ABSTRACT

HIDDEN MARKOV MODEL-SUPPORTED MACHINE LEARNING FOR CONDITION MONITORING OF DC-LINK CAPACITORS

by Viktoriia Sysoeva

Power electronics are critical components in society's modern infrastructure. In electrified vehicles and aircraft, losing power jeopardizes personal safety and incur financial penalties. Because of these concerns, many researchers explore condition monitoring (CM) methods that provide real-time information about a system's health. This thesis develops a CM method that determines the health of a DC-link capacitor in a three-phase inverter. The approach uses measurements from a current transducer in two Machine Learning (ML) algorithms, a Support Vector Machine (SVM), and an Artificial Neural Network (ANN), that classify the data into groups corresponding to the capacitor's health. This research evaluates six sets of data: time-domain, frequency-domain, and frequency-domain data subjected to four smoothing filters: the moving average with a rectangular window (MARF) and a Hanning window, the locally weighted linear regression, and the Savitzky–Golay filter. The results show that both ML algorithms estimate the DC-link capacitor health with the highest accuracy being 91.8% for the SVM and 90.7% for the ANN. The MARF-smoothed data is an optimal input data type for the ML classifiers due to its low computational cost and high accuracy. Additionally, a Hidden Markov Model increases the classification accuracy up to 98% when utilized with the ANN.

HIDDEN MARKOV MODEL-SUPPORTED MACHINE LEARNING FOR CONDITION MONITORING OF DC-LINK CAPACITORS

A Thesis

Submitted to the

Faculty of Miami University

in partial fulfillment of

the requirements for the degree of

Master of Science

by

Viktoriia Sysoeva

Miami University

Oxford, Ohio

2020

Advisor: Dr. Mark J. Scott

Co-Advisor: Dr. Chi-Hao Cheng

Reader: Dr. Peter Jamieson

This Thesis titled

HIDDEN MARKOV MODEL-SUPPORTED MACHINE LEARNING FOR CONDITION MONITORING OF DC-LINK CAPACITORS

by

Viktoriia Sysoeva

has been approved for publication by

The College of Engineering and Computing

and

Department of Electrical and Computer Engineering

____
Dr. Mark J. Scott

____
Dr. Chi-Hao Cheng

____
Dr. Peter Jamieson

Table of Contents

List of Tables

List of Figures

Glossary of Abbreviations and Acronyms

AE	Acoustic emissions
AI	Artificial Intelligence
ANFIS	Adaptive neuro-fuzzy inference system
ANN	Artificial Neural Network
ASA	Acoustic signature analysis
AWT	Adaptive Wavelet Transform
BPF	Band-pass filter
CM	Condition monitoring
CT	Current transducer
CWT	Continuous Wavelet Transform
DFT	Discrete Fourier Transform
DHWPT	Discrete Harmonic Wavelet Packet Transform
DSP	Digital signal processing
DWT	Discrete Wavelet Transform
EMI	Electromagnetic interference
ESA	Electrical Signature Analysis
ESR	Equivalent series resistance
FFT	Fast Fourier Transform
HHT	Hilbert-Huang Transform
HMM	Hidden Markov Model
IRT	Infrared thermography
LISN	Line impedance stabilization network
LMS	Least mean square
LRF	Locally weighted linear regression filter
MAHF	Moving average filter with a Hanning window
MARF	Moving average filter with a rectangular window
MEA	More electric aircraft
ML	Machine Learning
MLP	Multilayer perceptron
MOSFET	Metal oxide semiconductor field effect transistor
MPPF	Metalized polypropylene film capacitor
MRO	Maintenance, repair, and overhaul
MSF	Multiscale filtering
OVA	One-versus-all
OVO	One-versus-one
PHM	Prognostic and health management
PCB	Printed circuit board
PWM	Pulse width modulation
RBF	Radial basis function
RL	Reinforcement Learning
RMS	Root mean square
RUL	Remaining useful life
SGF	Savitzky–Golay filter
SL	Supervised Learning

SSL	Semi-supervised Learning
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
SVR	Support Vector Regression
TFR	Time-frequency representation
TSA	Thermal signature analysis
USSA	Ultrasonic signature analysis
UL	Unsupervised Learning
VSA	Vibration signature analysis
WT	Wavelet Transform

Chapter 1 Introduction

Power electronics fulfill mission-critical roles in renewable energy applications, power transmission and distribution, and electrified transportation. It is important to guarantee they operate reliably to protect the end-users and to minimize financial penalties that occur when they fail. This chapter introduces real-time reliability control, which is an important aspect of power electronics. It describes the applications of condition monitoring (CM) and addresses the concerns that motivate researchers to develop non-invasive prognostic and health management (PHM) systems, which is the focal point of this research work. Then it presents a new E-PHM DC-link capacitor CM approach, along with the contributions of this research. Finally, it provides the thesis objectives and structure.

1.1 Reliability and Maintenance Aspects of Power Electronic Applications

Safe operation and proper maintenance of equipment are crucial elements in all industries. For example, the airline industry spent about $76 billion on maintenance, repair, and overhaul (MRO) of its airline fleet globally in 2017 [1]. These expenses made up 11% of the total operating costs for a regular aircraft, which was a significant financial burden for the airline industry. Moreover, the MRO costs will increase by 4.6% annually, approaching a substantial $118 billion threshold by 2027 (see Fig. 1.1) [1]. In response to the rising costs, airline manufacturers and fleet operations are switching to more electric aircraft (MEA) to lower these costs as well as to reduce greenhouse gas emissions. Power electronics are critical elements in MEA, and reliable operation is crucial in this application.

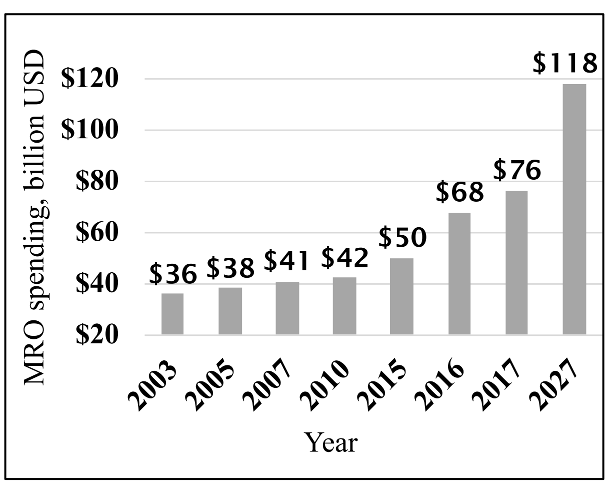


Fig. 1.1: Estimated world MRO spending by the airline industry per specified years in billion USD.

The railway industry is another area where power electronics are mission-critical components. Today innovative technologies contribute to the railway industry's progress towards shorter travel times and improved system efficiency. However, despite the recent progress, problems remain primarily in the areas of logistics and maintenance systems. The maintenance and operation systems are not always capable of identifying and detecting emerging faults, which often results in breakdowns between planned maintenance occasions. The safety issues are particularly critical for metalized polypropylene film (MPPF) capacitors installed in the DC-link of railway power trains [2]. Failures of the MPPF capacitors may lead to catastrophes, such as the Melbourne's explosion in Siemens trains in 2014 [3] and the Guildford's explosion in 2017 [4] that scattered the debris around the train (see Fig. 1.2). Thus, there is a need to ensure that MPPF capacitors do not pose a risk to human lives and to protect the organization of the equipment operation from costly losses. Hence, the railway industry is exploring the prospects of a transportation system with increased reliability and minimized operating costs.

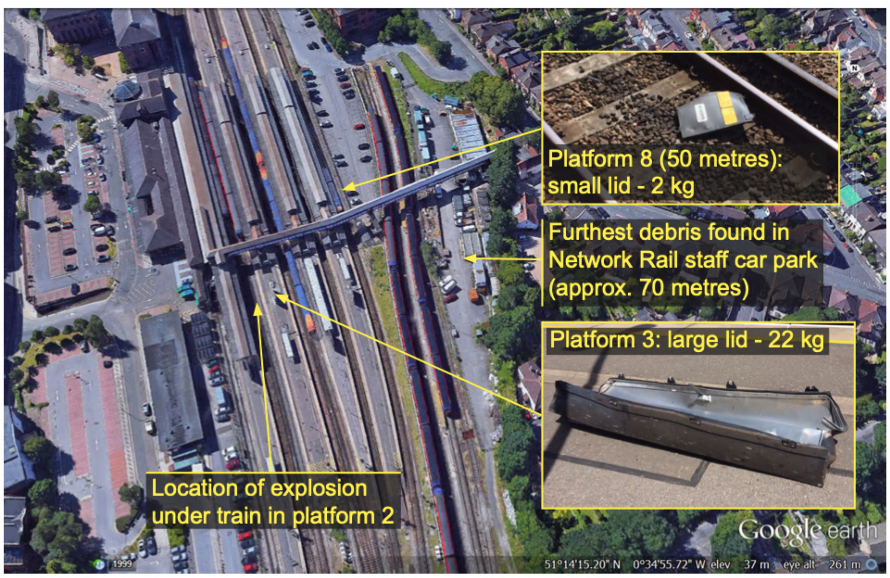


Fig. 1.2: Locations of debris found after the Guildford's explosion inside an underframe equipment [4].

The wind turbine industry is also an example of a power electronics application where unscheduled service incurs substantial penalty costs. The market for renewable energy has significantly grown in recent years, as it provides effective means of dealing with climate change associated with the traditional forms of energy. In 2017, the USA and China's combined wind power capacity made up more than half of the cumulative wind power capacity worldwide [5].

Moreover, in 2017, the USA power market added 9,598 MW of new capacity, which increased the total capacity to 89.077 GW, as Fig. 1.3 demonstrates [6]. These numbers represent a rising interest in the development of wind turbines. However, there exist challenges that adversely impact the widespread application of wind farms. Wind turbines are typically constructed in isolated environments with high humidity and wide temperature fluctuations, where occasional breakdowns are unavoidable. The disruption in the wind turbine operation may not only require an expensive and lengthy repair process, but it may also disrupt and destabilize the entire power grid. Besides, the complicated structure of wind turbines and the turbine parts located at the height of hundreds of meters make it difficult to gain access to and repair. Thus, timely fault prevention and reduction of operation and maintenance costs are the most significant challenges in the wind turbine industry [7].

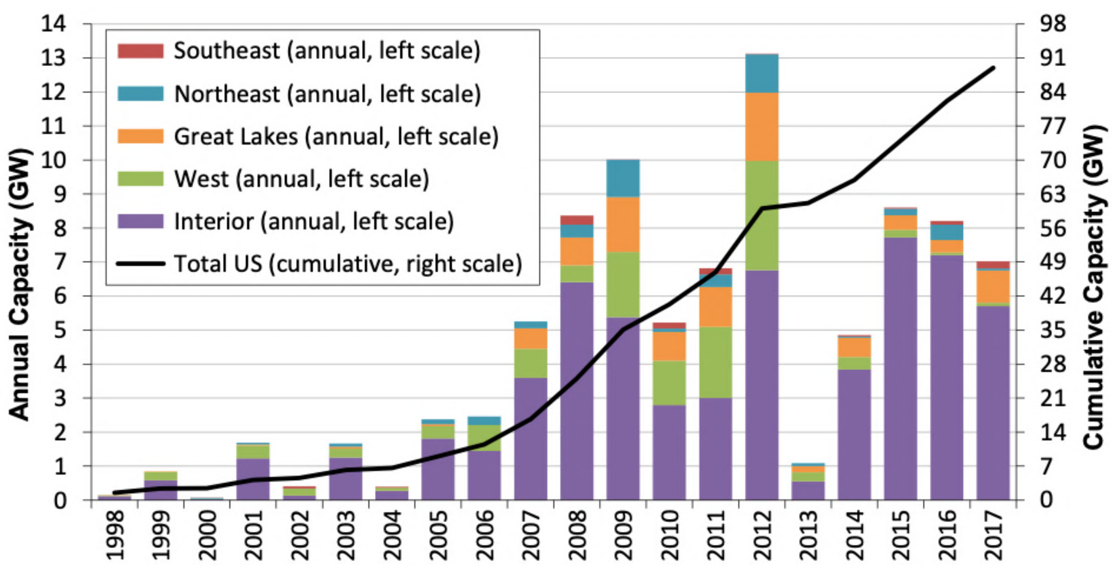


Fig. 1.3: Annual and cumulative growth in U.S. wind power capacity [6].

The concerns listed above are motivating researchers to focus on PHM methods for power electronics and tools that provide real-time information about equipment status [2], [7]-[95]. PHM systems detect, isolate and predict the occurrence, the cause, and the period of system degradation in order to make intelligent decisions about the status of the system and to develop strategic maintenance plans for a critical mission completing or to schedule maintenance for the next high production period [12].

A central theme of PHM systems is CM, which monitors the operating characteristics of a system using non-invasive measurements that do not interrupt the system's functionality. CM uses changes in signal measurements to predict potential failures. Equipment operators use this information to schedule preventative maintenance before a failure occurs in the system [13]. Table 1.1 summarizes the applications, goals, and advantages of CM. According to this Table, there are two main applications of CM. The first one is the safety-critical systems where unpredicted failures can lead to a catastrophe and jeopardize people's safety. The second application is the systems where failures incur financial ramifications. For these systems, CM aims to predict when a component will fail or to identify when degradation in one component performance will affect another component's health. CM is beneficial in its ability to provide a system's health status in real-time without interruption of its operation. It is a cost-effective solution to detect imminent faults at an early stage of their occurrence.

Table 1.1: Applications, Goals, and Advantages of Condition Monitoring

Applications	➢ Safety-critical systems where an unpredicted malfunction may cause a disastrous accident ➢ Systems where unscheduled service incurs substantial penalty costs
Goals	➢ Predict when a component is likely to fail ➢ Predict when degradation in one component performance will affect the life of another component
Advantages	➢ Real-time health status ➢ No interruption in system's operation ➢ Early-stage imminent fault detection ➢ Cost-effective solution

1.2 Problem Statement

E-PHM is a promising CM method based on the analysis of the spectrum of electromagnetic interference (EMI). An EMI signature is unique for a system status, and its change can be a reliable indicator of an equipment fault condition. The main benefit of the E-PHM method is that it can evaluate the total health of a power electronics system using a single EMI measurement. This is possible since different component failures have distinct spectral signatures that manifest in different parts of the EMI spectrum. Therefore, the elimination of individual sensors for each subsystem can reduce the CM implementation costs. Moreover, the diagnostics of a system does not require any additional hardware.

This research focuses on the applicability of the E-PHM approach to a single component in a three-phase inverter system, namely, a DC-link capacitor. It investigates if EMI can be used to estimate the DC-link capacitor health, the frequency ranges of interest, and a reasonable level of accuracy for the E-PHM approach.

1.3 Proposed Approach

This thesis introduces a new approach based on the EMI diagnostic tool to perform the DC-link capacitor CM in a three-phase inverter. The existing E-PHM approach applied to a DC-link in [58] uses line impedance stabilization networks (LISNs) to obtain the conducted EMI by measuring the RF voltage at the LISN's output [96]. This approach is not practical because LISNs have a large size and high cost. Thus, a need exists to implement this method without using LISNs. There are alternative techniques to measure conducted EMI. One of them uses current probes included in most power systems for implementing control [96]. For this reason, the following research measures EMI using a high-bandwidth current transducer (CT), hereby effectively addressing the aforesaid drawback of an existing E-PHM method.

The work [58] demonstrates that the changes in the capacitor's impedance due to the capacitor aging reduce its ability to filter the high-frequency content in the EMI spectrum up to resonant frequency. This thesis proposes and tests the hypothesis that in a higher frequency range (10's MHz) beyond the capacitor's resonant frequency (100's kHz), there exists a relationship between the magnitudes of the EMI harmonics and the capacitor age.

The approach in [58] uses only the Support Vector Machine (SVM) algorithm to classify the input data into capacitor classes corresponding to the capacitor age. This research investigates two Machine Learning (ML) algorithms, namely the SVM and the Artificial Neural Network (ANN), with application to CM of a DC-link capacitor. It also extends the ML classification with the Hidden Markov Model (HMM) to increase the accuracy of the ML prediction of a DC-link capacitor age. This thesis includes the supporting experimental results to evaluate and compare the performance of the SVM and the ANN concerning classification accuracy and to estimate the accuracy improvement provided by the HMM.

In contrast to [58], the CM procedure proposed in this thesis involves digital signal processing (DSP) techniques to determine the capacitor's health based on the spectrum of current. Data is subject to processing using the Fast Fourier Transform (FFT) and spectral smoothing techniques: the moving average filter with a rectangular window (MARF), the moving average filter with a Hanning window (MAHF), the locally weighted linear regression filter (LRF), and the Savitzky–Golay filter (SGF). The processed data forms the features supplied to the SVM and the ANN that classify the signals into the five classes corresponding to the capacitor's health.

The system diagram in Fig. 1.4 shows the E-PHM system for a three-phase inverter. A CT inserted between the DC source and the inverter measures the high-frequency content circulating between the two. LISNs isolate the DC power supply from the inverter to eliminate their influence on the measurements. The first CM stage includes the CT's data capturing and preprocessing using DSP techniques. The FFT converts the acquired time-domain signal to the frequency spectrum. Four different filters smooth the current spectrum: the MARF, the MAHF, the LRF, and the SGF. The next stage performs the initial capacitor health classification using the SVM and the ANN. Finally, the HMM provides the corrected output capacitor health. This research performs the signal analysis offline to demonstrate a proof-of-concept.

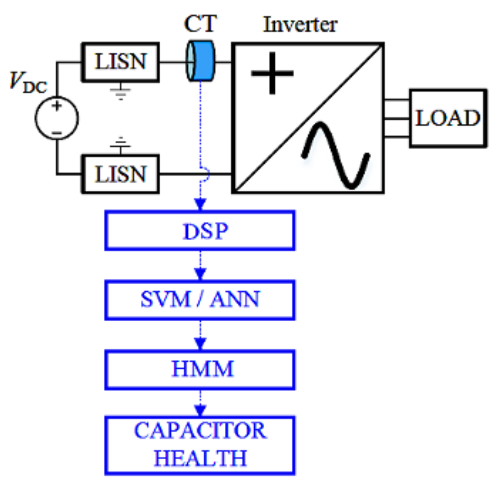


Fig. 1.4: E-PHM system diagram.

This research shows that the E-PHM method using current measurements can successfully estimate the DC-link capacitor health in a three-phase inverter. It demonstrates that a smoothed EMI current spectrum can be a reliable indicator of the capacitor's age. Fed with this data, either the SVM or the ANN can identify the capacitor age class with high accuracy. In addition, the

HMM-based output correction can significantly reduce misclassification errors of both the SVM and the ANN.

The contributions of this thesis are:

1) using a CT instead of a LISN to measure EMI for DC-link capacitor CM in the framework of the E-PHM method;

2) analyzing the EMI spectral content in a higher frequency range beyond the capacitor's resonant frequency;

3) subjecting the input signal to DSP techniques, namely, the FFT to convert the time-domain signal to frequency spectrum and four types of spectral smoothing filters, including the MARF, the MAHF, the LRF, and the SGF;

4) evaluating both the SVM and the ANN algorithms in terms of the DC-link capacitor age classification accuracy;

5) incorporating the HMM with the purpose of the output classification accuracy enhancing.

1.4 Thesis Objectives

The objectives of this thesis are:

- To justify the hypothesis that in a higher frequency range, there exists a relationship between the magnitudes of the EMI harmonics and the capacitor age.
- To determine the optimal DSP strategy for the proposed approach.
- To evaluate and compare the performance of the SVM and the ANN algorithms with respect to the capacitor age classification accuracy.
- To evaluate accuracy improvement achieved by incorporating the HMM.

1.5 List of the Chapters

The rest of this thesis is organized as follows.

Chapter 2 provides the literature review on the developed CM methods for power systems and DC-link capacitors.

Chapter 3 covers the theory on the aging mechanisms of capacitors, conducted EMI's sensitivity to the characteristics of DC-link capacitors aging, the DSP algorithms used in the proposed solution, as well as the principles of the SVM and ANN learning, along with the HMM technique.

Chapter 4 describes the procedure of acquiring and processing experimental data and provides a discussion of the results.

Chapter 5 concludes the thesis with the results from this research and provides the issues to be addressed in the future work.

Appendix contains the MATLAB code used for data processing in this thesis.

