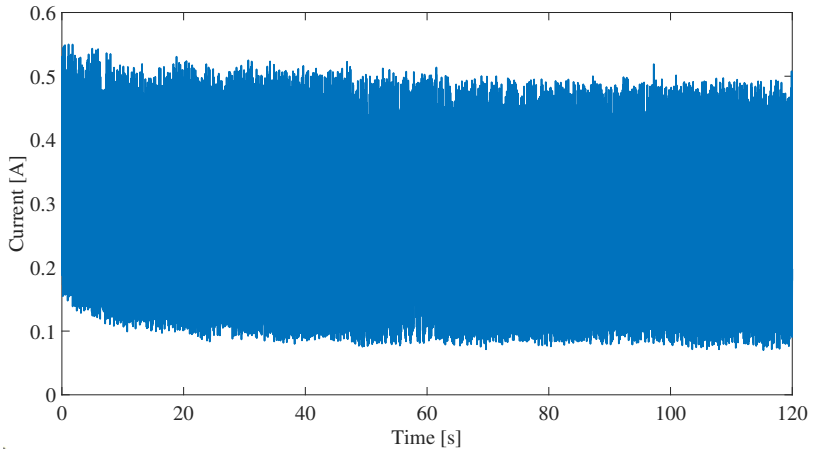
Motor	Noise (dB)	First set		Third set	
		Class 1	Class 2	Class 1	Class 2
1	56.5	100	0	100	0
2	59	13	87	100	0
3	60	100	0	100	0
4	60.2	83	17	100	0
5	60.5	32	68	99	1
6	60.5	100	0	100	0
7	61	99	1	100	0
8	61.3	13	87	99	1
9	61.5	100	0	100	0
10	62.4	21	79	0	100
11	63	94	6	100	0
12	63.2	100	0	11	90
13	63.5	68	32	100	0
14	63.5	0	100	99	1
15	64.5	0	100	100	0

(c) Model 2 gearmotors excluding the three most noisy motors

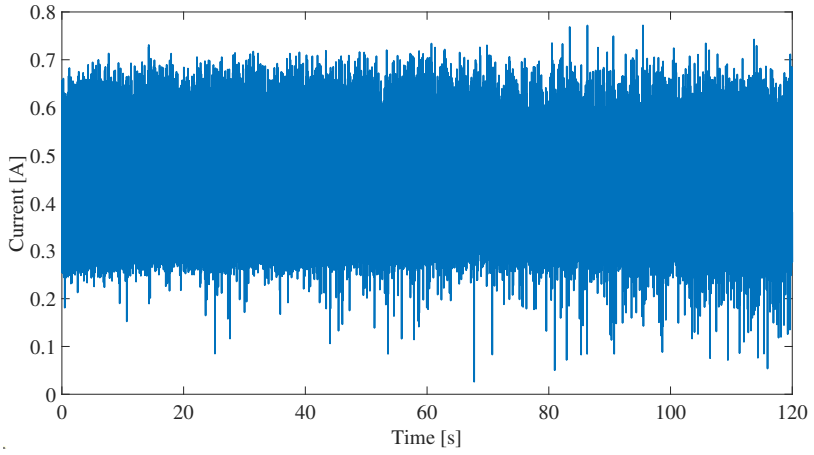
Table 6.1: K-means clustering with two classes for all gearmotors current measurements. Features are computed in time domain. First set of features: mean value, RMS, standard deviation, skewness, kurtosis, minimum, maximum, range, crest factor, and interquartile range. Second set of features: mean value, minimum, crest factor. A third set of features with the ones in the first set but not in the second one. In table (a) the test with the third set of features is not reported, since results are completely different for each iteration of k-means: they strongly depend on the initial position of centroids, which is random. The same happens with the second set of features in table (c).



(a) Motor 1 (noise: 56.5 dB)

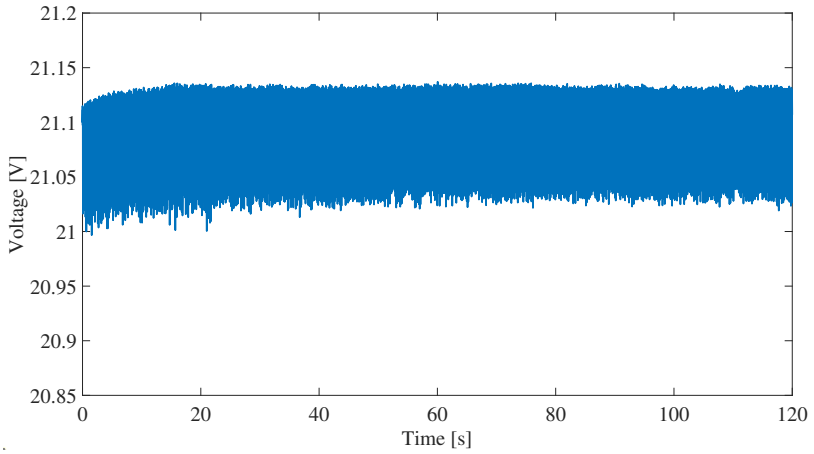


(b) Motor 15 (noise: 64.5 dB)

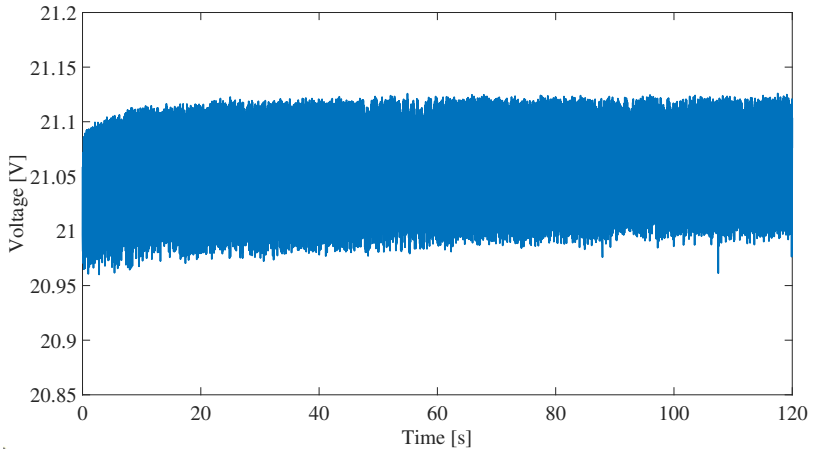


(c) Motor 18 (noise: 78.2 dB)

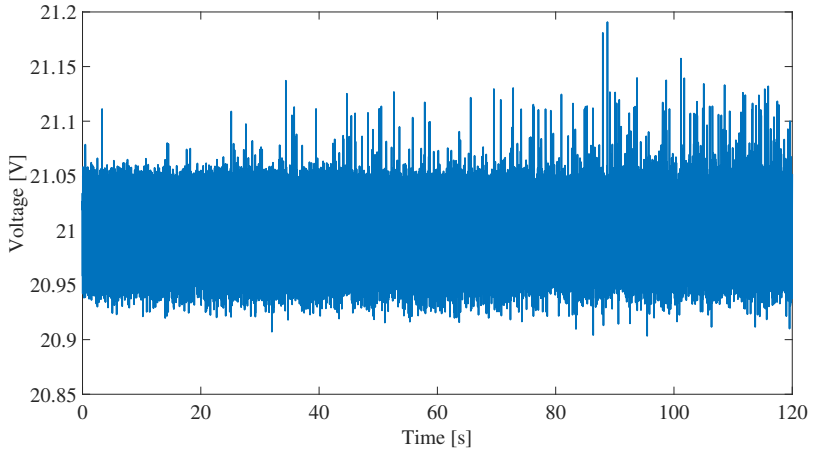
Figure 6.1: Model 2 gearmotor current measurements



(a) Motor 1 (noise: 56.5 dB)

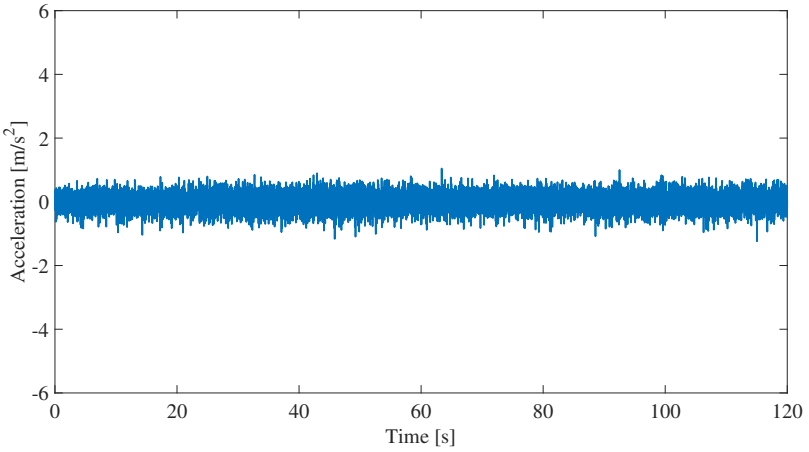


(b) Motor 15 (noise: 64.5 dB)

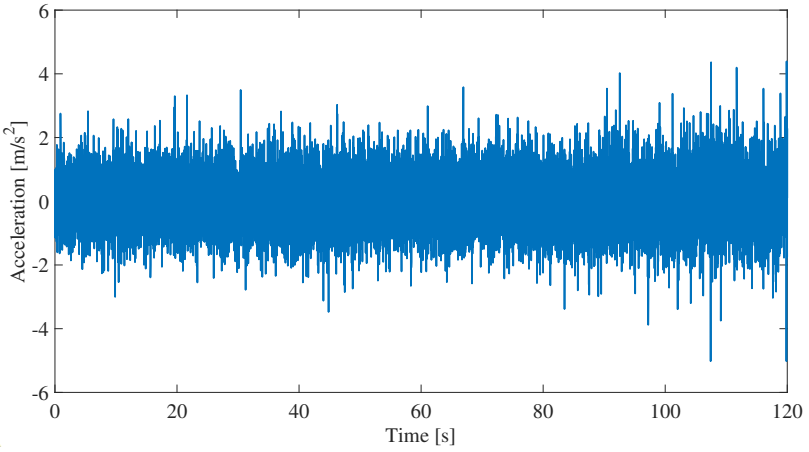


(c) Motor 18 (noise: 78.2 dB)

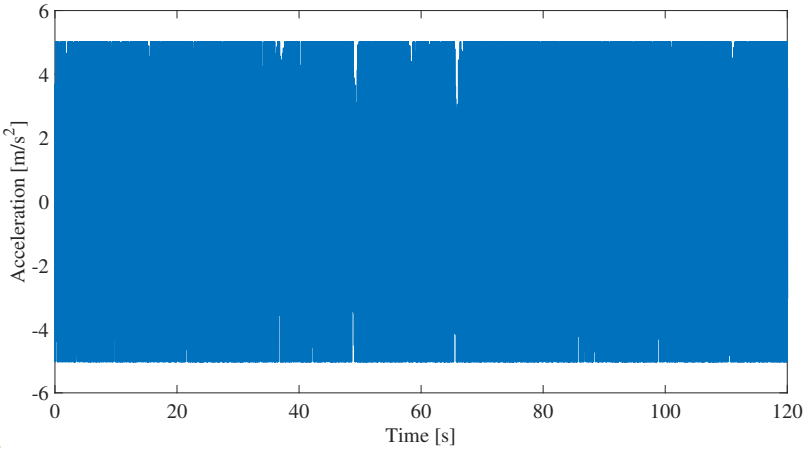
Figure 6.2: Model 2 gearmotor voltage measurements