Chapter 2 Literature Review

This chapter starts with a literature review of the different non-invasive CM approaches for power electronic applications. Then, it covers classified by groups CM methods applied to DC-link capacitors. It discusses the main principles of these methods, along with the advantages and disadvantages of their implementation.

2.1 Condition Monitoring Methods in Power Electronics

2.1.1 Signature Analysis

Each CM method for a power system analyzes the data signatures associated with a deterioration or an anticipated failure of the system components. Based on the categories of the monitored signals, there are five groups of health assessment strategies: vibration signature analysis (VSA), thermal signature analysis (TSA), ultrasonic signature analysis (USSA), acoustic signature analysis (ASA), and electrical signature analysis (ESA). These types of analysis commonly involve DSP techniques for measured data processing. Spectral techniques and digital filters allow to extract specific characteristics from the monitored signals that correlate with a system's health status.

Vibration Signature Analysis

VSA measures the physical vibration of rotary machine systems, such as machine tools, wind turbines, and electric motors. Every machine has a vibrational characteristic that forms the signature of that machine. When these systems operate normally, the signature remains unchanged. However, anomalies and failures in machine parts, bearings, rotors, and shafts change the machine's vibration signature. By detecting these changes, it is possible to estimate a machine's remaining useful life (RUL) as well as to diagnose a failure mechanism.

The methods that analyze time-invariant stationary vibration signals, such as periodic vibrations caused by a worn-out part, commonly use spectral techniques based on the FFT [14]. The authors in [15] performed spectral analysis using FFT on the stator current to detect bearing damage in induction machines. In [16], the vibration spectrum of faulty and healthy roller bearings obtained by the FFT determined motor condition and maintenance requirements.

The analysis of transient non-stationary signals that occur due to an abrupt breakdown of a machine part relies on time-frequency techniques. A signal processing technique presented in [17] used the Hilbert-Huang Transform (HHT) as a tool for the machine CM. Another method introduced in [18] also applied the HHT to extract features from non-stationary motor vibrations to diagnose bearing degradation and to predict faults. Several authors used the Wavelet Transform (WT) to extract time-frequency features from the highly variable signals. R. Yan et al. [19] applied the Discrete Harmonic Wavelet Packet Transform (DHWPT) to vibration signals measured from a bearing test bed. The early detection technique based on the Adaptive Wavelet Transform (AWT) proposed in [20] detected fatigue damage on rolling element bearings. The authors of [21] investigated the application of the Continuous Wavelet Transform (CWT) and the Discrete Wavelet Transform (DWT) to wind turbine signals for fault detection in direct-drive generators. An approach in [22] introduced a Wavelet-based adaptive filter for the CM of wind turbines.

Thermal Signature Analysis

TSA analyzes the thermal signature generated from equipment to identify component degradation or equipment failure. Faulty machinery, corroded electrical connections, and worn components are detectable by measuring differences in temperature [23]. Infrared thermography (IRT) is a commonly used temperature monitoring technique that uses images of infrared radiation released from a system to find any abnormal heat patterns or thermal anomalies in an electrical machine [24]. The underlying principle of the IRT is based on Planck's law and Stefan-Boltzmann's law. These state that objects with temperatures above 0 K emit electromagnetic radiation in the infrared region of the electromagnetic spectrum and that there exists a correlation between the infrared radiation intensity and the object's temperature [25].

There are two groups of IRT methods: passive and active IRT [26]. Passive IRT relies on the temperature gradients in the materials and structures, and its main application is the CM of electrical and mechanical systems, where abnormal temperature profiles indicate a potential problem [27]. Active IRT generates thermal contrasts externally. This process includes inducing heat flow on the investigated component and capturing the resulting heat flow to detect its defects [28]. Active IRT is suitable where passive thermography cannot identify the thermal gradient caused by very small defects.

Concerning temperature measurements, there are two IRT approaches, quantitative and qualitative. Quantitative approach analyses the object's temperature with respect to the ambient temperature, while the qualitative approach uses the temperature of the other equipment parts with similar conditions as a reference [25], [27].

CM methods using IRT can be manual and automatic. Manual methods imply a manual examination of the device's thermal images to infer the system's health. In turn, automatic methods use Artificial Intelligence (AI) algorithms to classify the data according to the system's health status. Intelligent thermographic diagnostics applied to surge arrests in [29] used neuro-fuzzy fault classification. The authors of [30] investigated the application of the ANN to determine the condition of the electrical equipment by extraction of the temperature values from each pixel of the thermal image. An approach introduced in [31] used the SVM learning algorithm for infrared image analysis in high-voltage power equipment diagnostics.

Ultrasonic Signature Analysis

USSA detects defects in electrical equipment, bearings, and rotating parts by measuring ultrasonic emissions with non-contact and contact ultrasound instruments [32]. Many physical events produce ultrasound, defined as the sound waves with a frequency above the human hearing limit of 20 kHz [33]. Analysis of the ultrasonic frequencies translated to the audible spectrum by the ultrasound transducers at different intensity or dB levels indicates if a system operates correctly or incorrectly [34]. It is possible to use USSA either alone or in combination with VSA and IRT.

There are two types of USSA techniques, passive and active ones. Passive techniques focus on the ultrasound produced in the monitored component by a physical process. Passive USSA method that uses contact measurements is valuable for bearing and gear fault detection [33]. Non-contact passive USSA is useful for predictive maintenance of heat exchangers, boilers and condensers [35] and for electrical discharge detection in power equipment [34]. Active USSA technique uses an ultrasound beam transmitted to a physical structure for further analysis of surface and subsurface discontinuities. Based on different characteristics of the obtained ultrasonic signal, such as its flight time, amplitude, and frequency, the technique evaluates the damage's depth [36], [37].

Acoustic Signature Analysis

ASA is a CM method based on the analysis of the sound waves emitted from the equipment due to defects or discontinuities, which occur before the structural failure. The acoustic emissions (AE) in electrical machines can originate from friction, material loss, turbulence, and leakage. [38], [39]. The AE spread through the material surface in the form of Rayleigh waves [38], and AE sensors measure the deviations in these waves. The sensors placed on the material surface collect the monitored data. The analysis of the discontinuities detected from the acquired signal characteristics estimates the existing defects in the structure.

The parameter-based analysis and the waveform analysis are the two approaches to perform ASA. The parameter-based approach only records and investigates the particular signal parameters, such as energy and amplitude, but not the signal itself [40]. This method reduces the amount of stored data and increases the analysis speed. However, as this approach loses track of the information about not recorded signal parameters, it limits the available defect characteristics for inferring [41]. In contrast, the waveform analysis examines the complete signal. This approach enables better data interpretation capability because it captures all the signal characteristics. This technique involves spectral analysis using such DSP methods as WT [42]-[44].

Electrical Signature Analysis

ESA is a set of CM methods based on the principles of deviations in electrical parameters of components for the identification of faults and defects in electrical equipment. A central theme of the ESA is measuring current and voltage signals and matching their trends with failure patterns.

Spectral analysis commonly extracts current signal characteristics. The CM in [45] identified variations in the fifth harmonic's amplitude of the inverter current to detect solder fatigue in a power module. An approach combining the Short-Time Fourier Transform (STFT) and the WT detected faults in power transformers during impulse testing [46]. Another method directly applied the FFT upon current signals for monitoring of motor bearings in wind turbines [47]. The generator current signal processing involved resampling, FFT, and multiscale filtering (MSF) procedures to obtain the power spectrum used to extract the fault characteristic frequencies.

CM of fast switching power semiconductors use on-state voltage measurements. The deviations of the on-voltage from a healthy behavior determine faulty conditions of the system [48], [49]. In [50], an intelligent power module periodically measured on-state voltage and current to indicate the integrity of the bond wires and the level of deterioration due to power cycling.

E-PHM is a method to estimate the hardware's health by analyzing the EMI's spectral content. Timperley et al. [51] applied E-PHM to assess the health of high-voltage apparatus such as generators, induction motors, transformers, and switchgear. The data collected in real-time revealed that an EMI signature is unique for a system status, and its change is a reliable indicator of equipment fault condition. EMI is highly sensitive to changes in device parameters. This property demonstrated its efficiency in estimating the condition of silicon carbide metal oxide semiconductor field effect transistors (MOSFETs) in a synchronous buck converter [52]. The experimental results demonstrated that the EMI's spectrum amplitude in a high-frequency range increased with the MOSFETs aging. The current-based online E-PHM technique presented in [47] provides a solution for the diagnosis and prognosis of the RUL of wind turbines.

2.1.2 Artificial Intelligence

In recent years, there has been a rising interest in AI methods for CM [53], which are suitable for any monitored signals discussed above. Such methods focus on classification schemes, which attempt to classify the system's operational status by analyzing its representative device condition. ML methods, such as the SVM and the ANN, demonstrated their efficiency in the fault diagnosis of power electronics equipment [18], [20], [54]-[60]. The authors of [55] presented advanced self-commissioning ANN training algorithms to detect changes in the measured currents and voltages for the electrical machine fault diagnostics. In [56], the ANN identified the changes in capacitor equivalent series resistance, diode turn-on voltage, and diode resistance. In [18], the SVM diagnosed bearing degradation using the features extracted from motor vibrations. An SVM-based approach described in [57] identified the type of faults in an oil-immersed power transformer.

The HMM is a powerful statistical technique first introduced in [61] and successfully realized in a range of CM applications for power electronics [62]-[65]. The primary aspect of these applications is the existence of an unobservable state transition process, and the observations are available in the form of a state-dependent sequence. The knowledge about probabilistic relationships between the states and the observations allows us to find the state sequence corresponding to the given observation sequence. In [62], the time-frequency representation (TFR) extracted features from the induction motor's current and vibration. The pre-trained HMM accepted these features, and its probabilities monitoring provided an online fault detection and diagnostic of an induction machine. Yan et al. [63] developed an HMM-based fault diagnostics method for power electronic circuits. An autoregressive model extracted the features of the states in a three-phase rectifier circuit. The HMM recognized the subsequently trained fault models. The technique in [64] applied the HMM to detect multiple faults in a DC-DC boost converter. This diagnostic selected the highest probability of the HMM and matched it to a corresponding fault pattern.

2.2 Condition Monitoring Methods for DC-link Capacitors

Capacitors are common devices used for DC-links in the majority of power applications. Nevertheless, they remain one of the least reliable components in power electronics [66], [67]. This reason accounts for a substantial amount of research on the DC-link capacitors' CM methods [2], [68]-[95].

Effective CM requires establishing the dependence of the DC-link capacitance's age with signals monitored externally. H. Soliman et al. [68] classified the CM methods for capacitors in power electronics from the different perspectives illustrated in Fig. 2.1. From the availability aspect, CM methods can be online, quasi-online, and offline. The online methods are of the primary interest as they allow to avoid operation interruption of a system. Lifetime indicators separate the methods based on two artifacts: an increase in the equivalent series resistance (ESR) with the capacitor age; a decrease in the capacitance with the capacitor age. The final perspective is the methodologies that estimate the values of a specific indicator. It determines three categories, namely, capacitor ripple-current sensor-based methods, circuit model-based methods, and data and advanced algorithms-based methods.

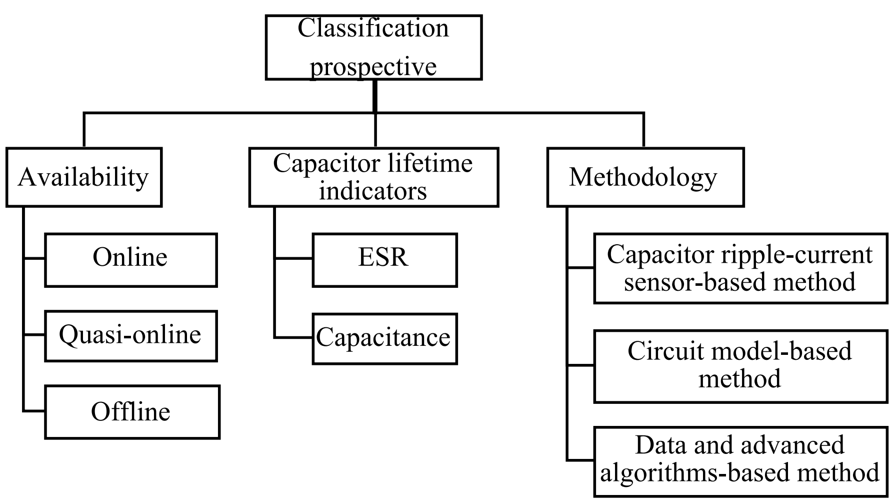


Fig. 2.1: Classification of capacitor CM methods.

Capacitor Ripple-Current Sensor-Based Methods

Capacitor ripple-current sensor-based methods use measurements of the capacitor voltage and ripple current [69]-[83]. Some of the methods use direct current sensors to obtain the capacitor ripple current [69]-[71]. A non-invasive method proposed in [69] evaluated the DC-link capacitance in a single-phase inverter. It used second-harmonic oscillations in the DC-link's voltage and current to predict capacitor health using the built-in control sensors. This approach is convenient for practical utilization since it does not require any additional hardware. L. Ren et al. [70] presented an online technique for aluminum electrolytic capacitors monitoring. It estimated the ESR for a boost converter's output capacitor by analyzing the output voltage ripple and the inductor current denoised using WT. The proposed ESR calculation model is beneficial in its low computational complexity and suitability for different operation conditions of the converter. A. Imam et al. [71] passed the measured capacitor current through a band-pass filter (BPF) and computed its root mean square (RMS) error to estimate the ESR of an electrolytic capacitor in a specific frequency range. An advantage of this approach is a simple analog circuit required for data acquisition.

Another group of methods obtains the information at a certain frequency by the injection of external signals [72]-[79]. A.G. Abo-Khalil et al. [72] estimated capacitance in real-time by injecting an external low-frequency AC voltage into the DC-link capacitor of a three-phase AC/DC/AC pulse width modulation (PWM) converter. The Support Vector Regression (SVR) analysis processed the AC power component extracted from the input to determine the value of the DC-link capacitance. A drawback of this technique is that it requires offline training. Another approach estimated the ESR of an electrolytic capacitor in a PWM converter [73]. It calculated the capacitor's loss from the AC current and voltage ripple. The current measurements were carried out using a PCB-located shunt resistor inserted in the DC-link power wiring, which in principle may lower the reliability of the overall converter. Thus, another current-sensing device should be considered for the practical implementation of this methodology. A quasi-online method in [74] performed CM of aluminum electrolytic capacitors used in the DC-link of solar inverters. This method injected low-frequency harmonic current in the grid in the absence of solar irradiance to evaluate capacitor impedances at different frequencies. It determined the ESR and capacitance from the impedance measurements using the least mean square (LMS) algorithm. A drawback of this approach is that the estimation of the parameters of interest is feasible only when the invertor is not operating. Some methods used spectral techniques, such as the Laplace transform [75] and the Discrete Fourier Transform (DFT) [76]-[78] to estimate the values of capacitance and ESR. An online method in [79] estimated capacitance for a DC-link capacitor in a three-phase AC/DC/AC PWM converter. It injected a controlled AC current to the converter's input and used the DC output AC ripples extracted with a second-order BPF to calculate the capacitance by applying the recursive least squares method. This method achieved a high capacitance estimation accuracy of 99.74%. However, a drawback of this approach is associated with the current injection that requires extra effort and additional filters. Table 2.1 provides the estimation accuracy of different capacitor ripple-current sensor-based methods.

Table 2.1: Accuracy of Capacitor Ripple-Current Sensor-Based Methods

Accuracy range	Accuracy	Reference
<90%	70% 88% 89%	[80], [83] [81] [76], [82]
90%-99%	90% 95% 96.8% 97.4% 99.74%	[71] [75] [72] [77] [79]

Circuit Model-Based Methods

Circuit model-based methods estimate the DC-link capacitor parameters using the operational principles of a circuit [2], [84]-[91]. Some methods within this group do not use signal injection [2], [84]-[86]. A. Wechsler et al. [84] obtained the ripple current of a DC-link capacitor in aerospace drives based on the operational principle of a PWM converter. An online method proposed in [85] applied the analysis of a boost converter supplied from a photovoltaic panel to determine a DC-link capacitor health. This method avoids using additional hardware, since the built-in the system microcontroller implements CM. A technique presented in [2] performed CM of MPPF capacitors installed in the railway power trains DC-link. It measured the DC-link voltage directly from the traction control block and estimated the capacitance from this signal using the LMS algorithm. This approach is advantageous in avoiding any additional sensors for measurements and the simplicity of the DSP stage. An online time-frequency analysis based on the WT estimated the ESR of the output capacitor in a boost converter [86]. This method also does not require any additional current sensors.

Another group of methods involves signal injection [87]-[91]. Current injection was applied for fault diagnostics of a DC-link capacitor in a three-phase PWM converter using an online estimation of ESR in [87] and in a single-phase PWM converter using an evaluation of capacitance in [88]. The same approach determined the health of DC-link capacitors in AC machine drives [89], submodule capacitors in modular multilevel converters [90], and DC-link capacitors in a drive system for electric vehicles [91]. One shortcoming of the signal injection methods is the requirement of additional hardware to generate external signals. Table 2.2 gives the values of estimation accuracy of different circuit model-based methods.

Table 2.2: Accuracy of Circuit Model-Based Methods

Accuracy range	Accuracy	Reference
90%-99%	91% 95% 96.4% 98% 98.8%	[84] [85], [86] [88] [89] [2]
>99%	99.82% 99% 99.7%	[87] [90] [91]

Data and Advanced Algorithms-Based Methods

Data and advanced algorithm-based methods use AI algorithms that receive measured signals as an input and output the DC-link capacitor health parameters. In [92], the ANN estimated the DC-link capacitance in a diode-bridge front-end three-phase motor drive based on the existing control information. This method does not require any additional hardware or external signal injection and has a relatively low maximum estimated error of 0.5%. The methodology described in [93] estimated the DC-link capacitance in a three-phase motor drive with an ANN algorithm. A switchable capacitor bank emulated DC-link capacitor aging. This approach measured the DC-link voltage, extracted its AC ripple and 300 Hz harmonic, and supplied this data to the ANN. The ANN estimated the DC-link capacitance with a 98.7% accuracy. This estimation accuracy demonstrated the ANN performance on only four test cases, which is not enough to conclude the ANN's predictive ability. Thus, a larger dataset should be tested to obtain more reliable information about the overall method efficiency. The authors of [94] applied an adaptive neuro-fuzzy inference system (ANFIS) algorithm for capacitor aging detection for the DC filters in the power electronic converters. The input voltage of the converter and the voltages across the DC filters served as inputs for the ANFIS. The algorithms output indices corresponding to the capacitor aging fault, and the estimation accuracy was 99.5%.

E-PHM method applied to electrolytic capacitors in [95] predicted failures in a boost converter's DC-link capacitor by monitoring the changes in the frequency spectrum of capacitor current and voltage ripple. Variations of the harmonic magnitudes over the time reflected the changes in capacitor parameters due to its aging. The analysis of the FFT spectrum conducted with the time-average technique estimated the RUL of the component. M. Boubin et al. [58] advanced the EMI diagnostic tool with a ML-based predictor. The proposed SVM algorithm trained on the three-phase inverter's EMI spectrum data, and its output estimated the age class of a DC-link capacitor. This approach relied on the concept of correlation between the changes in a capacitor's impedance at the resonant frequency and the capacitor's age. An attractive feature of the E-PHM method is that a single EMI measurement can predict the health of multiple components within the hardware. An advantage of E-PHM over many traditional approaches is that it reduces the amount of equipment needed to perform CM, saving space and money. Table 2.3 provides the values of the estimation accuracy of different data and advanced algorithm-based methods.

Table 2.3: Accuracy of Data and Advanced Algorithm-Based Methods

Accuracy range	**Accuracy**	**Reference**
<90%	88.8%	[58]
90%-99%	98.7%	[93]
>99%	99.5%	[92], [94]

2.3 Summary

This chapter investigated different CM methods used in power electronic systems to identify the most beneficial technique for DC-link capacitors health monitoring. DC-link capacitors are especially attractive for CM due to their widespread applications in the power electronic conversion systems for motor drives, wind turbines, automotive industry, electric vehicles, and aerospace. The type of the appropriate analysis depends on the monitored signal. Vibration signal does not correlate with the health characteristics of a capacitor. Thus, VSA is not applicable for the DC-link CM. USSA and ASA have not been investigated for the DC-link CM. IRT method requires additional equipment, such as thermal imaging cameras, to capture the infrared radiation released from a system. These cameras are very expensive and not cost-effective for most applications. ESA is a common approach to determine a DC-link capacitor's health. Most methods based on ESA rely on the spectral analysis of the system's current and voltage signals. AI algorithms that process this data result in a high estimation accuracy of the DC-link capacitor parameters. E-PHM is a promising ESA method for condition evaluation of the DC-link capacitors. Its main advantage is that a single EMI measurement can predict the health of multiple components within a system, reducing the amount of hardware needed to perform CM and decreasing its costs.