(a) Motor 1 (noise: 56.5 dB)

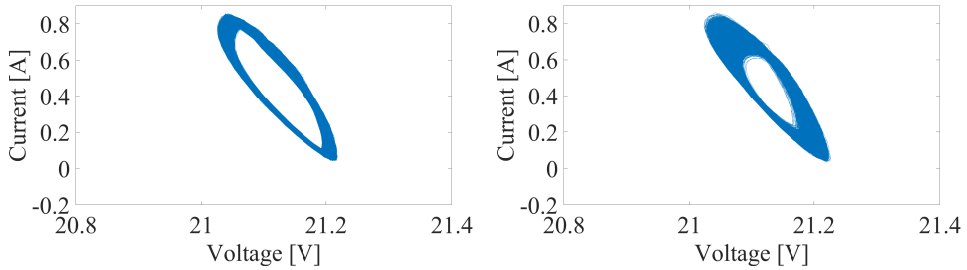


(b) Motor 15 (noise: 64.5 dB)



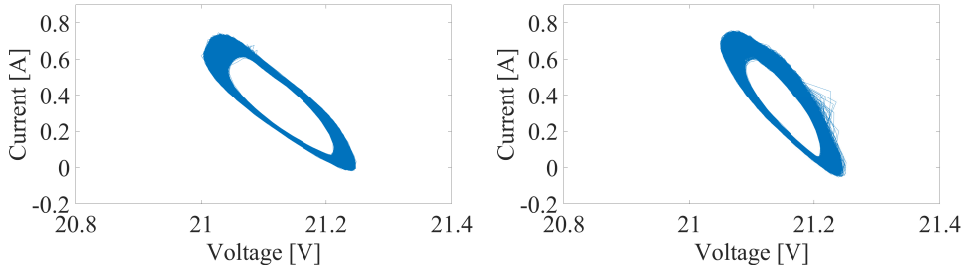
(c) Motor 18 (noise: 78.2 dB). Acceleration exceeds measurement range, which is $\pm 5\ m/s^2$

Figure 6.3: Model 2 gearmotor acceleration measurements



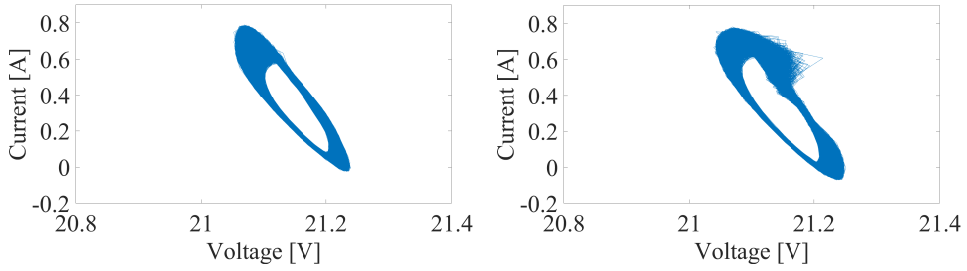
(a) Motor 1 (noise: 52.5 dB)

(b) Motor 2 (noise: 52.5 dB)



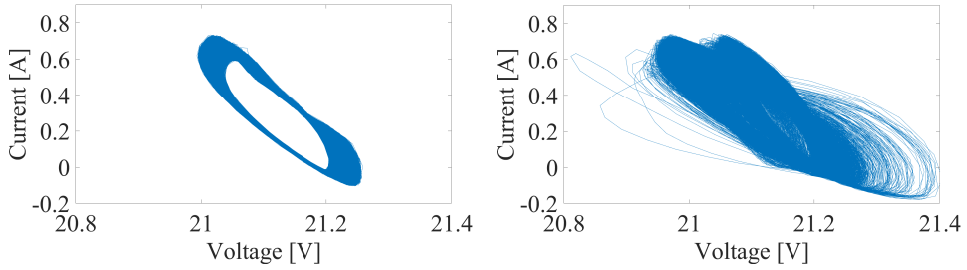
(c) Motor 3 (noise: 55.5 dB)

(d) Motor 4 (noise: 55.5 dB)



(e) Motor 5 (noise: 59.0 dB)

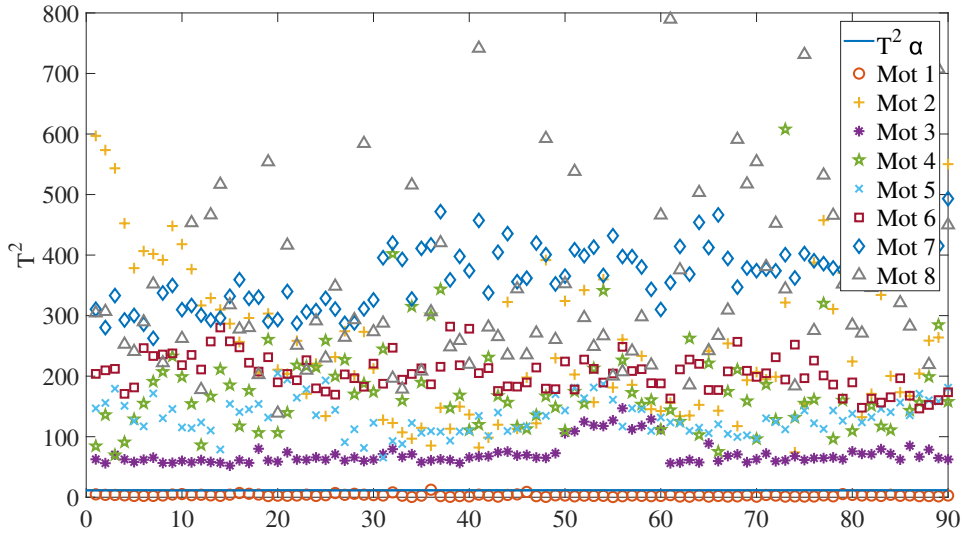
(f) Motor 6 (noise: 60.5 dB)



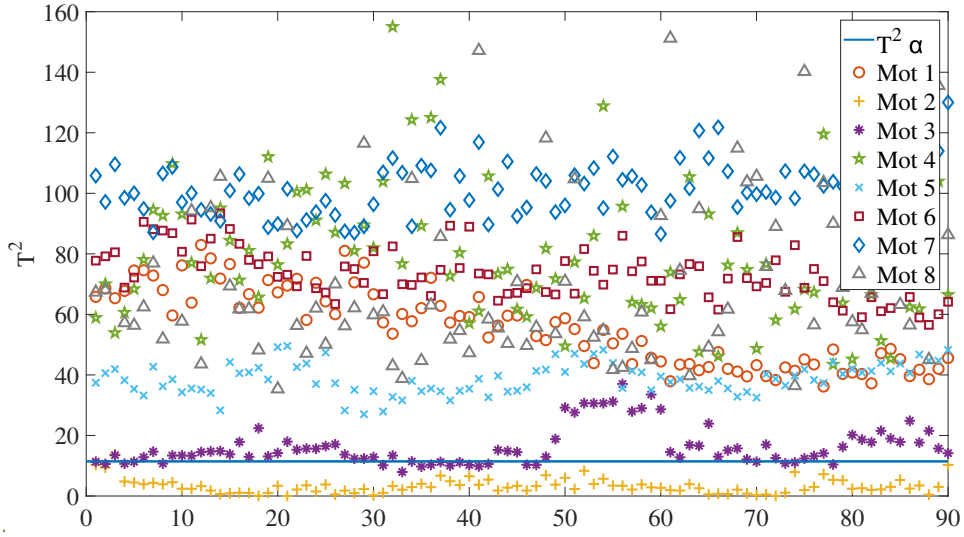
(g) Motor 7 (noise: 61.0 dB)

(h) Motor 8 (noise: 67.0 dB)

Figure 6.4: Voltage-current trajectories of model 1 gearmotors, first measurement, second minute



(a) Training set: motor 1



(b) Training set: motor 2

Figure 6.5: Results of Hotelling's T^2 distance for all current measurements of model 1 gearmotors with significance level $\alpha = 0.01$. Time domain features: mean value, minimum, crest factor. Only one value of T^2 every 10 is plotted to make the figure more readable.

Motor	Noise (dB)	Expected class	Test 1		Test 2		Test 3	
			Pred class	Avg score	Pred class	Avg score	Pred class	Avg score
1	56.5	1	1	1.9	1	1.3	1	1.8
2	59	1	1	3.2	-1	-0.5	1	7.3
3	60	1	1	3.2	-1	-0.3	1	3.8
4	60.2	1	1	5.9	-1	-0.4	1	12.9
5	60.5	1	-1	-0.5	-1	-0.2	1	0.2
6	60.5	1	1	8.0	-1	-0.5	1	31.8
7	61	1	1	2.4	1	1.3	1	2.0
8	61.3	-1	-1	-2.7	-1	-1.1	-1	-4.9
9	61.5	-1	1	1.4	1	0.8	1	1.5
10	62.4	-1	-1	-5.6	-1	-0.5	-1	-45.2
11	63	-1	1	2.5	1	0.0	1	2.9
12	63.2	-1	1	1.0	-1	-0.2	1	3.9
13	63.5	-1	-1	-0.4	-1	-0.4	1	0.4
14	63.5	-1	-1	-0.6	-1	-0.4	1	0.1
15	64.5	-1	-1	-3.0	-1	-1.1	-1	-2.6
16	76.3	-1	-1	-17.3	-1	-0.5	-1	-224.3
17	77	-1	-1	-13.7	-1	-0.5	-1	-64.3
18	78.2	-1	-1	-15.9	-1	-0.5	-1	-132.7

Table 6.2: SVMs with different kernels for all model 2 gearmotors current measurements. Features, computed in time domain, are: mean value, RMS, standard deviation, skewness, kurtosis, minimum, maximum, range, crest factor, and interquartile range. Training set: motors 1, 7 (non-noisy) and 8, 15 (noisy). Test 1: linear kernel; test 2: radial basis function kernel; test 3: cubic function kernel.

Motor	Noise (dB)	Acceleration		Current	
		Pred class	Avg score	Pred class	Avg score
1	52.5	1	1.3	1	1.4
2	52.5	1	1.3	1	1.2
3	55.5	1	1.0	-1	-0.3
4	55.5	1	0.9	1	0.2
5	59	1	0.9	-1	-0.5
6	60.5	1	1.1	-1	-0.5
7	61	1	0.9	-1	-0.7
8	67	-1	-1.4	-1	-1.4

Table 6.3: SVMs for all model 1 gearmotors with linear kernel. Acceleration vs current measurements. Features, computed in time domain, are: standard deviation, skewness, and kurtosis for acceleration, and mean value, minimum, and crest factor for current. Training set: motors 1 (non-noisy) and 8 (noisy). Therefore, no expected classification is defined.