Chapter 3 Background

This chapter begins with a review of the degradation behavior of MPPF capacitors. Next, it analyzes the EMI's sensitivity to the DC-link capacitor aging. It then presents an LTSPICE model that simulates a capacitor aging impact on the EMI and presents simulation results that validate the analysis. After that, it describes the DSP strategies to convert the input data to the frequency domain and smooth the resulting spectrum. Finally, it discusses how to utilize ML models and the HMM for capacitor age classification.

3.1 Degradation Behavior of Metallized Polypropylene Film Capacitors

MPPF capacitors consist of dielectric polypropylene films, the electrodes represented by thin metal layers (20 to 100 nm), evaporated onto the surface of the polypropylene films, and the terminals made of a tinned wire. The equivalent circuit of a capacitor consists of a capacitance (*C*), an equivalent series resistance (R_{ESR}), and an equivalent series inductance (L_{ESL}), as shown in Fig. 3.1.

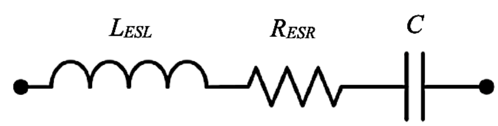


Fig. 3.1: Capacitor equivalent circuit.

The magnitudes of C and R_{ESR} vary with temperature and frequency. Capacitance varies with temperature within a range of temperatures between its upper and lower limits. The characteristic of the capacitance curve gradient is the capacitance temperature coefficient α_C, expressed in units of 10^{-6}/K and given by:

$$\alpha_C = \frac{C_2 - C_1}{C_3(T_2 - T_1)}, \tag{3.1}$$

where C_1 is the capacitance measured at temperature T_1, C_2 is the capacitance measured at temperature T_2, and C_3 is the reference capacitance measured at (20 ±2) °C. The temperature coefficient of MPPF capacitors is $-250\cdot10^{-6}$ 1/K. Fig. 3.2 shows the capacitance temperature characteristic expressed as a percentage of the capacitance change to the reference capacitance.

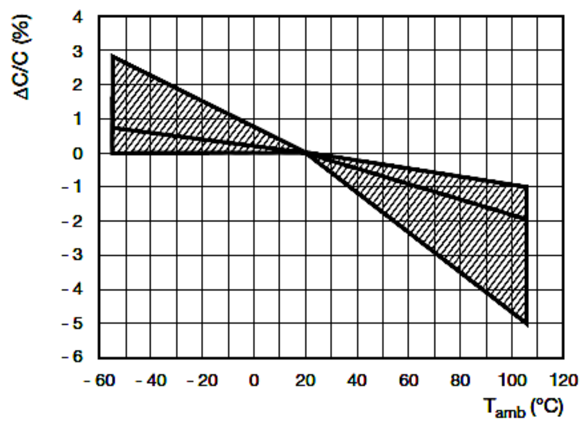


Fig. 3.2: Temperature characteristic of capacitance for MPPF capacitors [97].

The ESR varies with the frequency f as:

$$R_{ESR} = \frac{\tan \delta}{2\pi f C}, \tag{3.2}$$

where δ is the dissipation factor. Temperature does not largely affect the dissipation factor of MPPF capacitors, and hence, does not affect the ESR. For the capacitor model in Fig. 3.1, equation (3.3) defines the impedance Z, and equation (3.4) computes the resonant frequency f_{res}.

$$Z = \sqrt{{R_{ESR}}^2 + (2\pi f L_{ESL} - \frac{1}{2\pi f C})^2}, \tag{3.3}$$

$$f_{res} = \frac{1}{2\pi\sqrt{L_{ESL} C}}. \tag{3.4}$$

Fig. 3.3 shows the frequency characteristic of the impedance for MPPF capacitors. It demonstrates that the impedance strongly depends on the frequency.

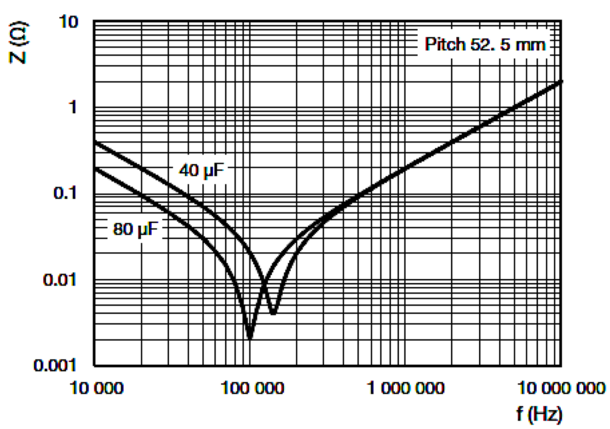


Fig. 3.3: Frequency characteristic of impedance for MPPF capacitors [97].

MPPF capacitors have a self-healing ability [98]. Self-healing is the process of restoring the electrical properties of a capacitor after a local breakdown. High-density current in the breakdown region provokes a discharge. This discharge produces heat, which, in turn, generates electrode layer evaporation. Self-healing isolates the defective part from the capacitor, which prevents its failure.

As discussed in [2], the most common reasons for MPPF capacitors failures are the breakdown of the dielectric film caused by poor self-healing, the separation of the "sprayed ends" from the capacitor roll, and the extreme capacitance losses. In some cases, a capacitor may fail to an open-circuit state within just several hundred milliseconds. However, one of the most hazardous situations occurs when a capacitor fails into a resistive state. In this case, the heat dissipation through the capacitor causes the dielectric film to melt, which releases hydrocarbon-based gases. In several document cases, the hydrocarbon gases filled an enclosed space and resulted in an explosion that created substantial damages [3], [4]. This explosion detached the case covers from the capacitor, threw them far away from the train, and significantly distorted the traction block.

Capacitor aging does not affect the value of L_{ESL} because the capacitor's geometry remains the same [99], but the values of C and R_{ESR} change [67]. Two aging mechanisms of the MPPF capacitors explain these changes. The capacitance changes because overvoltage conditions produce localized breakdowns in the dielectric film. The breakdowns create a short circuit between the electrodes. Self-healing eliminates this short circuit and leads to a slight capacitance drop as a result of partial destruction of the electrodes and dielectric [100]. As self-healing events accumulate over time, the capacitance gradually degrades. The on-resistance changes due to the demetallization of the capacitor's electrodes that result from the presence of high currents. Excessive evaporation inside the capacitor also causes corrosion on the metalized layers [101]. The electrode material losses increase R_{ESR}, thereby increasing the loss within the capacitor.

The impedance changes due to the capacitor aging are demonstrated in Fig. 3.4, where new capacitor characteristics are: C = 50 μF, R_{ESR} = 6 mΩ, L_{ESL} = 35 nH, and aged capacitor characteristics are: C = 45 μF, R_{ESR} = 19.4 mΩ, L_{ESL} = 35 nH. As the aged capacitor has higher R_{ESR} and lower C, its impedance up to f_{res} is higher than the one for the new capacitor. Above f_{res}, where L_{ESL} is dominant, the impedances are almost the same. Also, f_{res} of the aged capacitor is higher than f_{res} of the new capacitor [58].

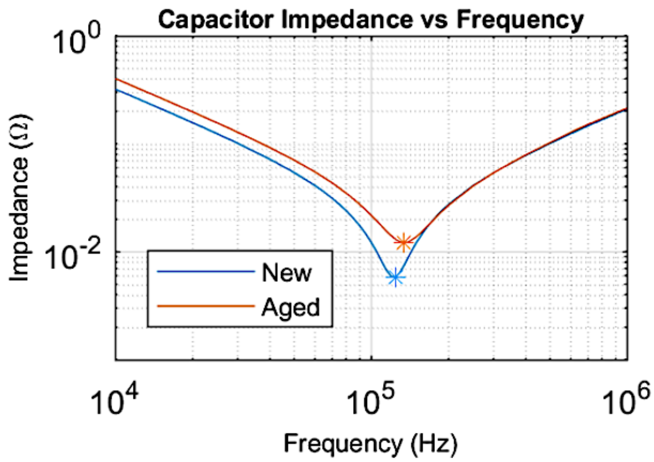


Fig. 3.4: Frequency characteristic of impedance for new and aged capacitors [58].

3.2 E-PHM System for DC-link Capacitor Condition Monitoring

The changes in the capacitor's impedance up to the resonant frequency (100's kHz) reduces its ability to filter the EMI high-frequency content. The authors of [58] used this trait to estimate the DC-link capacitor's health in a three-phase inverter. In contrast to [58], the following research analyzes the spectral content in a higher frequency range (10's MHz) beyond the capacitor's resonant frequency. In this band, the parasitic capacitances resonate with the stray inductances in the commutation loop to create oscillations after switching events. As the capacitor ages and R_{ESR} increases, the damping factor (ζ) increases, which reduces the peak overshoot on the MOSFET's drain-source voltage. The capacitance also decreases, which increases the resonant frequency of this second-order system. The analysis below explains this phenomenon in better detail.

A simplified lumped-parameter module of a three-phase inverter in an E-PHM system appears in Fig. 3.5. A CT measures the EMI current circulating between the DC source and the inverter. LISNs isolate the DC power supply from the inverter. The DC-link capacitor has a DC capacitance (C_{DC}), an equivalent series resistance (R_{ESR}), and an equivalent series inductance (L_{ESL}). Ideal switches (S_i) represent the six power devices in the inverter. Each switch has an output capacitance (C_{OSS}) and an inductance (L_S) that models the parasitics included in the packaging and the printed circuit board (PCB). The parasitic capacitance (C_S) that occurs between the heatsink and the device's die is connected between the inverter's output and ground. The output of the inverter connects to an R-L load represented by R_{LOAD} and L_{LOAD}. A stray inductance (L_{BUS}) exists between the DC-link capacitor and the LISN. The final elements are a LISN's capacitance (C_{LISN}), an inductance (L_{LISN}), a resistance (R_{LISN}), and a filter capacitance (C_{f}). The model in Fig. 3.5 represents a limited number of elements. More parasitics exist, but they are not included to simplify the analysis.

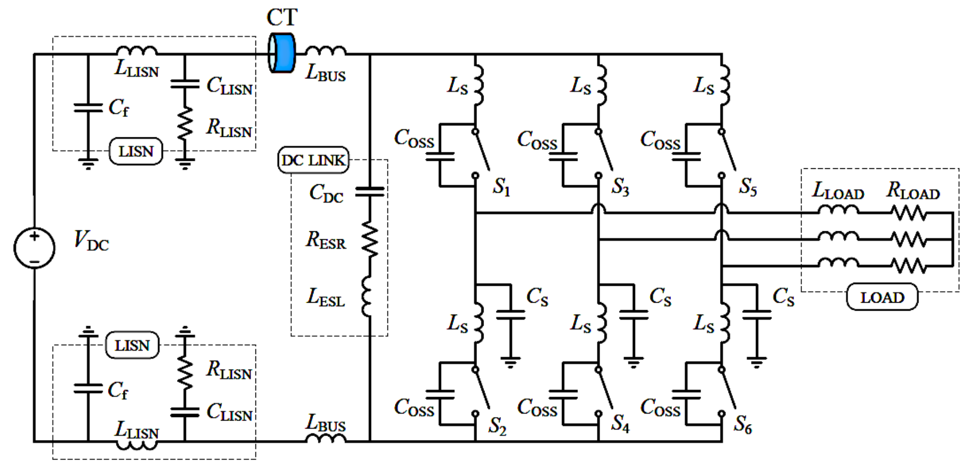


Fig. 3.5: Lumped-parameter model of a three-phase inverter with an R-L load.

Inverters are prone to ringing across the drain-source terminals of a transistor due to parasitic inductances (L_{S} and L_{ESL}) resonating with the device's C_{OSS} [102]. The ringing frequency depends on the operating state of transistors and can be problematic [103]. A second-order circuit is formed between the DC-link capacitor, C_{OSS}, L_{S}, and L_{ESL}. R_{ESR} provides damping for this circuit and limits the peak value of the overshoot. Since R_{ESR} increases as the capacitor ages, the highest current peak will be observed for the capacitor with the greatest health, and it will gradually decrease as the capacitor is aging.

A simplified high-frequency version of Fig. 3.5 appears in Fig. 3.6. By applying several alterations to Fig. 3.5's model, one arrives at the model shown in Fig. 3.6. First, the source (V_{DC}) is eliminated because L_{LISN} appears as an open circuit for the frequency ranges of interest. The DC-link capacitance assumes its position, and because C_{DC} is assumed to be very large, it is replaced with a stiff voltage source (v'_{dc}) that has the same voltage as the DC-link.

High-frequency analysis enables a short circuit to replace C_{LISN}. Inverter loads are inductive in many applications (e.g., motor drives, grid-tied inverters). Therefore, L_{LOAD} is approximated as a constant current source during a single switching cycle, and it becomes an open circuit for this analysis.

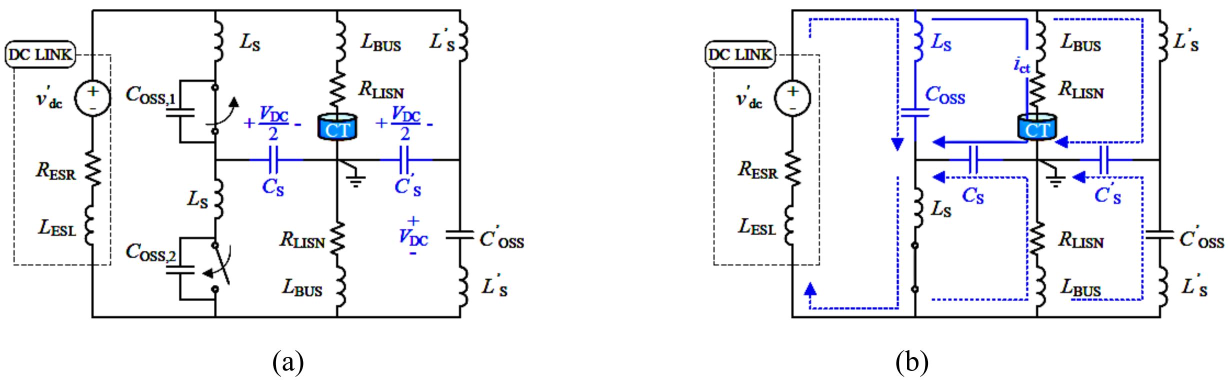


Fig. 3.6: Simplified high-frequency model to estimate the EMI (a) before and (b) after S_2 turns on (S_3 and S_5 are on for both states).

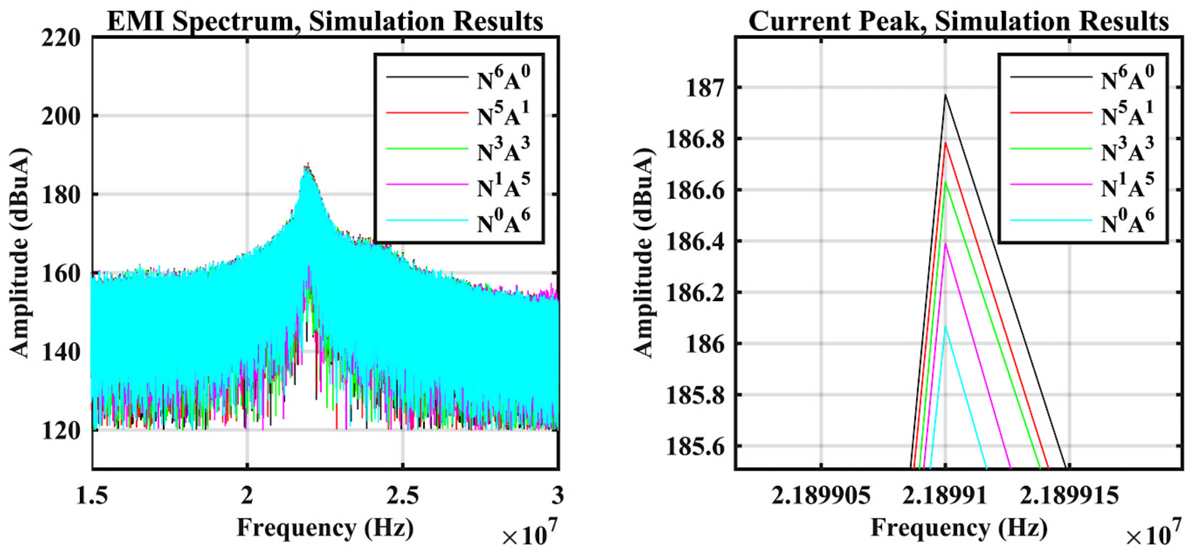
Fig. 3.6a captures the instant just before S_1 turns off, while both S_3 and S_5 are on. C'_S, C'_{OSS}, and L'_S represent the parallel combination of the elements in their respective branches. If both branches are identical, these capacitors will be two times higher, and the inductance will be one-half.

Finally, Fig. 3.6a assumes S_1, S_3, and S_5 have been *on* long enough for all the transients from the previous switching transition to have decayed. The voltages on C_S and C'_S are positive ½ V_{DC}, while C'_{OSS} is charged to V_{DC}. The voltage across output capacitors ($C_{OSS,1}$ and $C_{OSS,2}$) of S_1 and S_2 are 0 V and V_{DC}, respectively.

S_1 transitions from *on* to *off* at the same instant as S_2 changes from *off* to *on*. It occurs when current is freewheeling through S_1, and a gate signal is applied to S_2. Under these conditions, S_2 is hard-switched. As $C_{OSS,1}$ charges and C_S discharges, current circulates through each branch of the circuit, including the one containing the CT. $C_{OSS,1}$ oscillates with L_{ESL} and L_S, which impacts the mesh current (i_{ct}) formed by $C_{OSS,1}$ C_S, and the CT. As R_{ESR} increases, the peak of i_{ct} decreases correspondingly.

We created the lumped-parameter circuit in Fig. 3.5 in LTSPICE to model this behavior. We formed the DC-link with six capacitors connected in parallel. Each of them we discretely changed from a "New" one to an "Old" one to represent the DC-link's gradual aging with five capacitor classes: N^6A^0, N^5A^1, N^3A^3, N^1A^5, N^0A^6, where the upper indices determine the number of "New" (indicated as N) and "Aged" (indicated as A) capacitors in the DC-link. For the simulation, we selected the values of the DC-link's parameters in accordance with the ones specified in the experimental part. Section 4.2 provides the choice of the classes along with their detailed description.

Fig. 3.7 shows the simulation results. It is not evident in Fig. 3.7a that a difference exists between the current signatures for the capacitor classes. However, the zoomed view in Fig. 3.7b clearly illustrates this phenomenon. The magnitudes of the oscillation peaks decrease as the capacitor age increases. This relationship between the current spectral content behavior and the capacitor age is consistent with the theoretical analysis, and it supports the initial hypothesis. As the relative proximity of the current spectrum for different classes may be problematic for efficient ML classification, this research considers smoothing DSP techniques to obtain more distinctive signal trends.



(a) (b)

Fig. 3.7: Simulation results for five capacitor classes: (a) EMI spectrum near the resonant frequency and (b) current peaks for different capacitor classes.

3.3 Digital Signal Processing

Fast Fourier Transform

The FFT is a conventional technique for harmonic analysis [104]. In this research, we use the FFT to obtain the EMI spectrum of the acquired current signal. As the FFT of a real-valued current signal is symmetrical around DC, we analyze only the positive half of the spectrum to classify circuit classes. For the N-point time-domain signal acquired at sampling frequency f_s, we calculate the number of frequency points N_{FFT} in a single-sided spectrum using (3.5), and define the frequency resolution f_r by (3.6):

$$N_{FFT} = \frac{N}{2}, \tag{3.5}$$

$$f_r = \frac{f_s}{N}. \tag{3.6}$$

The stage of signal preparation for subsequent ML processing frequently involves smoothing. Smoothing algorithms can substantially decrease the noisy data variance and help to identify an underlying signal trend. This work examines four different filters: the MARF, the MAHF, the LRF, and the SGF. We chose these filters because they are common for data smoothing in DSP applications in [105] as well as in the ML data preprocessing [106].