Motor	Noise (dB)	Expected class	Acceleration		Current (test1)		Current (test2)	
			Pred class	Avg score	Pred class	Avg score	Pred class	Avg score
1	56.5	1	1	4.4	1	2.6	1	1.9
2	59	1	1	3.0	1	10.1	1	3.2
3	60	1	1	2.8	1	12.2	1	3.2
4	60.2	1	1	3.5	1	11.6	1	5.9
5	60.5	1	-1	-3.9	-1	-4.0	-1	-0.5
6	60.5	1	1	2.1	1	23.4	1	8.0
7	61	1	1	1.9	1	6.5	1	2.4
8	61.3	-1	-1	-2.0	-1	-3.8	-1	-2.7
9	61.5	-1	-1	-0.9	1	4.8	1	1.4
10	62.4	-1	-1	-2.0	-1	-6.6	-1	-5.6
11	63	-1	-1	-0.4	1	5.8	1	2.5
12	63.2	-1	1	1.2	1	2.4	1	1.0
13	63.5	-1	-1	-3.9	1	3.2	-1	-0.4
14	63.5	-1	-1	-6.3	-1	-0.7	-1	-0.6
15	64.5	-1	-1	-4.6	-1	-4.8	-1	-3.0
16	76.3	-1	-1	-24.8	-1	-37.7	-1	-17.3
17	77	-1	-1	-37.5	-1	-27.9	-1	-13.7
18	78.2	-1	-1	-49.7	-1	-9.2	-1	-15.9
Training accuracy			1		1		1	
Validation accuracy			0.86		0.64		0.71	

Table 6.4: SVMs for all model 2 gearmotors with linear kernel. Acceleration vs current measurements. Features, computed in time domain, are: standard deviation, skewness, and kurtosis for acceleration; mean value, minimum, and crest factor for current (test 1); mean value, RMS, standard deviation, skewness, kurtosis, minimum, maximum, range, crest factor, and interquartile range for current (test 2). Training set: motors 1, 7 (non-noisy) and 8, 15 (noisy).

Number of expected non-noisy motors	Acceleration		Current (test 1)		Current (test 2)	
	Tr	Val	Tr	Val	Tr	Val
6	1	0.92	0.83	0.42	1	0.42
7	1	0.83	1	0.58	1	0.67
8	1	0.83	1	0.67	1	0.58
9	0.83	0.75	1	0.5	1	0.5
10	1	0.67	1	0.5	1	0.25
11	1	0.5	1	0.83	1	0.83
12	1	0.58	1	0.83	1	0.67
Average	0.98	0.73	0.98	0.62	1	0.56

Table 6.5: SVMs training and validation accuracy for model 2 gearmotors and different expected classifications. Features, computed in time domain, are: standard deviation, skewness, and kurtosis for acceleration; mean value, minimum, and crest factor for current (test 1); mean value, RMS, standard deviation, skewness, kurtosis, minimum, maximum, range, crest factor, and interquartile range for current (test 2). Three noisy motors and three non-noisy ones are used for training set: for each class, the most noisy motor, the less noisy, and one in the middle are chosen.

Number of expected non-noisy motors	Training accuracy	Validation accuracy
3	1	0.39
4	1	0.64
5	1	0.99

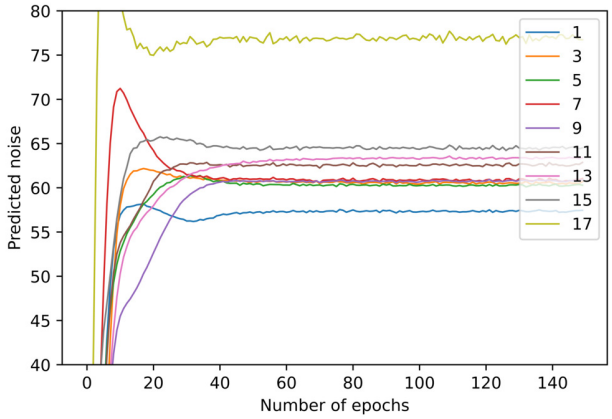
Table 6.6: NNs accuracy for model 1 gearmotors and different expected classifications. Features are all the presented ones computed in time domain. Two noisy motors and two non-noisy ones are used for training set: for each class, the most noisy motor and the less noisy one are chosen.

Number of expected non-noisy motors	Training accuracy	Validation accuracy
6	1	0.62
7	1	0.66
8	0.99	0.74
9	1	0.87
10	1	0.57
11	1	0.88
12	1	0.91
Average	1	0.75

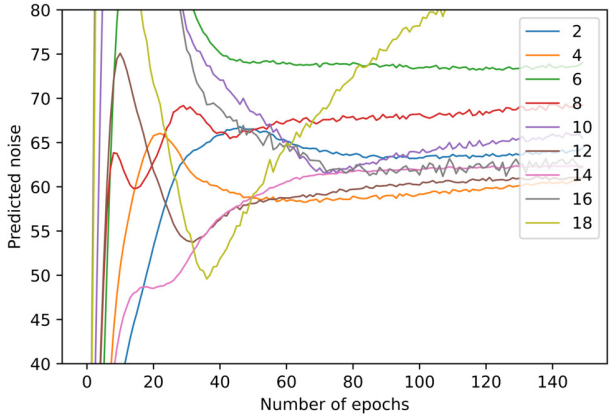
Table 6.7: NNs accuracy for model 2 gearmotors and different expected classifications. Features are all the presented ones computed in time domain. Three noisy motors and three non-noisy ones are used for training set: for each class, the most noisy motor, the less noisy, and one in the middle are chosen.

Motor	Noise (dB)	Expected class	Test 1 predicted class	Test 1 predicted class
1	56.5	1	1	1
2	59	1	-1	1
3	60	1	1	1
4	60.2	1	-1	1
5	60.5	1	1	-1
6	60.5	1	1	1
7	61	1	1	1
8	61.3	1	1	1
9	61.5	1	1	1
10	62.4	-1	-1	-1
11	63	-1	-1	1
12	63.2	-1	1	-1
13	63.5	-1	-1	-1
14	63.5	-1	-1	-1
15	64.5	-1	-1	-1
16	76.3	-1	-1	-1
17	77	-1	-1	-1
18	78.2	-1	-1	-1
Training accuracy			1	1
Validation accuracy			0.68	0.75

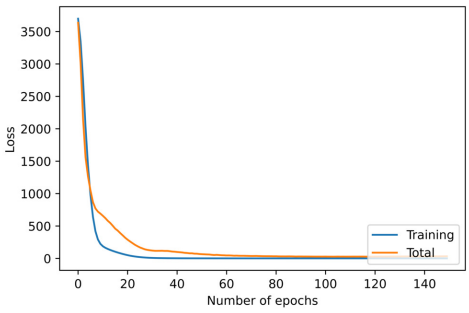
Table 6.8: NNs results for model 2 gearmotors classification with 9 motors belonging to the training set and the other 9 to the validation set. test 1: odd motors belong to the training set; test 2: even motors belong to the training set. Features are all the presented ones computed in time domain. Training and validation accuracy are computed considering all samples predictions.



(a) Training set (one curve for each motor)

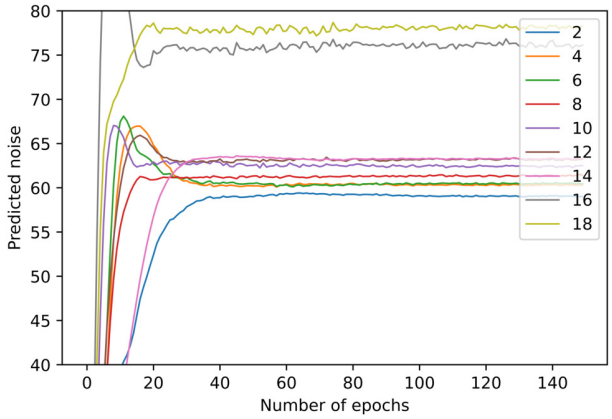


(b) Validation set (one curve for each motor)

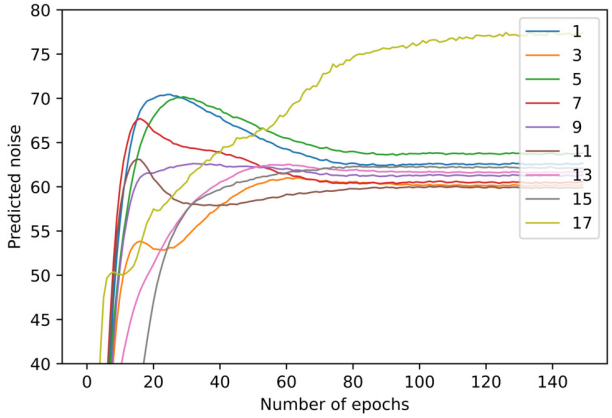


(c) Mean of the cost function of the training set and of the entire dataset

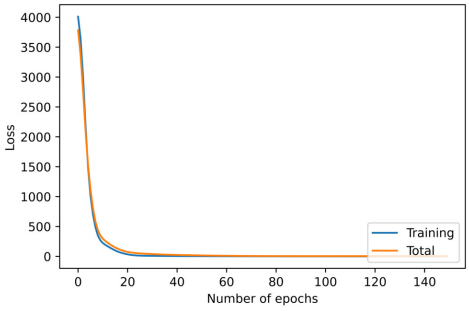
Figure 6.6: NNs results for model 2 gearmotors regression problem with 9 motors (odd motors) belonging to the training set and the other 9 (even motors) to the validation set. Features are all the presented ones computed in time domain. Results are evaluated at the end of each epoch: the features matrix corresponding to one motor is passed to the NN and the mean of its outputs is the predicted noise for that motor.



(a) Training set (one curve for each motor)



(b) Validation set (one curve for each motor)



(c) Mean of the cost function of the training set and of the entire dataset

Figure 6.7: NNs results for model 2 gearmotors regression problem with 9 motors (even motors) belonging to the training set and the other 9 (odd motors) to the validation set. Features are all the presented ones computed in time domain. Results are evaluated at the end of each epoch: the features matrix corresponding to one motor is passed to the NN and the mean of its outputs is the predicted noise for that motor.

Bibliography

[1] Rasool Mohsenzadeh, Hojjat Majidi, Mohsen Soltanzadeh, and Karim Shelesh-Nezhad. Wear and failure of polyoxymethylene/calcium carbonate nanocomposite gears. *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, 234(6):811–820, 2020.

[2] Amit Aherwar. An investigation on gearbox fault detection using vibration analysis techniques: A review. *Australian Journal of Mechanical Engineering*, 10(2):169–183, 2012.

[3] Y Gritli, A Bellini, C Rossi, D Casadei, F Filippetti, and GA Capolino. Condition monitoring of mechanical faults in induction machines from electrical signatures: Review of different techniques. In *2017 IEEE 11th international symposium on diagnostics for electrical machines, power electronics and drives (SDEMPED)*, pages 77–84. IEEE, 2017.

[4] François Combet. Gear fault diagnosis by motor current analysis–application to industrial cases. In *International Conference Surveillance*, volume 7, pages 1–7, 2015.

[5] Xiang-Qun Liu, Hong-Yue Zhang, Jun Liu, and Jing Yang. Fault detection and diagnosis of permanent-magnet dc motor based on parameter estimation and neural network. *IEEE transactions on industrial electronics*, 47(5):1021–1030, 2000.

[6] Vikas Sharma and Anand Parey. A review of gear fault diagnosis using various condition indicators. *Procedia Engineering*, 144:253–263, 2016.

[7] Hossein Davari Ardakani, Zongchang Liu, Jay Lee, Inaki Bravo-Imaz, and Aitor Arnaiz. Motor current signature analysis for gearbox fault diagnosis in transient speed regimes. In *2015 IEEE Conference on Prognostics and Health Management (PHM)*, pages 1–6. IEEE, 2015.