Moving Average – Rectangular Window

The moving average algorithm [107] is one of the most frequently implemented smoothing techniques in DSP. The MARF generates an output signal by taking the average of input signals within a sliding window as:

$$y_i = \frac{1}{2M+1} \sum_{j=-M}^{M} x_{i+j}, \tag{3.7}$$

where x is the input signal, y is the output signal, and $2M+1$ is the length of the window. The desired outcome determines the size of the window. Larger windows show the main trends in the data, while smaller windows capture trend changes better.

Moving Average – Hanning Window

The window type determines the weight coefficients for signals in the moving average smoothing. In this study, we consider both rectangular and Hanning windows [108]. The Hanning window's function is:

$$w(n) = 0.5\left(1 - \cos\left(\frac{2\pi n}{M}\right)\right), -M \leq n \leq M, \tag{3.8}$$

where $2M+1$ is the window length. This weight function assigns the greatest weight to the signal points in the center of the window, and increasingly smaller weights to values that are further away from the center. The input signal points (x) multiplied with the corresponding weights produce the smoothed output signal (y) as:

$$y_i = \frac{1}{M} \sum_{j=-M}^{M} x_{i+j}\, w_j. \tag{3.9}$$

The MAHF effectively reduces high-frequency signal noise while preserving low-frequency signal characteristics.

Locally Weighted Linear Regression

Linear regression, or least square moving average, is extensively used in various statistical and DSP applications, particularly for spectral data smoothing [109]. The underlying principle behind this technique is the fitting of the input data points to a linear regression function, given by:

$$y = a + bx, \tag{3.10}$$

where x is the independent variable, y is the variable dependent on x, a is the y-intercept of the regression line, and b is the regression line slope. By computing parameters a and b, we can estimate y for any given x. The slope is a ratio determined as:

$$b = r\frac{S_y}{S_x}, \tag{3.11}$$

where r is the Pearson correlation coefficient, S_y is standard deviation of y, and S_x is standard deviation of x. Given that $\bar{x}$ is mean of N data points of x, and $\bar{y}$ is mean of N data points of y, equations (3.12)-(3.14) compute the values of r, S_y, and S_x, and equation (3.15) calculates the y-intercept of the regression line.

$$r = \frac{\sum_{i=1}^{N}((x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}, \tag{3.12}$$

$$S_y = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \bar{y})^2}{N-1}}, \tag{3.13}$$

$$S_x = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}}, \tag{3.14}$$

$$a = \bar{y} - b\bar{x}. \tag{3.15}$$

The LRF [110] estimates y for each value of x using the neighboring data points within the defined window of *2M+1*. The weights assigned to the neighboring points depend on the distance from neighbors to x, particularly, the highest weight is obtained for the x point itself, and the minimum weight is obtained for the most distant from x point. Equation (3.16) determined the weight function.

$$w_i = \left(1 - \left|\frac{d(x_i, x)}{max_i d(x_i, x)}\right|^3\right)^3, \tag{3.16}$$

where x is an input data point, x_i are the neighboring points for x within a specified window, $d(x_i, x)$ is the distance along the abscissa from the neighbor x_i to x, and $max_i d(x_i, x)$ is the distance from x to its most distant neighbor within the window. For each input point x, the parameters computed using (3.17)-(3.19) define a local regression model.

$$\bar{x} = \frac{\sum_{i=-M}^{M} w_i x_i}{\sum_{i=-M}^{M} w_i}, \tag{3.17}$$

$$\bar{y} = \frac{\sum_{i=-M}^{M} w_i y_i}{\sum_{i=-M}^{M} w_i}, \tag{3.18}$$

$$b = \frac{\sum_{i=-M}^{M} w_i x_i y_i - \bar{x}\bar{y}\sum_{i=-M}^{M} w_i}{\sum_{i=-M}^{M} w_i\, x_i^2 - \bar{x}^2 \sum_{i=-M}^{M} w_i}. \tag{3.19}$$

As the LRF involves more operations than other investigated smoothing techniques, it is a more computationally expensive method.

Savitzky-Golay Filter

The SGF [111] is another technique well adapted for data smoothing. In the MARF, the underlying function uses approximation by a constant *(1/(2M+1))*, provided by (3.3). Unlike the MARF, the SGF smoothing approximates the input data by a local polynomial least-squares fitting within a sliding window around each data point. The coefficients of a fitted polynomial are linear. Thus, taking linear combinations performs the data fitting. For the selected polynomial degree D and window size $2M+1$, there exists a specific set of filter coefficients c_n computed in (3.22) using (3.20)-(3.21):

$$\begin{gathered} A = \{a_{i,j}\}, \\ a_{i,j} = \big(i - (M+1)\big)^{j-1}, \\ i = 1, \ldots, 2M+1, j = 1, \ldots, D+1, \end{gathered} \tag{3.20}$$

$$B = (A^T A)^{-1}, \tag{3.21}$$

$$c_n = \sum_{m=1}^{D+1} b_{1,m} n^{m-1}\, , -M \le n \le M, \tag{3.22}$$

where A is the design matrix with elements $a_{i,j}$, B is the inverse matrix of the product of A and the transpose of A, and $b_{1,m}$ is the element from the first raw and the m^{th} column of the matrix B [112].

After calculating the set of coefficients, we apply them to input data points within the window. For each next point, the window shifts to perform a new least-squares fit. Given the input data x, consisting of N points, we obtain the smoothed output data y by:

$$y_i = \sum_{n=-M}^{M} c_n \, x_{i+n}, M \leq i \leq N - M. \tag{3.23}$$

One of the main benefits of the SGF smoothing technique is that it maintains the area, position, and width of peaks. In comparison with the MARF with the same window length, the SGF has an advantage in its ability to preserve the shape of the original signal more accurately.

In this research, we apply the four above-mentioned smoothing approaches to the FFT spectrum data with the purpose of constructing high-performance ML models.

3.4 Machine Learning

ML is a leading technique implemented to solve complex problems and develop novel and innovative solutions. The evidence why ML is such an important tool for researchers and scientists lies in its ability to identify patterns in data and make predictions based on those identified patterns.

There are four basic types of ML: Supervised Learning (SL), Semi-supervised Learning (SSL), Unsupervised Learning (UL), and Reinforcement Learning (RL) [113]. In SL, a number N of samples x with assigned labels y represents the dataset D_{SL}:

$$D_{SL} = \{(x_i, y_i)\}_{i=1}^{N}. \tag{3.24}$$

Each sample x_i is a vector of features, or characteristic values of the sample. The labels y_i in classification problems determine the accessory of a sample to one of the defined classes. Given an input dataset, SL generates a model that receives a feature vector as input and outputs the label corresponding to this sample. In contrast to SL, UL deals with the datasets D_{UL} of samples that are not labeled:

$$D_{UL} = \{x_i\}_{i=1}^{N}, \tag{3.25}$$

and the goal of UL is to build a model that transforms an input feature vector into another vector or a quantity to solve a specific problem. SSL is similar to SL in terms of its goal; however, the difference is in the examined datasets, which in case of SSL consist of both labeled and unlabeled samples. RL does not require labeling the datasets. It relies on the concept of maximizing the expected reward from an action generated for a particular input feature vector [113]. This research focuses on the SL algorithms for data classification, due to the fact that they work well with labeled datasets [114].

Support Vector Machine

SVM is an SL algorithm that performs regression and classification analysis of data [115]. The input data consists of the labeled vectors of features. Given a specific training dataset labeled with the actual class information, the algorithm generates the classification function, which outputs class identifiers for new unlabeled data. The SVM uses the principle of separation the vectors of samples (*x*) with different class labels (*y*) by a hyperplane that determines decision boundaries and has the function given by:

$$h(w, b) = wx \pm b, \tag{3.26}$$

where *w* is a weight vector and *b* is a bias term [116]. To maximize the distance (or separation) of different groups, the SVM solves the following optimization problem:

$$\min \frac{1}{l} \sum_{i=1}^{l} \max\{0, 1 - y_i(wx_i - b)\} + C\|w\|^2, \tag{3.27}$$

where l is the number of the feature vectors x_i in the training set, y_i are their binary labels (-1 or 1), and *C* is a penalty parameter of misclassification, controlling the trade-off between the error frequency and the decision rule complexity.

The described approach creates a linear SVM classifier. In case it does not efficiently separate the classes, a non-linear classifier is more suitable. It relies on kernel functions to map the data into a higher dimensional feature space and to establish the hyperplane's nonlinearity level. The commonly utilized non-linear kernel types are radial basis function (RBF), polynomial, and sigmoid kernels [117]. The kernel type and parameters are the critical factors at the SVM training stage, as they strongly influence the output model accuracy. For this reason, it is particularly important to design an appropriate kernel function through the optimization of its type and adjustment of its parameters. In this work, we created the SVM model with a linear kernel function and set the penalty parameter *C* to 25.

Depending on the number of classes, the SVM solves two types of problems: binary and multiclass classifications. In binary problems, only two classes are involved, and multiclass problems deal with the number of classes higher than two. As the number of classes referred to capacitor configurations in this research is five (N^6A^0, N^5A^1, N^3A^3, N^1A^5, N^0A^6), we consider a multiclass classification problem.

The complexity of a multiclass SVM is associated with the choice of the one-versus-all (OVA) or the one-versus-one (OVO) strategy [118]. The primary difference between the two methods is in the number of constructed classifiers, as well as in the way of selecting the output classes. The OVA SVM method introduced in [119] generates k SVM models, where k is the number of classes. It selects the class when the SVM accepts this class and rejects the rest of the classes. OVO method was introduced in [120], and its first application to the SVM was demonstrated in [121]. In the OVO SVM, the number of classifiers Nc is given by:

$$N_c = k(k-1)/2, \tag{3.28}$$

and each classifier trains on data from two classes by solving a binary classification problem [122]. The trained classifiers perform testing using the voting approach [121], which predicts the class of the sample based on the maximum vote number for a particular class. We selected the OVO strategy because the training data is uniform for each binary classifier.

Artificial Neural Network

ANN is another SL algorithm used for data classification [123]. We used a multilayer perceptron (MLP) neural network to classify the data. The MLP is a specific feedforward structure of the ANN that organizes the neurons into layers, from the input to the output layer, with a certain number of hidden layers between them, and the signal processing direction is from the input to the output [124].

Fig. 3.8 shows the ANN architecture diagram for this research. The MLP topology has one input layer, one hidden layer, and one output layer. The input layer consists of a set of n neurons representing input feature vectors x fed into the algorithm. This layer is an integral part of the ANN as it is the primary depository of the available amount of data that is subsequently transferred to the ANN. The selected hidden layer's size is 40 neurons because, in our application, it provides an optimal trade-off between computation time and estimation accuracy. The primary task of this layer is to weight the inputs, sum them, and pass through an activation function. In the training stage, the weights update by comparing the produced outputs with the target labels. These weights characterize the ANN learning. The output layer consists of 5 neurons corresponding to the 5 capacitor classes. It receives the values from the hidden layer and matches the targets with the desired output using an output activation function.

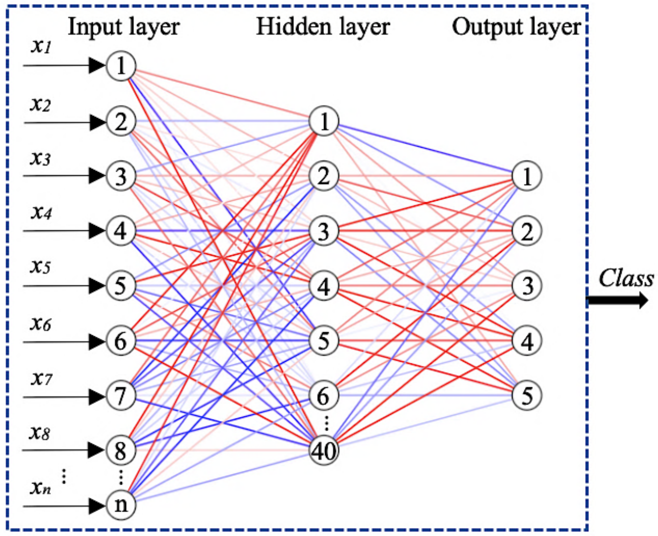


Fig. 3.8: ANN architecture diagram.

A three-layer MLP ANN consisting of n_1, n_2, and n_3 input, hidden, and output nodes, respectively, has the following representation:

$$y_r = f_2\left(\sum_{q=0}^{n_2} w_{qr}^2 f_1\left(\sum_{p=0}^{n_1} w_{pq}^1 x_p\right)\right), r \in [1, n_3], \tag{3.29}$$

where y_r are the ANN outputs, $f_2(g)$ is the output layer's activation function, $f_1(g)$ is the hidden layer's activation function, p is the index of an input node, q is the index of a hidden node, r is the index of an output node, x_p is an input node, w_{pq}^1 are the weights connecting the input and hidden layers, and w_{qr}^2 are the weights connecting the hidden and output layers [125].

3.5 Hidden Markov Model

HMM is a stochastic model that characterizes the statistical proprieties of a signal and is related to Markov sources or Markov chains' probabilistic functions [126]. A Markov chain is a random process of discrete variables associated with system states. It describes the possible transitions between the states and the relations between states and observations in a probabilistic form. The defining characteristic of a Markov chain is that the state transition depends only on the current state and is not influenced by the past states. System modeling with the HMM method involves the generation of two types of sequences. The "hidden" state sequence is a not directly visible Markov chain of true system states. The observation sequence represents the observable data, by which the true state characteristics can be inferred. Fig. 3.9 shows a generalized architecture of an HMM diagram with three hidden states (s_1, s_2, s_3) and three possible observations (o_1, o_2, o_3).

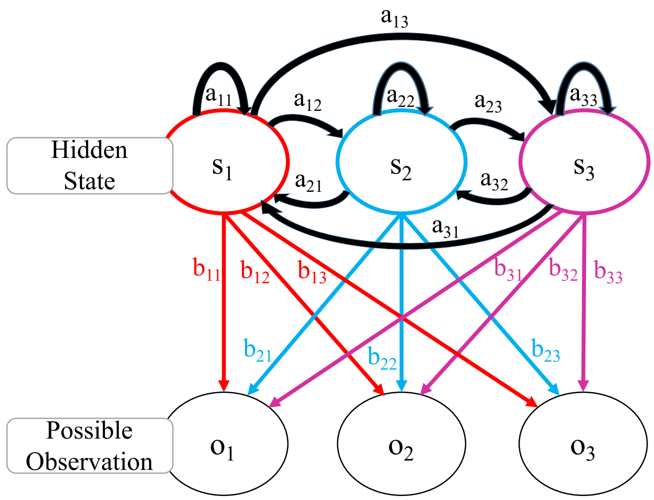


Fig. 3.9: Generalized HMM architecture.

An HMM model with a discrete output probability distribution has the following compact form:

$$\lambda = (A, B, \pi), \tag{3.30}$$

where λ is a model, A is the state transition probability distribution, B is the observation probability distribution, and π is the initial state probability distribution. For the model with N number of states, the set of states is given by:

$$S = \{s_1, s_2, \dots, s_N\}. \tag{3.31}$$

HMM is a finite-state machine which changes state every time unit. Thus, the state sequence Q comprises the q_t states of the model at a given time t:

$$Q = \{q_t\}, q_t \in S, 1 \le t \le T, \tag{3.32}$$

where T is the number of observations. The state transition probability matrix has the following form:

$$A = \{a_{i,j}\}, 1 \le i \le N, 1 \le j \le N, \tag{3.33}$$

$$a_{i,j} = P(q_{t+1} = s_j | q_t = s_i), a_{i,j} \ge 0, \sum_{j=1}^{N} a_{i,j} = 1. \tag{3.34}$$

The observation sequence consists of a set of visible states generated by the model λ:

$$O = \{o_k\}, 1 \le k \le T, \tag{3.35}$$

and the observation probability distribution is:

$$B = \{b_{i,j}(k)\}, 1 \le i \le N, 1 \le j \le N, 1 \le k \le T, \tag{3.36}$$

$$b_{i,j}(k) = P(o_k | q_t = s_t), b_{i,j} \ge 0, \sum_{j=1}^{N} b_{i,j} = 1. \tag{3.37}$$

The initial state distribution is the probabilities of the initial system states defined as:

$$\pi_i = P(q_1 = s_i), 1 \le i \le N. \tag{3.38}$$

A common purpose of using the HMM is to retrieve the sequence of data that cannot be observed when the information that depends on this sequence is available. In classification problems, the probability distribution produced by the model provides insight into the information about the system's states. The model that achieves the highest probability determines the data class. There are three basic algorithms for solving HMM evaluation problems. The forward-backward algorithm operates with defined model parameters to calculate the probability of a specific observation sentence. The Viterbi algorithm similarly operates with established model parameters, but searches for a hidden state sequence with the highest probability of producing a given output sequence. The Baum-Welch algorithm searches the state transition probabilities and observation probabilities for a particular observation sequence [127].

3.6 Summary

This chapter provides the analyses of the introduced E-PHM system. It explains the capacitor's aging mechanisms and establishes the relationship of the DC-link capacitance's age on the externally monitored EMI behavior. The analytical and simulation analysis determines that the magnitudes of the EMI peaks decrease as the capacitor age increases. The discussed DSP techniques allow to obtain distinctive spectral trends for different capacitor classes. This is crucial to perform accurate capacitor classification using ML algorithms. SL methods are most suitable for this purpose, since the labels for the investigated data are available. The selected SVM model is the OVO multiclass SVM with a linear kernel because the number of classes is five, the training data has equal number of feature vectors for each class, and the classes are linearly separatable. The selected ANN model is the MLP with three layers because this is a common ANN structure that is able to approximate mapping of the inputs to the outputs through a hidden layer with a reliable accuracy. The HMM is a probabilistic model that connects true system states with the observations obtained from measurements. Its goal in this research is to retrieve a sequence of the actual DC-link age classes from the initially estimated sequence of classes provided by ML classification.

Chapter 4 Experimental Results

This chapter describes the CM procedure within the proposed HMM-supported ML approach. Then, it provides the experimental results to evaluate the accuracy of the DC-link capacitor age class estimation with respect to different types of input signals and two types of ML models (the SVM and the ANN). Also, it determines the ML classification accuracy improvement achieved by using the HMM. The results show that the HMM-supported ML CM is an efficient approach for DC-link age class estimation using only the data from CT measurements. In addition, they demonstrate that both the SVM and the ANN can classify the DC-link capacitor's age condition from the smoothed current spectrum, and the HMM significantly reduces ML misclassification errors.

4.1 Condition Monitoring Procedure

Our CM approach to estimate a DC-link capacitor's health consists of four stages: data acquisition, data preprocessing and feature selection, ML classification, and HMM-based output correction, as shown in Fig. 4.1.

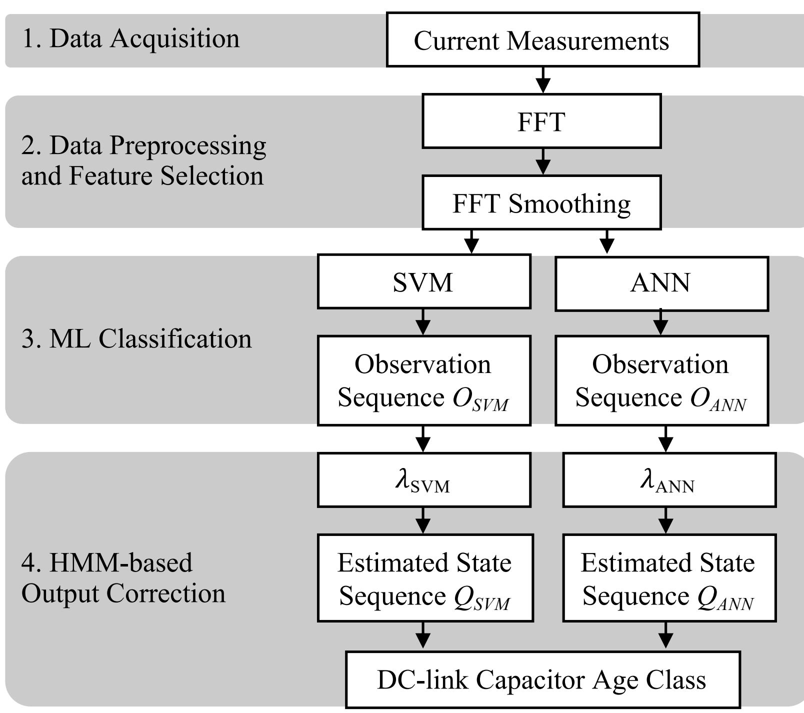


Fig. 4.1: HMM-supported ML CM framework architecture.