[8] Shuanfeng Zhao, Pengfei Wang, and Shijun Li. Study on the fault diagnosis method of scraper conveyor gear under time-varying load condition. *Applied Sciences*, 10(15):5053, 2020.

[9] Chinmaya Kar and AR Mohanty. Vibration and current transient monitoring for gearbox fault detection using multiresolution fourier transform. *Journal of Sound and Vibration*, 311(1-2):109–132, 2008.

[10] Syed Sajjad H Zaidi, Selin Aviyente, Mutasim Salman, Kwang-Kuen Shin, and Elias G Strangas. Prognosis of gear failures in dc starter motors using hidden markov models. *IEEE Transactions on industrial Electronics*, 58(5):1695–1706, 2010.

[11] Subhasis Nandi, Hamid A Toliyat, and Xiaodong Li. Condition monitoring and fault diagnosis of electrical motors—a review. *IEEE transactions on energy conversion*, 20(4):719–729, 2005.

[12] Satish Rajagopalan, Jos M Aller, Jos A Restrepo, Thomas G Habetler, and Ronald G Harley. Detection of rotor faults in brushless dc motors operating under nonstationary conditions. *IEEE Transactions on Industry Applications*, 42(6):1464–1477, 2006.

[13] Yong Chen, Siyuan Liang, Wanfu Li, Hong Liang, and Chengdong Wang. Faults and diagnosis methods of permanent magnet synchronous motors: A review. *Applied Sciences*, 9(10):2116, 2019.

[14] Ruiming Fang and Hongzhong Ma. Application of mcsa and svm to induction machine rotor fault diagnosis. In *2006 6th World Congress on Intelligent Control and Automation*, volume 2, pages 5543–5547. IEEE, 2006.

[15] Anjali Jawadekar, Sudhir Paraskar, Saurabh Jadhav, and Gajanan Dhole. Artificial neural network-based induction motor fault classi-

fier using continuous wavelet transform. *Systems Science & Control Engineering: An Open Access Journal*, 2(1):684–690, 2014.

[16] Masoud Hajiaghajani, Hamid A Toliyat, and Issa MS Panahi. Advanced fault diagnosis of a dc motor. *IEEE Transactions on energy conversion*, 19(1):60–65, 2004.

[17] Sai Munikoti, Laya Das, Balasubramaniam Natarajan, and Babji Srinivasan. Data-driven approaches for diagnosis of incipient faults in dc motors. *IEEE Transactions on Industrial Informatics*, 15(9):5299–5308, 2019.

[18] MA Awadallah and MM Morcos. Anfis-based diagnosis and location of stator interturn faults in pm brushless dc motors. *IEEE Transactions on energy conversion*, 19(4):795–796, 2004.

[19] A Glowacz. Diagnostics of dc and induction motors based on the analysis of acoustic signals. *Measurement Science Review*, 14(5):257–262, 2014.

[20] Abbas A Wahab, NF Abdullah, and MAH Rasid. Commutator fault detection of brushed dc motor using thermal assessment. In *IOP Conference Series: Materials Science and Engineering*, volume 469, page 012057. IOP Publishing, 2019.

[21] Vineetha P Raj, K Natarajan, and Sri TG Girikumar. Induction motor fault detection and diagnosis by vibration analysis using mems accelerometer. In *2013 International Conference on Emerging Trends in Communication, Control, Signal Processing and Computing Applications (C2SPCA)*, pages 1–6. IEEE, 2013.

[22] Kawthar Alameh, Ghaleb Hoblos, and Georges Barakat. Statistical vibration-based fault diagnosis approach applied to brushless dc motors. *IFAC-PapersOnLine*, 51(24):338–345, 2018.

[23] Parasuram P Harihara, Kyusung Kim, and Alexander G Parlos. Signal-based versus model-based fault diagnosis-a trade-off in complexity and performance. In *4th IEEE International Symposium on Diagnostics for*

Electric Machines, Power Electronics and Drives, 2003. SDEMPED 2003., pages 277–282. IEEE, 2003.

[24] Li-Ming Wang and Yi-Min Shao. Crack fault classification for planetary gearbox based on feature selection technique and k-means clustering method. *Chinese Journal of Mechanical Engineering*, 31(1):1–11, 2018.

[25] Ziyuan Jiang, Qinkai Han, and Xueping Xu. Fault diagnosis of planetary gearbox based on motor current signal analysis. *Shock and Vibration*, 2020, 2020.

[26] Salman Hajiaghasi, Zahra Rafiee, Ahmad Salemnia, Mohammadreza Aghamohammadi, and Tohid Soleymaniaghdam. A new strategy for induction motor fault detection based on wavelet transform and probabilistic neural network. In *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, pages 118–123. IEEE, 2019.

[27] Issah Ibrahim, Rodrigo Silva, MH Mohammadi, Vahid Ghorbanian, and David A Lowther. Surrogate-based acoustic noise prediction of electric motors. *IEEE Transactions on Magnetics*, 56(2):1–4, 2020.

[28] Sunghyuk Park, Wonho Kim, and Sung-Il Kim. A numerical prediction model for vibration and noise of axial flux motors. *IEEE Transactions on Industrial Electronics*, 61(10):5757–5762, 2014.

[29] Dimitri Torregrossa, Amir Khoobroo, and Babak Fahimi. Prediction of acoustic noise and torque pulsation in pm synchronous machines with static eccentricity and partial demagnetization using field reconstruction method. *IEEE Transactions on Industrial Electronics*, 59(2):934–944, 2011.

[30] Chong Wang, Joseph CS Lai, and Duco WJ Pulle. Prediction of acoustic noise from variable-speed induction motors: deterministic versus statistical approaches. *IEEE Transactions on Industry Applications*, 38(4):1037–1044, 2002.