First, we obtain a dataset of current signals by a set of measurements with a CT. In the preprocessing stage, we directly apply the FFT upon acquired time-domain data to obtain signal features in the frequency domain. We can remove the negative half of the spectral data without loss of relevant information. Thus, we select the features that compose only the positive half of the spectrum for further processing. Such feature selection simplifies ML models, reduces their training time, and decreases overfitting [128]. Then, we obtain the spectral trend with one of the smoothing filters. We examine four different filters: the MARF, the MAHF, the LRF, and the SGF.

In the third stage, we employ the SVM and the ANN learning algorithms to cluster the feature vectors into five classes, corresponding to a specific capacitor age. Each algorithm performs the classification process through three phases, namely, training, validation, and testing phase. In the training phase, the ML model adjusts its parameters to fit an input training dataset. To prevent the ML model overfitting, we use validation of the trained model with a validation dataset, which stops the training when the performance of the validation is starting to deteriorate. In the testing phase, the validated model predicts class labels for the unseen test dataset. We measure the classification performance of the ML model on the test data in terms of the accuracy defined as a ratio of the number of correctly classified samples to the total number of classification outputs given by:

$$Accuracy = \frac{TC}{TC + FC}, \tag{4.1}$$

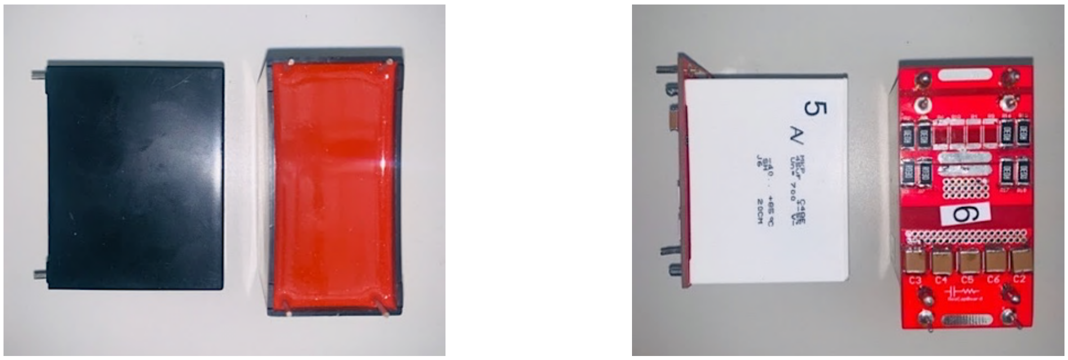
where TC is the number of true classes and FC is the number of false classes predicted by the model. The ML classification results indicate the true state of the DC-link capacitor with an original accuracy, which is improved in the next stage.

In the fourth stage, we construct two models, λ_{SVM} and λ_{ANN}, based on the information learned from the ML classification. After defining the HMMs parameters, we develop a correction method and apply it to the ML outputs to estimate the DC-link capacitor class with an enhanced accuracy.

To implement the CM procedure, we created a MATLAB code presented in the Appendix and uploaded to the GitHub repository (as of 2020) [129].

4.2 Capacitor Classes

For this experiment, we constructed the DC-link using six capacitors connected in parallel. Each of them we designed in two versions, "New" and "Aged". We implemented the "New" version (Fig. 4.2a) using MPPF capacitors (p/n: MKP1848C65090JY5 [97]) with the following characteristics: C_N = 50 µF, and $R_{ESR,N}$ = 6 mΩ. We created the "Aged" version from MPPF capacitors (p/n: C4AEJBW5450A3MJ [130]) using interface boards (Fig. 4.2b) to emulate aging represented by the values of C_A = 45 µF and $R_{ESR,A}$ = 19.4 mΩ. These values reflect findings reported in the literature that a 2 to 5% drop in capacitance and 2 to 3x increase in ESR is regarded as a failure for film capacitors [67].



(a) (b)

Fig. 4.2: Capacitors used in the DC-link of type (a) "New" and (b) "Aged".

Our design strategy tests a specific configuration of the five capacitor classes. Its notation is: N^xA^y, where $x \in [0, 1, 3, 5, 6]$ is the number of "New" capacitors, and $y \in [0, 1, 3, 5, 6]$ is the number of "Aged" capacitors in a DC-link bus. Each class represents the age condition with the parameters defined in Table 4.1.

Table 4.1: DC-link Capacitor Class Parameters

Class	**Configuration**	**Age Condition**	R_{ESR}	C_{DC}
1	N^6A^0	New	1.000 mΩ	300 µF
2	N^5A^1	Partially aged	1.130 mΩ	295 µF
3	N^3A^3	Half aged	1.527 mΩ	285 µF
4	N^1A^5	Nearly aged	2.356 mΩ	275 µF
5	N^0A^6	Aged	3.233 mΩ	270 µF

Fig. 4.3 illustrates the diagram of the E-PHM system that uses regular CM readings of the estimated capacitor class, or age, to create a predictive maintenance plan. It takes periodic measurements, generates an alert when the DC-link capacitor lifetime end is approaching, and produces an alarm when it requires the capacitor replacement to avoid its anticipated failure. The lead time indicates the available time interval to implement a replacement.

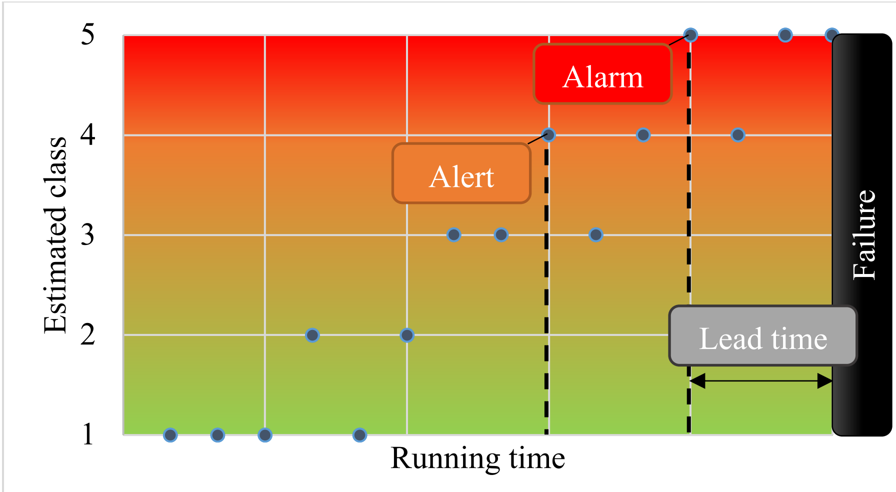


Fig. 4.3: E-PHM diagram.

In this thesis, we consider five capacitor classes for the E-PHM system to implement five types of maintenance actions. Table 4.2 specifies these actions provided by the indicated capacitor health. In such a system, class 1 ensures the successful replacement of a capacitor and controls the quality of the new component. If, for example, the installed device's parameters slightly deviate from its declared values, the system indicates this by the estimated class 2. In addition, based on the RUL of the capacitor inferred from the initial capacitor health (classes 1 and 2), the system plans the operation cycle of the inverter module. When the capacitor reaches the class 3 stage, the E-PHM system enables more frequent measurements in order to timely identify class 4, reflecting the approaching end of the RUL. At a time when it detects class 4, it schedules a maintenance plan, and the first detection of class 5 signals the maintenance requirement, and the component should be replaced before it fails. For the additional decision-making purposes, more capacitor classes can be considered. However, this may increase the computational time of the involved algorithms.

Table 4.2: E-PHM Maintenance Actions

Estimated Class	**Maintenance Action**
1	Control a replacement
2	Plan an operation cycle
3	Increase the frequency of measurements
4	Schedule a replacement
5	Implement a replacement

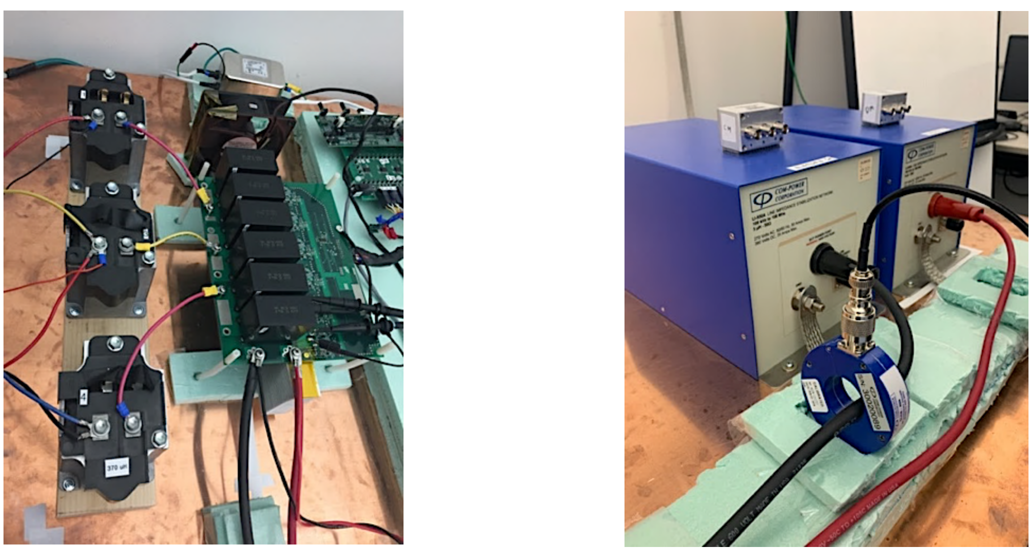
4.3 Experimental Setup and Data Acquisition

We carried out the CM experiment on a three-phase inverter operated at 20 kHz switching frequency, 60 Hz fundamental frequency, 540 VDC, 7.1 A of input current, and a modulation index of 0.9. The R-L load consisted of 370 μH inductors and 24 Ω resistors (see Table 4.3).

Table 4.3: Operational Parameters of the Three-phase Inverter

Parameter	Value	Unit
Switching frequency	20	kHz
Fundamental frequency	60	Hz
Input voltage	540	V
Input current	7.1	A
Modulation index	0.9	-
Resistive load	24	Ω
Inductive load	370	μH

The testbench for measuring the conducted EMI appears in Fig. 4.4. In the three-phase inverter (see Fig. 4.4a), the DC-link consisted of six switchable capacitors. We acquired the EMI current data using a CT with a 3dB bandwidth of a 10 kHz – 400 MHz [131] (Fig. 4.4b). An oscilloscope measured its signal.



(a) (b)

Fig. 4.4: Experimental setup with (a) three-phase inverter and (b) CT at a LISN's output.

For each of the five capacitor classes, we acquired two signals at a sample rate of 0.1 GSample/sec. The duration of the signals was 100 ms. The test conditions were the same for each of the measured signals for the purpose of using in the further ML models training. Also, we acquired 1000 signals of 1 ms length and the same sampling rate of 0.1 GSample/sec for the ML models testing. Table 4.4 summarizes the training data (D_{train}) and testing data (D_{test}) acquisition parameters.

Table 4.4: Data Acquisition Parameters

Parameter	D_{train}	D_{test}
Number of signals per class	2	200
Total number of signals	10	1000
Signal duration	100 ms	1 ms
Sample rate	0.1 GSample/sec	0.1 GSample/sec
Number of samples per signal	10^7	10^5

4.4 Data Preprocessing and Feature Selection

We passed the experimental data through several stages of preprocessing in MATLAB software to obtain six types of data with different behavioral patterns, namely, time-domain current, current FFT spectrum, and four smoothed types of FFT. We arranged the outputs in the complete matrices of features along with the corresponding target matrices for the ML phase, described below. Table 4.5 provides a summary of the preprocessing stages.

Table 4.5: Preprocessing Stages

Stage	Input Data	Processing Type	Output Feature Matrix	Feature Matrix Size	Target Matrix Size	Output Data Type	Data Type Description
1	$\mathbf{D}_{train}$	Splitting	$\mathbf{X}_{train}$	10^3x10^5	10^3x1	**X**	Time-domain signal
	$\mathbf{D}_{test}$	-	$\mathbf{X}_{test}$	10^3x10^5	10^3x1		
2	$\mathbf{X}_{train}$	Single-sided FFT	$\mathbf{F}_{train}$	10^3x$5\cdot10^4$	10^3x1	**F**	FFT spectrum
	$\mathbf{X}_{test}$		$\mathbf{F}_{test}$	10^3x$5\cdot10^4$	10^3x1		
3	$\mathbf{F}_{train}$	MARF smoothing	$\mathbf{MAR}_{train}$	10^3x$5\cdot10^4$	10^3x1	**MAR**	MARF-smoothed spectrum
	$\mathbf{F}_{test}$		$\mathbf{MAR}_{test}$	10^3x$5\cdot10^4$	10^3x1		
4	$\mathbf{F}_{train}$	MAHF smoothing	$\mathbf{MAH}_{train}$	10^3x$5\cdot10^4$	10^3x1	**MAH**	MAHF-smoothed spectrum
	$\mathbf{F}_{test}$		$\mathbf{MAH}_{test}$	10^3x$5\cdot10^4$	10^3x1		
5	$\mathbf{F}_{train}$	LRF smoothing	$\mathbf{LR}_{train}$	10^3x$5\cdot10^4$	10^3x1	**LR**	LRF-smoothed spectrum
	$\mathbf{F}_{test}$		$\mathbf{LR}_{test}$	10^3x$5\cdot10^4$	10^3x1		
6	$\mathbf{F}_{train}$	SGF smoothing	$\mathbf{SG}_{train}$	10^3x$5\cdot10^4$	10^3x1	**SG**	SGF-smoothed spectrum
	$\mathbf{F}_{test}$		$\mathbf{SG}_{test}$	10^3x$5\cdot10^4$	10^3x1		

1) Data type **X** (time-domain signal): we equally split each of the 10 acquired current signals into 100 signals of 1 ms duration to form a dataset $\mathbf{X_{train}}$ of 1000 time-domain signals for further training. We included another 1000 acquired signals in the test dataset $\mathbf{X_{test}}$. The number of features for data type **X** was equal to the signal length, which was 10^5 points. Fig. 4.5a shows the variations of a current signal with different capacitor classes.

2) Data type **F** (FFT spectrum): for each signal of type **X**, we obtained a single-sided FFT to create datasets $\mathbf{F_{train}}$ and $\mathbf{F_{test}}$, each containing 1000 signals of data type **F**. The number of points in an **F** signal was 50000, and the frequency resolution was 1 kHz. The FFT spectrum for different capacitor classes can be observed in Fig. 4.5b.

3) Data type **MAR** (spectrum smoothed with the MARF): we smoothed each spectrum from datasets $\mathbf{F_{train}}$ and $\mathbf{F_{test}}$ with the MARF using a window of 1000 point size. The resulting datasets $\mathbf{MAR_{train}}$ and $\mathbf{MAR_{test}}$ consisted of 2000 signals in total, each with 50000 valid data points. MARF-smoothed spectrums for each capacitor class appears in Fig. 4.5c.

4) Data type **MAH** (spectrum smoothed with the MAHF): we applied a 1000-point MAHF to each signal of data type **F** to produce datasets $\mathbf{MAH_{train}}$ and $\mathbf{MAH_{test}}$ of 2000 total signals with 50000 points length each (see Fig. 4.5d).

5) Data type **LR** (spectrum smoothed with the LRF): we performed the LRF smoothing with a window size of 1000 points on data type **F** to get datasets $\mathbf{LR_{train}}$ and $\mathbf{LR_{test}}$ of 2000 total signals, 50000 points length each. Fig. 4.5e demonstrates LRF-smoothed spectrums for different capacitor classes.

6) Data type **SG** (spectrum smoothed with the SGF): we smoothed each spectrum of type **F** with the SGF using a 1000-point window to get datasets $\mathbf{SG_{train}}$ and $\mathbf{SG_{test}}$ of 2000 smoothed signal spectrums, each containing 50000 points (see Fig. 4.5f).

We labeled all feature vectors with the values (1, 2, 3, 4, or 5) corresponding to the actual class accessory. These labels composed the target vectors for the ML classification.

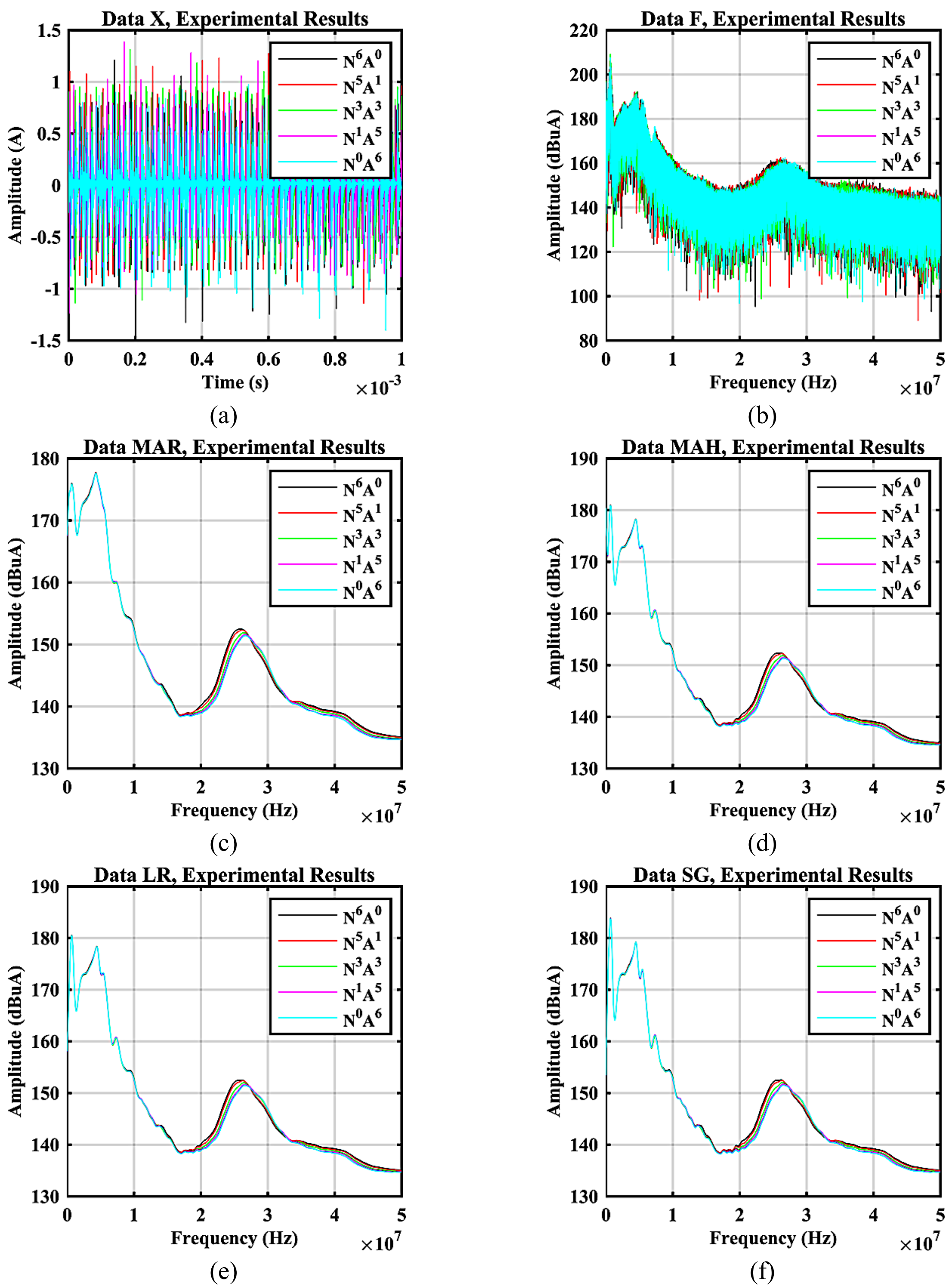


Fig. 4.5: Experimental results on different data types: (a) **X**, (b) **F**, (c) **MAR**, (d) **MAH**, (e) **LR**, (f) **SG**.

The experimental results verify that as the capacitor age increases, the magnitudes of the oscillation peaks decrease and the resonant frequency increases. The zoomed view of the smoothed spectrum in Fig. 4.6 clearly illustrates these relationships.

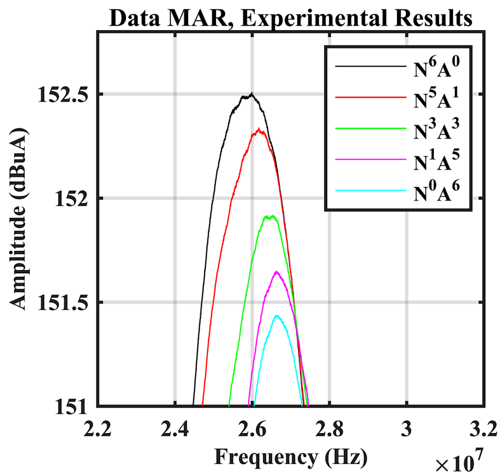


Fig. 4.6: Experimental magnitude of current peaks for different capacitor classes.

4.5 Machine Learning Classification

For all the input datasets (**X**, **F**, **MAR**, **MAH, LR**, and **SG**), we used the same conditions to train and evaluate ML models. For training, we used the datasets $\mathbf{X}_{\mathbf{train}}$, $\mathbf{F}_{\mathbf{train}}$, $\mathbf{MAR}_{\mathbf{train}}$, $\mathbf{MAH}_{\mathbf{train}}$, $\mathbf{LR}_{\mathbf{train}}$, and $\mathbf{SG}_{\mathbf{train}}$. Using one of the most common training/validation ratios of 70%/30% [132], we allocated 30% of the training data for validation. To test the resulting validated models, we used the datasets $\mathbf{X}_{\mathbf{test}}$, $\mathbf{F}_{\mathbf{test}}$, $\mathbf{MAR}_{\mathbf{test}}$, $\mathbf{MAH}_{\mathbf{test}}$, $\mathbf{LR}_{\mathbf{test}}$, and $\mathbf{SG}_{\mathbf{test}}$.

We evaluated the performance of the SVM and the ANN models in terms of the average and maximum prediction accuracy of the classifiers on the test data. The first characteristic defines the estimated predictive ability of the models, providing the average classification success of 100 SVM/ANN runs. The second one demonstrates the highest achieved accuracy within these runs. Table 4.6 summarizes the SVM and ANN performance for the investigated datasets, and Fig. 4.7-4.8 show the corresponding confusion matrices.

Table 4.6: ML Classification Performance

Data Type	SVM		ANN	
	Average Test Accuracy, %	Maximum Test Accuracy, %	Average Test Accuracy, %	Maximum Test Accuracy, %
X	19.22	20.00	18.06	19.50
F	73.92	75.30	64.14	70.00
MAR	88.19	90.80	88.02	90.20
MAH	89.01	91.80	88.28	90.00
LR	88.03	90.20	88.38	90.70
SG	88.46	90.40	87.47	89.70

The classification results indicate poor SVM and ANN performance on the data type **X**: maximum test accuracy is only 20.0% and 19.5%, respectively. The confusion matrices (Fig. 4.7a and Fig. 4.8a) reveal that the classifiers confuse each predicted class with the rest of the classes. These unsatisfactory results indicate that the non-stationary time-domain current signal is not suitable for the ML classification due to its varying over time frequency components.

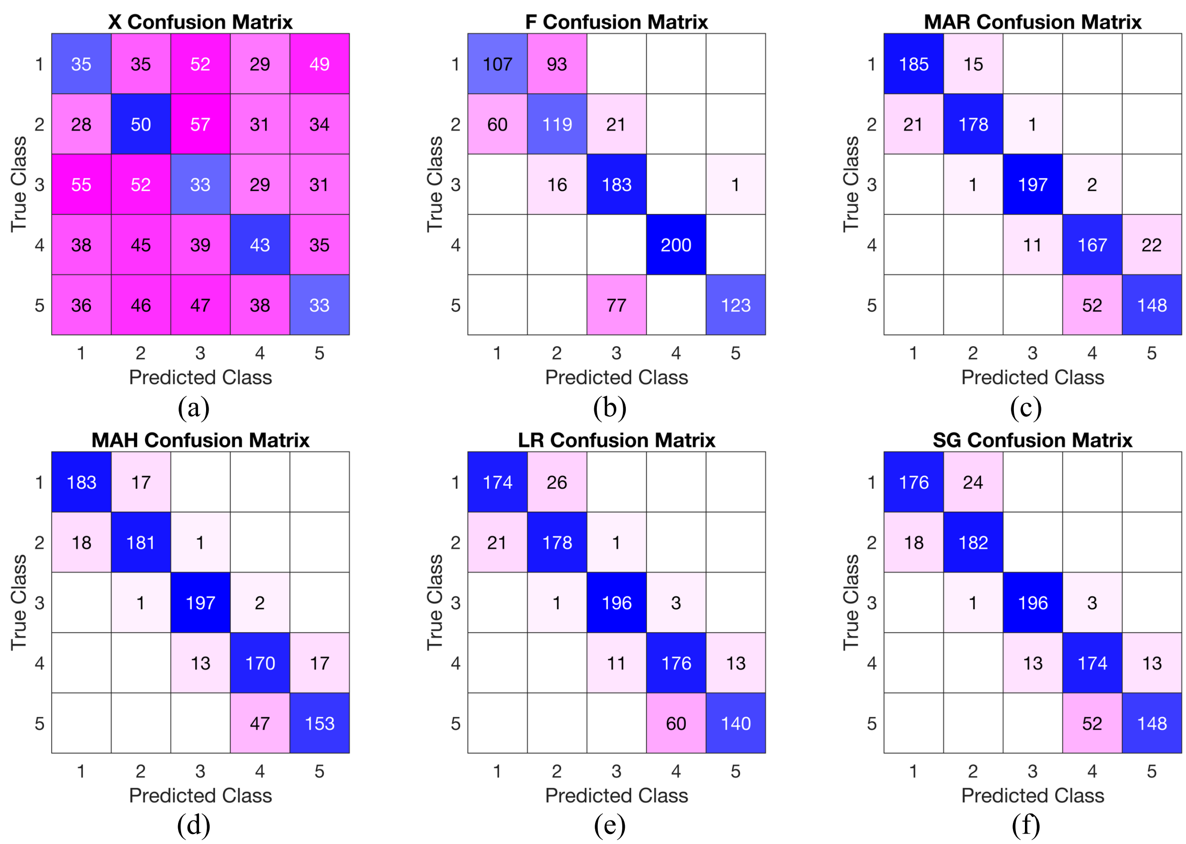


Fig. 4.7: Test confusion matrices, SVM results using different data types:
(a) **X**, (b) **F**, (c) **MAR**, (d) **MAH**, (e) **LR**, (f) **SG**.

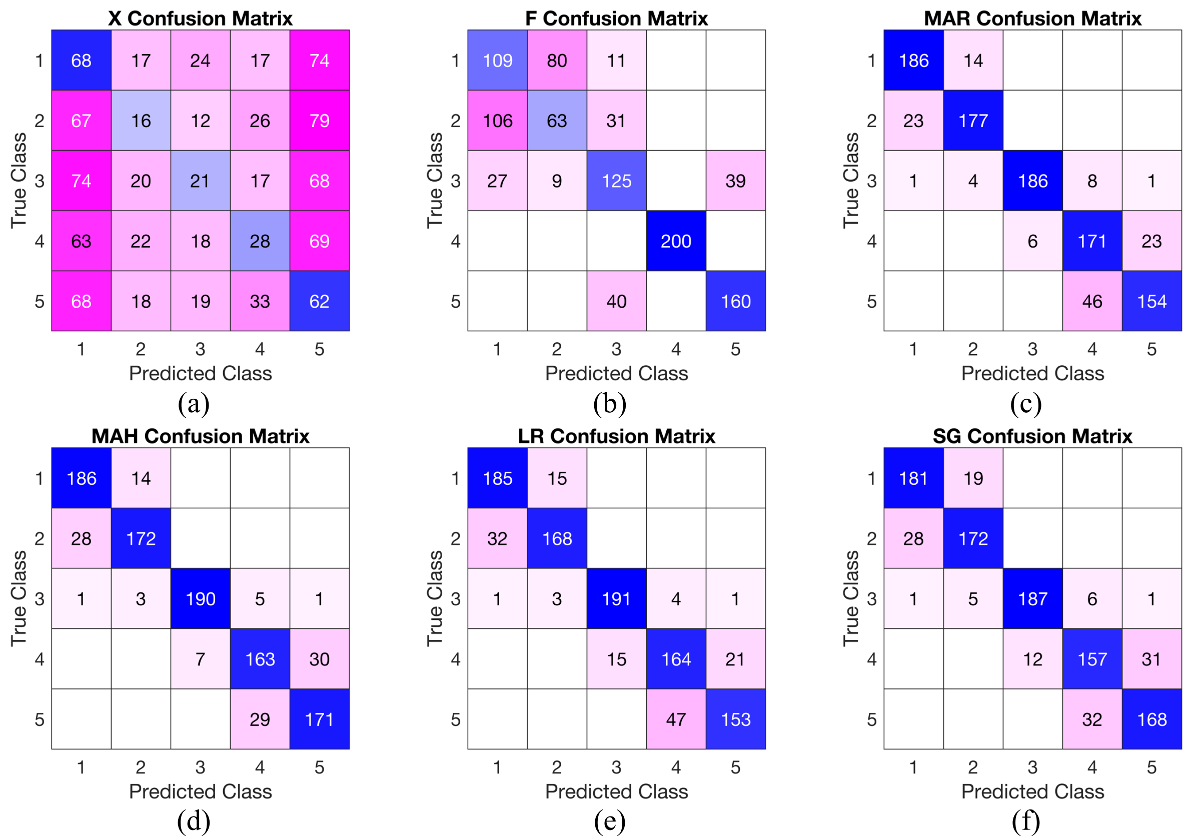


Fig. 4.8: Test confusion matrices, ANN results using different data types:
(a) **X**, (b) **F**, (c) **MAR**, (d) **MAH**, (e) **LR**, (f) **SG**.

Analyzing the results for data type **F**, it is reasonable to conclude that the FFT spectrum produces comparatively acceptable classification accuracy, as initially hypothesized. The highest achieved classification accuracy is 75.3% for the SVM and 70.0% for the ANN. However, the main issue with this data type is that besides confusing adjacent classes, the SVM and the ANN do not always correctly distinguish medium-aged capacitors (class 3) and aged capacitors (class 5), as can be seen in Fig. 4.7b and Fig. 4.8b. In a problematic scenario, the predictor may determine the capacitor condition as satisfactory, while actually it is close to failure, which is contradictory to the objectives of the stated methodology. Thus, data type **F** cannot be used to generate reliable capacitor health estimation.

For the smoothed spectral data **MAR**, **MAH**, **LR**, and **SG**, the features form a more distinguishable trend for different classes, making it feasible to create a computationally effective classifier, as shown in Table 4.6. The performance evaluation for these four types of data demonstrates the consistency in the model predictive ability with respect to each class, indicated by the numbers of correctly predicted cases in the corresponding confusion matrices (Fig. 4.7c-f and Fig. 4.8c-f). All of the four abovementioned types of filtered data result in a high and relatively similar average test accuracy, with deviations of less than 1%. Among them, the spectrum smoothed by the 1000-point MAHF achieves the highest classification accuracy, specifically 91.8%, using the SVM classifier. Comparing all the smoothing techniques, the MARF algorithm involves lower computational time, associated with the number of additions and multiplications per sample, as demonstrated in Table 4.7. Thus, the MARF is the preferable smoothing technique for our approach in an embedded system application.

Table 4.7: Number of Operations per Sample for Smoothing Filters with 1000-point Window

Smoothing filter	**Number of multiplications**	**Number of additions**
MARF	1	999
MAHF	1001	999
LRF	11008	5998
SGF	1001	999

Comparing the SVM and ANN classifiers, in most cases the SVM performs better than the ANN. This finding is consistent with other research works reported in the literature [133]-[135]. The ANN uses only empirical risk minimization in the training stage, while the SVM considers both empirical and structural risk minimization that results in an improved generalization capability of the training [136]. Moreover, the test confusion matrices for the smoothed spectral data demonstrate that the SVM confuses only adjacent classes of capacitors, while the ANN also confuses class 3 with the nonadjacent classes 1 and 5. Consequently, the SVM is the preferred ML algorithm for the DC-link capacitor CM within our approach.

4.6 HMM-based Output Correction

In this stage, we introduce the HMM with the aim of increasing the accuracy of the ML classification on the smoothed spectral data. Fig. 4.9 shows the diagram of the HMM. The capacitor configurations, related to the true age class of a DC-link capacitor, represent the hidden states of the HMM. For each sequence of measurements, the ML classification algorithm produces an output sequence of classes treated as an observation sequence by the HMM.

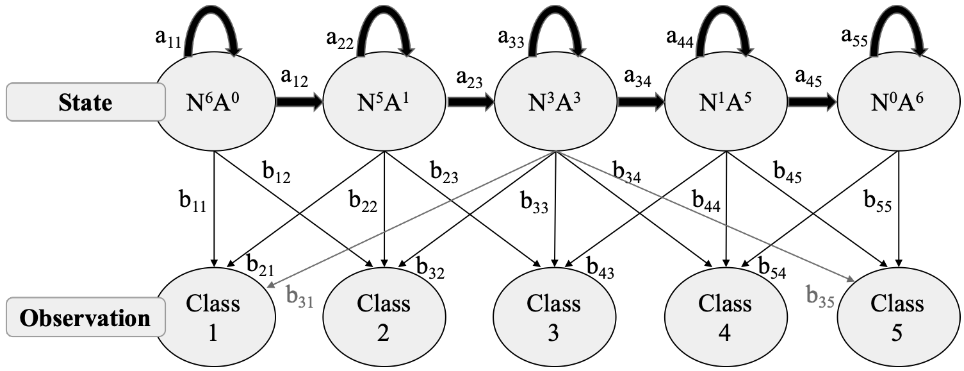


Fig. 4.9: HMM diagram.

We built two HMM models based on the outputs produced by the ANN and the SVM for smoothed spectrum data types. The first model based on the SVM observations has a form given in (4.2) with parameters defined in (4.3)-(4.5).

$$\lambda_{SVM} = (A_1, B_1, \pi_1), \tag{4.2}$$

$$A_1 = \begin{bmatrix} 0.98 & 0.02 & 0 & 0 & 0 \\ 0 & 0.98 & 0.02 & 0 & 0 \\ 0 & 0 & 0.98 & 0.02 & 0 \\ 0 & 0 & 0 & 0.98 & 0.02 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.3}$$

$$B_1 = \begin{bmatrix} 0.8 & 0.2 & 0 & 0 & 0 \\ 0.1 & 0.85 & 0.05 & 0 & 0 \\ 0 & 0.1 & 0.85 & 0.05 & 0 \\ 0 & 0 & 0.1 & 0.85 & 0.05 \\ 0 & 0 & 0 & 0.2 & 0.8 \end{bmatrix}, \tag{4.4}$$

$$\pi_1 = [0.2 \quad 0.2 \quad 0.2 \quad 0.2 \quad 0.2]. \tag{4.5}$$

For the second HMM model based on the ANN observations, equations (4.6)-(4.9) define its form and parameters.

$$\lambda_{ANN} = (A_2, B_2, \pi_2), \tag{4.6}$$

$$A_2 = \begin{bmatrix} 0.98 & 0.02 & 0 & 0 & 0 \\ 0 & 0.98 & 0.02 & 0 & 0 \\ 0 & 0 & 0.98 & 0.02 & 0 \\ 0 & 0 & 0 & 0.98 & 0.02 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.7}$$

$$B_2 = \begin{bmatrix} 0.8 & 0.2 & 0 & 0 & 0 \\ 0.1 & 0.85 & 0.05 & 0 & 0 \\ 0.02 & 0.06 & 0.85 & 0.05 & 0.02 \\ 0 & 0 & 0.1 & 0.85 & 0.05 \\ 0 & 0 & 0 & 0.2 & 0.8 \end{bmatrix}, \tag{4.8}$$

$$\pi_2 = [0.2 \quad 0.2 \quad 0.2 \quad 0.2 \quad 0.2]. \tag{4.9}$$

The derivation of the state transition probability distributions (A_1 and A_2) relies on the average lifespan of a capacitor. It assumes that the capacitor with the characteristics corresponding to the current state remains in this state for the 98% of the lifetime indicated for this state. As the capacitor ages, its characteristics change, and within the remaining 2% of the lifetime, they cross the boundary of the values between the current state and the next state, and the capacitor transits to a higher age state. When the capacitor reaches the upper age category limit, it remains in this state for the 100% of this state's lifetime. The zero elements in the state transition probability matrix indicate that the aging process is a forward movement, through which the capacitor cannot transit to the previous lower age state.

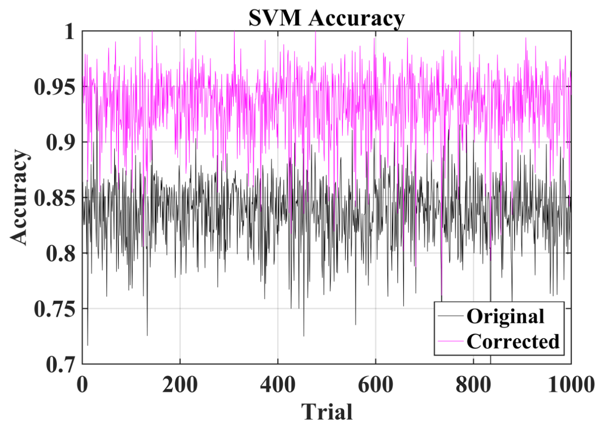
The observation probability distributions (B_1 and B_2) represent the probabilities of the true capacitor state being estimated by the ML models as each of the five capacitor classes. The probabilities calculation relies on the number of correctly and incorrectly classified cases extracted from the SVM and ANN confusion matrices (Fig. 4.7c-f and Fig. 4.8c-f).

The initial state distribution (π_1 and π_2) for both models represent that each of the five capacitor classes can be an initial state of the DC-link capacitor with equal probability.

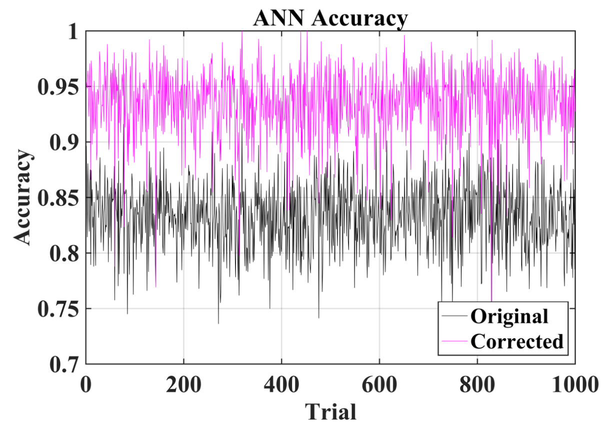
Both HMM models have the same transition probability distribution and the initial state distribution because they are defined by the system's physics. However, the observation probability distributions are different. They reflect the fact that the SVM performed on the smoothed spectral data confuses only the adjacent capacitor classes, while the ANN may also assign any of the five classes to the true class 3.

After defining the HMMs parameters, we developed an output correction method to estimate the most likely state sequence in the model that produces the observed output sequence. This method's concept is holding on the current classification decision until the next measurement is available. If the next classification result does not resolve the uncertainty about the system state, the algorithm processes one more measurement. A sequence of corrected classification decisions referred to the estimated state sequence determines the DC-link condition at a time of each measurement.

We initially modeled the HMMs to test the efficiency of the error correction method. With 1000 trials, the average accuracy improves from 83.6% to 93.3% for the SVM and from 83% to 93% for the ANN, and the accuracy of corrected states is always higher than original accuracy, as demonstrated in Fig. 4.10.



(a)



(b)

Fig. 4.10: Simulation results: original and HMM-corrected ML accuracy for (a) SVM and (b) ANN.

We then applied the error correction method to the experimental data, namely, the original ML classification sequences. The results in Fig. 4.11-4.12 show that both the SVM and the ANN correction enhance the original ML classification accuracy for all investigated types of smoothed spectral data. The confusion matrices in Fig. 4.13-4.14 demonstrate that the accuracy improvement is due to the reduced misclassification of adjacent classes. Besides, the ANN correction eliminates the confusion between class 3 and classes 1 and 5.