[31] Andreas Hauer and Fabian Fischer. Open adsorption system for an energy efficient dishwasher. *Chemie Ingenieur Technik*, 83(1-2):61–66, 2011.

[32] G Santori, A Frazzica, A Freni, M Galieni, L Bonaccorsi, F Polonara, and G Restuccia. Optimization and testing on an adsorption dishwasher. *Energy*, 50:170–176, 2013.

[33] Isiyaku Abubakar, SN Khalid, MW Mustafa, Hussain Shareef, and Mamunu Mustapha. An overview of non-intrusive load monitoring methodologies. In *2015 IEEE Conference on Energy Conversion (CENCON)*, pages 54–59. IEEE, 2015.

[34] Ahmed Zoha, Alexander Gluhak, Muhammad Ali Imran, and Sutharshan Rajasegarar. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 12(12):16838–16866, 2012.

[35] D Srinivasan, WS Ng, and AC Liew. Neural-network-based signature recognition for harmonic source identification. *IEEE Transactions on Power Delivery*, 21(1):398–405, 2005.

[36] Lei Jiang, Jiaming Li, Suhuai Luo, Sam West, and Glenn Platt. Power load event detection and classification based on edge symbol analysis and support vector machine. *Applied Computational Intelligence and Soft Computing*, 2012, 2012.

[37] Marisa B Figueiredo, Ana De Almeida, and Bernardete Ribeiro. An experimental study on electrical signature identification of non-intrusive load monitoring (nilm) systems. In *International Conference on Adaptive and Natural Computing Algorithms*, pages 31–40. Springer, 2011.

[38] Kunjin Chen, Yu Zhang, Qin Wang, Jun Hu, Hang Fan, and Jinliang He. Scale-and context-aware convolutional non-intrusive load monitoring. *IEEE Transactions on Power Systems*, 35(3):2362–2373, 2019.

[39] Herman Aguinis, Ryan K Gottfredson, and Harry Joo. Best-practice recommendations for defining, identifying, and handling outliers. *Organizational Research Methods*, 16(2):270–301, 2013.

[40] Yayu Peng, Wei Qiao, Liyan Qu, and Jun Wang. Gearbox fault diagnosis using vibration and current information fusion. In *2016 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 1–6. IEEE, 2016.

[41] Khalaf Salloum Gaeid and Hew Wooi Ping. Wavelet fault diagnosis of induction motor. In *MATLAB for Engineers–Applications in Control, Electrical Engineering, IT and Robotics*. University of Malaya Malaysia, 2011.

[42] Charles Knapp and Glifford Carter. The generalized correlation method for estimation of time delay. *IEEE transactions on acoustics, speech, and signal processing*, 24(4):320–327, 1976.

[43] Hong Yin Lam, GSK Fung, and WK Lee. A novel method to construct taxonomy electrical appliances based on load signaturesof. *IEEE Transactions on Consumer Electronics*, 53(2):653–660, 2007.

[44] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[45] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

[46] G Willems, Greet Pison, PJ Rousseeuw, and Stefan Van Aelst. A robust hotelling test. *Metrika*, 55(1-2):125–138, 2002.

[47] Asa Ben-Hur, David Horn, Hava T Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of machine learning research*, 2(Dec):125–137, 2001.

[48] Harris Drucker, Chris JC Burges, Linda Kaufman, Alex Smola, Vladimir Vapnik, et al. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161, 1997.

[49] Tsung-Chin Wu, Zhirou Zhou, Hongyue Wang, Bokai Wang, Tuo Lin, Changyong Feng, and Xin M Tu. Advanced machine learning methods in psychiatry: an introduction. *General psychiatry*, 33(2), 2020.

[50] Lipo Wang. *Support vector machines: theory and applications*, volume 177. Springer Science & Business Media, 2005.

[51] Marouane Hachimi, Georges Kaddoum, Ghyslain Gagnon, and Poulmanogo Illy. Multi-stage jamming attacks detection using deep learning combined with kernelized support vector machine in 5g cloud radio access networks. In *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–5. IEEE, 2020.

[52] Hsuan-Tien Lin and Chih-Jen Lin. A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods. *submitted to Neural Computation*, 3(1-32):16, 2003.

[53] John C Platt, Nello Cristianini, John Shawe-Taylor, et al. Large margin dags for multiclass classification. In *nips*, volume 12, pages 547–553, 1999.

[54] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001.

[55] Huan Wu, Liang Wang, Zhiyong Zhao, Chester Shu, and Chao Lu. Support vector machine based differential pulse-width pair brillouin optical time domain analyzer. *IEEE Photonics Journal*, 10(4):1–11, 2018.

[56] Björn RH Blomqvist, Richard P Mann, and David JT Sumpter. Using bayesian dynamical systems, model averaging and neural networks to determine interactions between socio-economic indicators. *Plos one*, 13(5):e0196355, 2018.

[57] Sagar Sharma. Activation functions in neural networks. *towards data science*, 6, 2017.

[58] Jin-Cheng Li, Wing WY Ng, Daniel S Yeung, and Patrick PK Chan. Bi-firing deep neural networks. *International Journal of Machine Learning and Cybernetics*, 5(1):73–83, 2014.

[59] Vahid Nourani, Biswajeet Pradhan, Hamid Ghaffari, and Seyed Saber Sharifi. Landslide susceptibility mapping at zonouz plain, iran using genetic programming and comparison with frequency ratio, logistic regression, and artificial neural network models. *Natural hazards*, 71(1):523–547, 2014.

[60] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[61] Ekachai Phaisangittisagul. An analysis of the regularization between l2 and dropout in single hidden layer neural network. In *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, pages 174–179. IEEE, 2016.