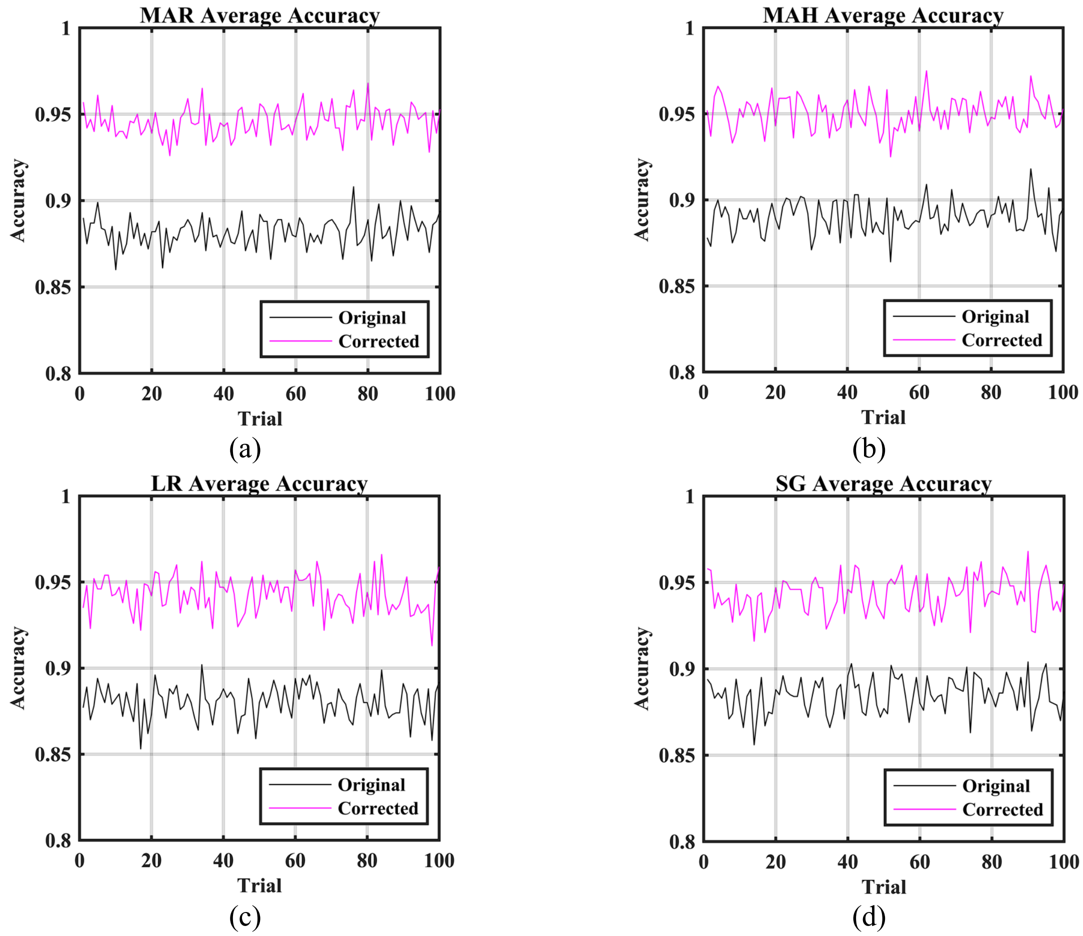
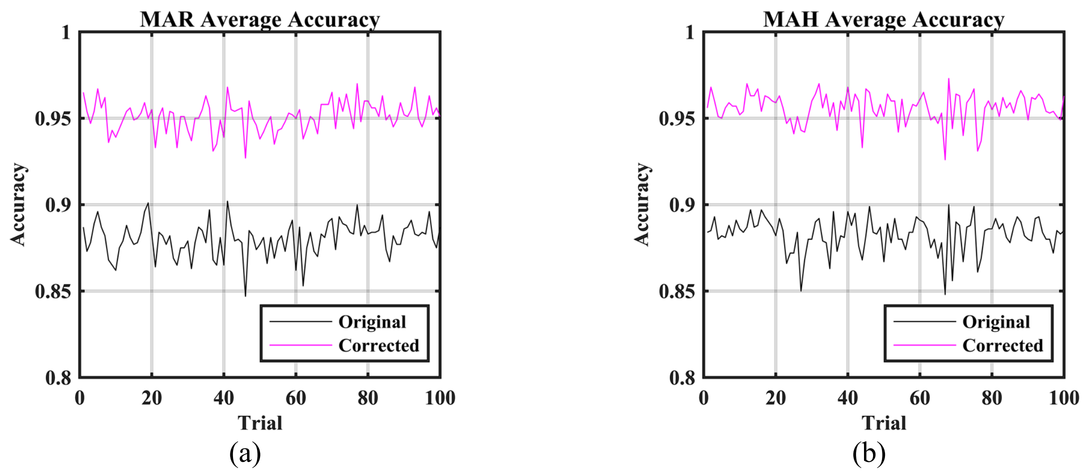
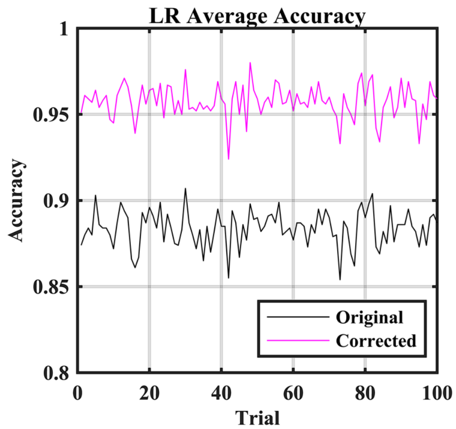
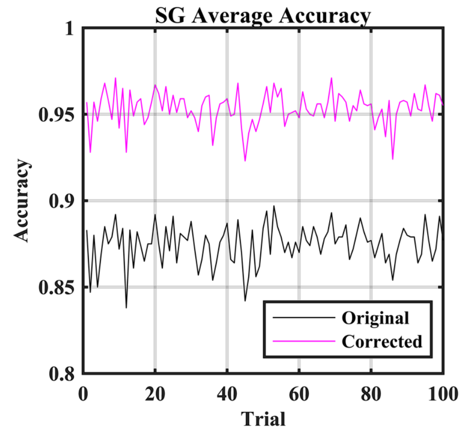


Fig. 4.11: Experimental results: original and HMM-corrected SVM accuracy for different data types: (a) **MAR**, (b) **MAH**, (c) **LR**, (d) **SG**.



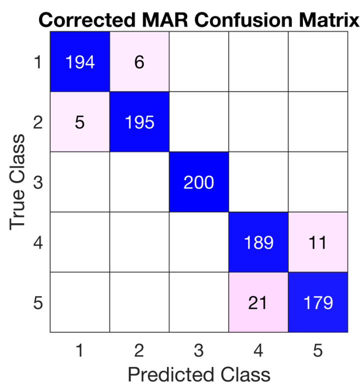


(c)

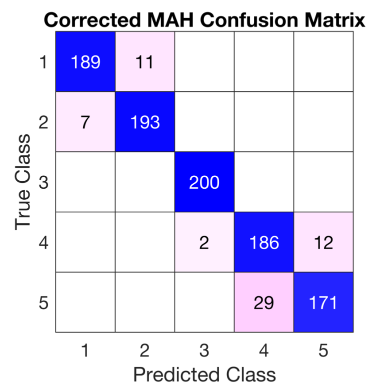


(d)

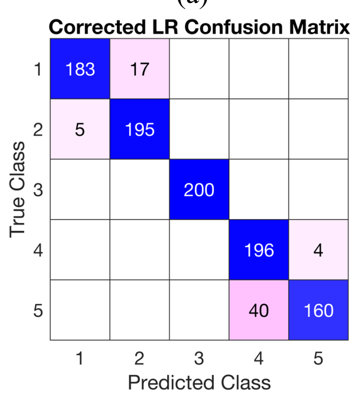
Fig. 4.12: Experimental results: original and HMM-corrected ANN accuracy for different data types: (a) **MAR**, (b) **MAH**, (c) **LR**, (d) **SG**.



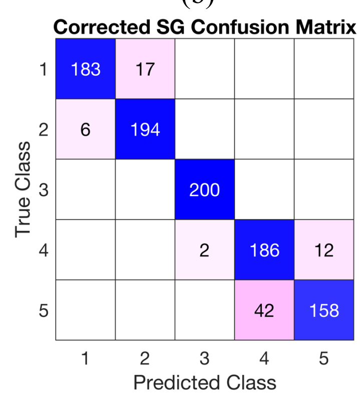
(a)



(b)



(c)



(d)

Fig. 4.13: Test confusion matrices, HMM-corrected SVM results using different data types: (a) **MAR**, (b) **MAH**, (c) **LR**, (d) **SG**.

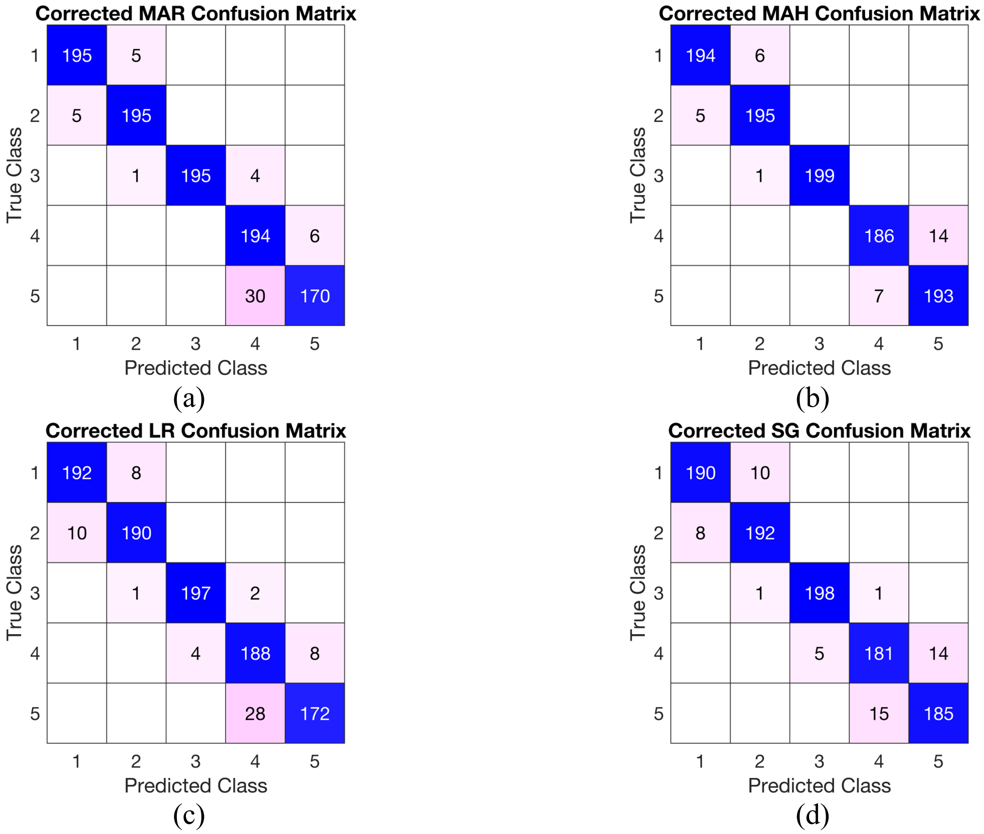


Fig. 4.14: Test confusion matrices, HMM-corrected ANN results using different data types: (a) **MAR**, (b) **MAH**, (c) **LR**, (d) **SG**.

The ML classification results obtained in stage 3 of the CM procedure indicate that using smoothed spectral data, the SVM and the ANN can estimate the age class of a DC-link capacitor with the average accuracy of 88.4% and 88.0%, respectively, and the maximum achieved accuracy is 91.8% (see Table 4.6).

Table 4.8 provides the performance of the HMM-supported ML classification on the analyzed data types. With incorporated HMM in stage 4, the average accuracy increases to 94.5% for the SVM and to 95.4% for the ANN. Thus, the HMM-based output correction can provide the average accuracy improvement of 6.1% for the SVM and 7.4% for the ANN. A higher percentage improvement for the ANN over the SVM is due to the elimination of the nonadjacent misclassification in addition to the reduction of the adjacent classification error. The best result recorded for the HMM-supported ML classification performance is 98.0%.

Table 4.8: HMM-supported ML Classification Performance

Data Type	SVM		ANN	
	Average Test Accuracy, %	Maximum Test Accuracy, %	Average Test Accuracy, %	Maximum Test Accuracy, %
MAR	94.55	96.80	95.10	97.00
MAH	95.06	97.50	95.60	97.30
LR	94.24	96.60	95.74	98.00
SG	94.25	96.80	95.32	97.10

4.7 Summary

This chapter presents the HMM-supported ML CM procedure for the DC-link in a three-phase inverter. It provides the experimental results to evaluate the efficiency of the approach and to validate the dependence of the DC-link capacitance's age on the externally monitored EMI behavior. It demonstrates that with the information only from a CT, the HMM-supported ML can estimate the capacitor health with a 98% maximum accuracy.

The study of the presented approach indicates that the smoothed spectral data characterized by the substantially decreased noise variance results in a higher ML performance than the original spectral data. The comparison of the smoothing DSP techniques reveals that the 1000-point MARF is the most optimal filter for the proposed methodology, because it requires a lower number of operations for its implementation than the other investigated smoothing algorithms with the same window size.

The results show that both the SVM and the ANN can classify the DC-link capacitor's age condition after the current signal is converted to frequency-domain and the resulting spectrum is smoothed. The comparative analysis of the ML techniques demonstrates that on average, the SVM is a better classification tool that the ANN. The output DC-link age class estimated by the SVM or the ANN is a reliable indicator of the true state with an 88.2% average accuracy. When the output sequence of measurements is available, it provides additional information about the circuit state. The HMM built on the known system probabilities generates more accurate estimates after one or several additional measurement cycles. The HMM-supported ML estimates the DC-link capacitor age class with an average accuracy of 94.9%.

Chapter 5 Conclusion and Future Work

This chapter serves to provide a summary of the entire body of this thesis and the research's contributions. It also includes a description of future research directions that this study produced.

5.1 Conclusion

This thesis explores the E-PHM HMM-supported ML approach for the CM of a DC-link capacitor in a three-phase inverter. The approach uses non-invasive current measurements from a CT to estimate the capacitor health. The thesis starts with an introduction to the PHM systems and highlights the importance of developing non-invasive CM approaches with the goal of health estimation of power electronics systems. Then it provides the description of the proposed approach.

Chapter 2 presents the literature review on the currently developed methods of non-invasive diagnostics for power systems and DC-link capacitors. Moreover, it highlights some of the crucial benefits associated with the EMI-based technique that make it a viable solution for CM of DC-link capacitors in a three-phase inverter.

Chapter 3 includes the detailed theoretical analysis of the degradation behavior of capacitors and the suggested E-PHM system. It also presents the simulation results of the system model. They support theoretically established correlation between the current spectral content behavior and the capacitor age. Then this chapter describes the DSP algorithms integrated in the processing of experimental data. It provides the details on the ML models for data classification. In addition, it discusses the principles of the HMM used to enhance the ML classification results.

Chapter 4 describes the proposed CM procedure. The experimental part includes data acquisition and further preprocessing, followed by capacitor health estimation using the SVM and the ANN and the output HMM-based correction. This chapter also demonstrates the efficiency of the proposed methodology through the high accuracy of classification results obtained from both ML approaches.

This research's results show that:

- In a high-frequency range (10's MHz), the magnitudes of the EMI current oscillation peaks decrease as the capacitor age increases. These oscillations are due to the resonating of the MOSFETs' parasitic capacitances with their stray inductances and the DC-link's equivalent series inductance. As the capacitor ages, its ESR and damping factor increase accordingly. It reduces the peak overshoot of the MOSFET's drain-source voltage, and the magnitude of the current peak decreases correspondingly. The consistency of theoretical, simulation, and experimental results validate this relationship.

- With the information only from a CT, the DC-link capacitor age class can be estimated with a 94.9% average accuracy. In comparison with the approach that uses measurements form LISNs [58], the new approach avoids using these devices, thereby reducing space and costs for the CM implementation. Moreover, the new solution's average estimation accuracy is higher by 6.1% than the accuracy of the solution presented in [58].

- The optimal DSP strategy for the proposed approach is to convert current data to frequency-domain using the FFT and smooth the resulting spectrum with the 1000-point MARF. Table 5.1 compares six investigated data types with application to the E-PHM HMM-supported ML approach for CM of DC-link capacitors. The time-domain signal (data type **X**) is not suitable for this approach because it results in a low ML accuracy and the confusion of each predicted class with the rest of the classes. The current spectrum (data type **F**) is also not suitable for CM because it leads to the confusion between classes 3 and 5. The spectrum smoothed with the MARF (data type **MAR**), the MAHF (data type **MAH**), the LRF (data type **LR**), or the SGF (data type **SG**) is suitable for the generation of reliable capacitor health estimates. These four data types allow to achieve the high and relatively similar classification accuracy, with deviations of less than 1%. Among them, **MAR** is the preferred data type because its generation involves fewer operations than for **MAH**, **LR**, and **SG**.

Table 5.1: Comparison of Data Types

Data Type	Maximum achieved accuracy	Characteristics	Comments on application for CM
X	20.0%	Confusion between all classes	Not suitable
F	75.3%	Confusion between classes 3 and 5	Not suitable
MAR	97.0%	Number of operations {1; 999}	Preferred
MAH	97.5%	Number of operations {1001; 999}	Suitable
LR	98.0%	Number of operations {11008; 5998}	Suitable
SG	97.1%	Number of operations {1001; 999}	Suitable

- Either the SVM or the ANN can classify capacitor health with high accuracy. However, on average, the SVM is a better classification tool that the ANN because, first, the SVM produces more accurate estimates, and second, it confuses only adjacent classes, while the ANN also confuses class 3 with classes 1 and 5.

- The accuracy of the HMM-supported ML DC-link age class estimation is higher than the original ML classification accuracy. This is due to the HMM-based output correction that reduces the number of misclassified cases. The correction method performs similarly on all four smoothed spectral data types (**MAR**, **MAH**, **LR**, **SG**), improving their ML classification accuracy by 6.1% using the SVM and by 7.4% using the ANN. The correction of both the SVM and the ANN outputs reduces the error in the classification of the adjacent classes. In addition, the correction method applied to the ANN results eliminates the misclassification of nonadjacent classes, namely of class 3 and classes 1 and 5. Therefore, the HMM-based correction of the ANN outputs provides a higher accuracy improvement than the SVM output correction. The highest accuracy of the HMM-supported ML results is 98% achieved using the ANN on data type **LR**.

5.2 Future Work

This research evaluates the accuracy of the HMM-supported ML approach for the three-phase inverter operated under the conditions specified in Table 4.3. In order to obtain a more holistic and comprehensive insight into the method's robustness, additional experiments should be conducted. With the same test setup, the data should be collected from the inverter with different operational parameters. The conditions presented in Table 5.2 offer a possible starting point for this investigation. These experiments should examine how the variations in the parameters affect the performance of the ML algorithms. The changes in the input voltage and current alter the values of the EMI current oscillation peaks magnitudes. This can influence the magnitude differences between the capacitor classes, and hence, the ability of the ML models to distinguish them. Variations in the modulation index and switching frequency shift the position of the oscillation peaks. Examination of different values of these parameters should indicate if the suggested signal record length is enough to capture the appropriate length of a signal for its successful ML processing. In addition, DC-links with different capacitance and ESR values should be tested to quantify the limitations of the proposed approach.

Table 5.2: Three-phase Inverter Operational Parameters in Future Experiments

Experiment #	Variated Parameter	Tested Value	Value to be Tested	Unit
1	Input voltage	540	520	V
2	Input current	7.1	3.5	A
3	Modulation index	0.9	0.7	-
4	Switching frequency	20	22.5	kHz
5	Input current	7.1	3.5	A
	Switching frequency	20	30	kHz

The analyzed EMI signal represents the total noise of the EMI. The future work should carry out the common mode and differential mode EMI measurements to identify and compare their influence on the DC-link age estimation accuracy.

Another concern is the sensitivity of the algorithm to the error in measurements. An additional theoretical analysis of the circuit in Fig. 3.6 should be conducted to identify the dependence of the current peak on the sampling frequency and resolution.

This research analyzes a second-order capacitor model. There is a need to examine more complex models to determine how such characteristics as leakage current change with the capacitor age. Moreover, the complete removal of the LISNs from the experiment and the impact of different power sources should be studied.

This research identifies just the capacitor aging relationship with the EMI spectrum. The future study should determine the impact of aging in other components, such as MOSFETs, on the E-PHM.

One more potential study can address the feasibility of implementing the proposed CM algorithm in commercial applications. It should carry out a computational analysis of the algorithm and introduce an embedded system to perform online CM.

References

[1] International Air Transport Association, “Airline maintenance cost executive commentary,” *Maintenance Cost Task Force*, Montreal, CA, Nov. 2018.

[2] G. M. Buiatti, J. A. MartÍn-Ramos, A. M. R. Amaral, P. Dworakowski, and A. J. M. Cardoso, “Condition monitoring of metallized polypropylene film capacitors in railway power trains,” *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 10, pp. 3796-3805, Oct. 2009.

[3] M. Wong, “Melbourne’s Exploding Siemen's Trains,” [Online] Available: https://wongm.com/2015/01/melbourne-exploding-siemens-trains/

[4] Rail Accident Report – Explosion inside an underframe equipment case at Guilford, 7 July 2017, Rail Accident Investigation Branch, Report 05/2018. [Online] Available: https://assets.publishing.service.gov.uk/media/5acb313ded915d5a90e44be4/R052018_180320_Guildford.pdf

[5] S. Sawyer, Q. Liming, L. Fried, “Global Wind Report - Annual Market Update 2017,” Global Wind Energy Council, May 2018.

[6] R.H. Wiser and M.H. Bolinger, “2017 Wind Technologies Market Report,” U.S. Department of Energy, Aug. 2018.

[7] B. Lu, Y. Li, X. Wu, and Z. Yang, “A review of recent advances in wind turbine condition monitoring and fault diagnosis,” *2009 IEEE Power Electronics and Machines in Wind Applications*, Lincoln, NE, 2009, pp. 1-7.

[8] S. Mollov and F. Blaabjerg, “Condition and health monitoring in power electronics,” *2018 10th International Conference on Integrated Power Electronics Systems (CIPS)*, Stuttgart, Germany, 2018, pp. 1-8.

[9] E. Vinberg, M. Martin, A.H. Firdaus, Y. Tang, and A. Qazizadeh, “Railway applications of condition monitoring,” KTH Royal Institute of Technology, Stockholm, Jan. 2018.

[10] S. Manohar, A. Sahoo, A. Subramaniam, and S. K. Panda, “Condition monitoring of power electronic converters in power plants — a review,” *2017 20th International Conference on Electrical Machines and Systems (ICEMS)*, Sydney, NSW, 2017, pp. 1-5.

[11] S. Yang, D. Xiang, A. Bryant, P. Mawby, L. Ran, and P. Tavner, “Condition monitoring for device reliability in power electronic converters: a review,” *IEEE Trans. on Power Electron.*, vol. 25, no. 11, pp. 2734-2752, Nov. 2010.

[12] M. Pecht, “Prognostics and health management of electronics,” *Encyclopedia of Structural Health Monitoring*, pp. 1-13, Sept. 2009.

[13] Y. Han and Y. H. Song, “Condition monitoring techniques for electrical equipment - a literature survey,” *IEEE Trans. Power Del.*, vol. 18, no. 1, pp. 4-13, Jan. 2003.

[14] S. A. Mcinerny and Y. Dai, “Basic vibration signal processing for Bearing Fault Detection,” *IEEE Trans. on Education*, vol. 46, no. 1, pp. 149-156, Feb. 2003.

[15] R. R. Schoen, T. G. Habetler, F. Kamran, and R. G. Bartheld, "Motor bearing damage detection using stator current monitoring," *IEEE Trans. on Industry Applications*, vol. 31, no. 6, pp. 1274-1279, Nov.-Dec. 1995.

[16] A. Shrivastava, S. Wadhwani, "Vibration signature analysis for ball bearing of three phase induction motor," *IOSR Journal of Electrical and Electronics Engineering*, vol. 1, no. 3, pp. 46-50, 2012.

[17] R. Yan and R. Gao, "Hilbert–Huang transform-based vibration signal analysis for machine health monitoring," *IEEE Trans. on Instrumentation and Measurement*, vol. 55, no. 6, pp. 2320-2329, Dec. 2006.

[18] A. Soualhi, K. Medjaher, and N. Zerhouni, "Bearing health monitoring based on Hilbert–Huang transform, support vector machine, and regression," *IEEE Trans. on Instrumentation and Measurement*, vol. 64, no. 1, pp. 52-62, Jan. 2015.

[19] R. Yan and R. X. Gao, "An efficient approach to machine health diagnosis based on harmonic wavelet packet transform," *Robotics and Computer Integrated Manufacturing*, vol. 21, no.4-5, pp. 291-301, 2005.

[20] X. Chiementin, F. Bolaers, and JP Dron, "Early detection of fatigue damage on rolling element bearings using Adapted Wavelet," *Journal of Vibration and Acoustics*, vol. 129, no. 4, pp. 495-506, 2007.

[21] W. Yang, P.J. Tavner, and M.R. Wilkinson, "Condition monitoring and fault diagnosis of a wind turbine synchronous generator drive train," *IET Renewable Power Generation*, vol. 3, no. 1, pp. 1-11, 2009.

[22] W. Yang, P.J. Tavner, C.J. Crabtree, and M. Wilkinson, "Research on a simple, cheap but globally effective condition monitoring technique for wind turbines," *Proceedings of the 2008 International Conference on Electrical Machines*, pp. 1-5, 2008.

[23] S. Bagavathiappan, B. B. Lahiri, T. Saravanan, J. Philip, and T. Jayakumar, "Infrared thermography for condition monitoring – a review," *Infrared Physics and Technology*, vol. 60, pp. 35-55, 2013.

[24] Y.C. Chou and L. Yao, "Automatic diagnostic system of electrical equipment using infrared thermography," *2009 International Conference of Soft Computing and Pattern Recognition (SOCPAR),* 2009, pp. 155-160.

[25] M. S. Jadin and S. Taib, "Recent progress in diagnosing the reliability of electrical equipment by using infrared thermography," *Infrared Physics and Technology*, vol. 55, no. 4, pp. 236–245, July 2012.

[26] V.A. Kumar, A.K. Ray, S.P. Singh, D. Banerjee, and S. Samuel, "A review of recent advances in the use of thermography in pulp and paper industry," *Indian Pulp and Paper Technical Association (IPPTA)*, vol.21, no.2, pp. 55-58, Mar. 2009.

[27] S. Taib, M. Shawal, and S. Kabir, "Thermal imaging for enhancing inspection reliability: detection and characterization," Infrared Thermography, InTech, pp. 209-236, Mar. 2012.

[28] C. Ibarra-Castanedo, M. Genest, J.-M. Piau, S. Guibert, A. Bendada, and X. P. V. Maldague, "Active infrared thermography techniques for the nondestructive testing of materials," *Ultrasonic and Advanced Methods for Nondestructive Testing and Material Characterization*, pp. 325-348, 2007.

[29] C.A.L. Almeida *et al.*, "Intelligent thermographic diagnostic applied to surge arresters: a new approach," *IEEE Trans. on Power Delivery*, vol. 24, no. 2, pp. 751-757, 2009.

[30] M.A. Shafii and N. Hamzah, "Internal fault classification using Artificial Neural Network," *2010 4th International Power Engineering and Optimization Conference (PEOCO)*, pp. 352-357, 2010.

[31] B. Li, X. Zhu, S. Zhao, and W. Niu, "HV power equipment diagnosis based on infrared imaging analyzing," *2006 Int. Conf. on Power System Technology*, pp. 1-4, Oct. 2006.

[32] A. Bandes, "Ultrasonic condition monitoring in power plants," *ASME 2009 Power Conference (POWER 2009)*, pp. 345-351, 2009.

[33] A. Rienstra and J. Hall, "Ultrasonic detection," *The Journal of Machinery Fluids, Tests and Standards*, vol. 3, no. 2, pp. 1-4, 2002.

[34] J.M. Buckley, "An overview of the predictive maintenance applications of airborne ultrasound testing," *4th Int. Conf. on NDT*, Oct. 2007.

[35] A. Rienstra and S.D.T.N. America, "Airborne ultrasound : predictive maintenance for the masses," *SDT North America*, pp. 1-17, 2005.

[36] R. Raišutis, E. Jasiūnienė, R. Šliteris, and A. Vladišauskas, "The review of non-destructive testing techniques suitable for inspection of the wind turbine blades," *Ultragarsas*, vol. 63, pp. 26-30, 2008.

[37] R. Raišutis, E. Jasi, and E. Žukauskas, "Ultrasonic NDT of wind turbine blades using guided waves," *Ultragarsas (ultrasound)*, vol. 63, no. 1, pp. 7-11, 2008.

[38] D. Mba, "Development of acoustic emission technology for condition monitoring and diagnosis of rotating machines: bearings, pumps, gearboxes, engines, and rotating structures," *The Shock and Vibration Digest,* vol. 38, no. 1, pp. 3-16, Jan. 2006.

[39] L. Alfayez and D. Mba, "Detection of incipient cavitation and determination of the best efficiency point for centrifugal pumps using acoustic emission," *Proc. of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, vol. 219, no. 4, pp. 327-344, 2005.

[40] M. Kaphle, "Analysis of acoustic emission data for accurate damage assessment for structural health monitoring," Ph.D. Thesis, Queensland University of Technology, 2012.

[41] L. Gao, F. Zai, S. Su, H. Wang, P. Chen, and L. Liu, "Study and application of acoustic emission testing in fault diagnosis of low-speed heavy-duty gears," *Sensors (Basel)*, vol. 11, no. 1, pp. 599-611, Jan. 2011.

[42] A.R. Oskouei, M. Ahmadi, and M. Hajikhani, “Wavelet-based acoustic emission characterization of damage mechanism in composite materials under mode I delamination at different interfaces,” *Express Polymer Letters*, vol. 3, no. 12, pp. 804-813, 2009.

[43] W. I. M. Sweldens, “The lifting scheme : a construction of second generation wavelets,” *SIAM Journal on Mathematical Analysis*, vol. 29, no. 2, pp. 511-546, 1998.

[44] M. Surgeon, C. Buelens, and M. Wevers, “Waveform based analysis techniques for the reliable acoustic emission testing of composite structures,” *Proc. of 24th European Conference on Acoustic Emission Testing EWGAE*, pp. 63-68, May 2000.

[45] D. Xiang, L. Ran, P. Tavner, S. Yang, A. Bryant and P. Mawby, “Condition monitoring power module solder fatigue using inverter harmonic identification,” *IEEE Trans. on Power Electronics*, vol. 27, no. 1, pp. 235-247, Jan. 2012.

[46] L. Satish, “Short time Fourier and Wavelet transform for fault detection in power transformers during impulse tests,” *Science, Measurement and Technology, IEE Proceedings*, vol. 145, no. 2, pp. 77-84, Mar. 1998.

[47] W. Qiao and L. Qu, “Prognostic condition monitoring for wind turbine drivetrains via generator current analysis,” *Chinese Journal of Electrical Engineering*, vol. 4, no. 3, pp. 80-89, Sept. 2018.

[48] J. Lehmann, M. Netzel, R. Herzer, and S. Pawel, “Method for electrical detection of bond wire lift-off for power semiconductors,” *ISPSD 03 2003 IEEE 15th Int. Symp. Power Semicond. Devices ICs 2003 Proc.*, pp. 333-336, 2003.

[49] Y. Xiong, X. Cheng, Z. J. Shen, C. Mi, H. Wu, and V. Garg, “A Prognostic and warning system for power electronic modules in electric, hybrid, and fuel cell vehicles,” *Ind. Appl. Conf. 2006. 41st IAS Annu. Meet. Conf. Rec. 2006 IEEE*, vol. 3, no. c, pp. 1578-1584, 2006.

[50] V. Sankaran and X. Xu, “Integrated power module diagnostic unit,” U.S. Patent 5528446, 1996.

[51] J. E. Timperley and J. M. Vallejo, “Condition assessment of electrical apparatus with EMI diagnostics,” *IEEE Trans. on Industry Applications*, vol. 53, no. 1, pp. 693-699, Jan.-Feb. 2017.

[52] S. Pu, E. Ugur, B. Akin and H. Akca, “Investigation of EM radiation changes in SiC based converters throughout device aging,” *2017 IEEE 5th Workshop on Wide Bandgap Power Devices and Applications (WiPDA)*, Albuquerque, NM, 2017, pp. 190-194.

[53] T. Marwala, “Condition monitoring using computational intelligence methods. Applications in mechanical and electrical systems,” Springer, London, 2012.

[54] J. Chen and S. Zou, “An intelligent condition monitoring approach for spent nuclear fuel shearing machines based on noise signals,” *Applied Sciences*, vol. 8, no. 5, 838 , May 2018.

[55] R.M. Tallam, T.G. Habetler and R.G. Harley, “Self-commissioning training algorithms for neural networks with applications to electric machine fault diagnostics,” *IEEE Trans. on Power Electronics*, vol. 17, no. 6, pp. 1089-1095, Nov. 2002.

[56] S. Mohagheghi, R.G. Harley, T.G. Habetler, and D.Divan, "Condition monitoring of power electronic circuits using Artificial Neural Networks," *IEEE Trans. Power Electron.*, vol. 24, no. 10, pp. 2363-2367, 2009.

[57] H. Malik and S. Mishra, "Fault identification of power transformers using Proximal Support Vector Machine (PSVM)," *2014 IEEE 6th India International Conference on Power Electronics (IICPE)*, Kurukshetra, 2014, pp. 1-5.

[58] M. Boubin, J.P. Doran, W. Guo, Y. Rajasekhar, and M. Scott, "EMI diagnostics of three phase inverters using machine learning algorithms," *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*, Portland, OR, 2018, pp. 4062-4069.

[59] T. Sorsa, H. Koivoand, H. Koivisto, "Neural networks in process fault diagnosis," *IEEE Trans. on Systems, Man, and Cybernetics 21*, pp. 815-825, 1991.

[60] M. Beibei, S. Yanxia, W. Dinghui and Z. Zhipu, "Three level inverter fault diagnosis using EMD and support vector machine approach," *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Siem Reap, 2017, pp. 1595-1598.

[61] L.E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Stat.*, vol. 37, pp. 1554-1563, 1966.

[62] L. Abdesselam and C. Guy, "Diagnosis of induction machine by time frequency representation and hidden Markov modelling," *2007 IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives*, Cracow, 2007, pp. 272-276.

[63] R. Yan and J. Cai, "Application of hidden Markov model to fault diagnosis of power electronic circuit," *2009 IEEE Circuits and Systems International Conference on Testing and Diagnosis*, Chengdu, 2009, pp. 1-4.

[64] M. H. Hafizi and A. Izadian, "Model-based fault diagnosis of a DC-DC boost converters using hidden Markov model," *IEEE International Conference on Electro-Information Technology , EIT 2013*, Rapid City, SD, 2013, pp. 1-4.

[65] A. Soualhi, G. Clerc, H. Razik, and A. Lebaroud, "Fault detection and diagnosis of induction motors based on hidden Markov model," *2012 XXth International Conference on Electrical Machines*, Marseille, 2012, pp. 1693-1699.

[66] H. Wang, M. Liserre, and F. Blaabjerg, "Toward reliable power electronics — challenges, design tools and opportunities," *IEEE Ind. Electron. Mag.*, vol. 7, no. 2, pp. 17-26, Jun. 2013.

[67] H. Wang and F. Blaabjerg, "Reliability of capacitors for DC-link applications in power electronic converters — an overview," *IEEE Trans. on Industry Applications*, vol. 50, no. 5, pp. 3569-3578, Sept.-Oct. 2014.

[68] H. Soliman, H. Wang, and F. Blaabjerg, "A review of the condition monitoring of capacitors in power electronic converters," *IEEE Transactions on Industry Applications*, vol. 52, no. 6, pp. 4976-4989, Nov.-Dec. 2016.

[69] M.W. Ahmad, P.N. Kumar, A. Arya and S. Anand, "Noninvasive technique for DC-link capacitance estimation in single-phase inverters," *IEEE Trans. on Power Electronics*, vol. 33, no. 5, pp. 3693-3696, May 2018.

[70] L. Ren, C. Gong and Y. Zhao, "An online ESR estimation method for output capacitor of boost converter," *IEEE Trans. on Power Electronics*, vol. 34, no. 10, pp. 10153-10165, Oct. 2019.

[71] A. Imam, D. Divan, R. Harley, and T. Habetler, "Real-time condition monitoring of the electrolytic capacitors for power electronics applications," *Proc. 22nd Annu. Appl. Power Electron. Conf.*, Feb. 2007, pp. 1057–1061.

[72] A.G. Abo-Khalil and D. Lee, "DC-link capacitance estimation in AC/DC/AC PWM converters using voltage injection," *IEEE Trans. on Industry Applications*, vol. 44, no. 5, pp. 1631-1637, Sept.-Oct. 2008.

[73] M.A. Vogelsberger, T. Wiesinger and H. Ertl, "Life-cycle monitoring and voltage-managing unit for DC-link electrolytic capacitors in PWM converters," *IEEE Trans. on Power Electronics*, vol. 26, no. 2, pp. 493-503, Feb. 2011.

[74] N. Agarwal, M.W. Ahmad and S. Anand, "Quasi-online technique for health monitoring of capacitor in single-phase solar inverter," *IEEE Trans. on Power Electronics*, vol. 33, no. 6, pp. 5283-5291, June 2018.

[75] A. Amaral and A. Cardoso, "An experimental technique for estimating the ESR and reactance intrinsic values of aluminum electrolytic capacitors," *Proc. IEEE Instrum. Meas. Technol. Conf.*, Apr. 2006, pp. 1820-1825.

[76] A. Amaral, G. Buatti, H. Ribeiro, and A. Cardoso, "Using DFT to obtain the equivalent circuit of aluminum electrolytic capacitors," *Proc. 7th Int. Conf. Power Electron. Drive Syst.*, Nov. 2007, pp. 434-438.

[77] A. Amaral and A. Cardoso, "A simple offline technique for evaluating the condition of aluminum electrolytic capacitors," *IEEE Trans. Ind. Electron.*, vol. 56, no. 8, pp. 3230-3237, Aug. 2009.

[78] A. Amaral and A. Cardoso, "Using a sinosoidal PWM to estimate the ESR of aluminum electrolytic capacitors," *Proc. Int. Conf. Power Eng., Energy Elect. Drives*, Mar. 2009, pp. 691-696.

[79] D.C. Lee, K.J. Lee, J.K. Seok, and J.W. Choi, "Online capacitance estimation of DC-link electrolytic capacitors for three-phase ac/dc/ac PWM converters using recursive least squares method," *IEE Proc. Elect. Power Appl.*, vol. 152, no. 6, pp. 1503-1508, Nov. 2005.

[80] A. Imam, T. Habetler, R. Harley, and D. Divan, "LMS based condition monitoring of electrolytic capacitor," *Proc. 31st Annu. Conf. IEEE Ind. Electron. Soc.*, Nov. 2005, pp. 848-853.

[81] A. Amaral and A. Cardoso, "An economic offline technique for estimating the equivalent circuit of aluminum electrolytic capacitors," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 12, pp. 2697-2710, Dec. 2008.

[82] A. Imam, T. Habetler, R. Harley, and D. Divan, "Condition monitoring of electrolytic capacitor in power electronic circuits using input current," *Proc. 5th IEEE Int. Symp. Diagn. Elect. Mach. Power Electron. Drives*, Sep. 2005, pp. 1-7.

[83] A. Amaral and A. Cardoso, "Use of ESR to predict failure of output filtering capacitors in boost converters," *Proc. IEEE Int. Symp. Ind. Electron.*, vol. 2, May 2004, pp. 1309-1314.

[84] A. Wechsler, B. Mecrow, D. Atkinson, J. Bennett, and M. Benarous, "Condition monitoring of dc-link capacitors in aerospace drives," *IEEE Trans. Ind. Appl.*, vol. 48, no. 6, pp. 1866-1874, Nov. 2012.

[85] M. Ahmad, A. Arya, and S. Anand, "An online technique for condition monitoring of capacitor in PV system," *Proc. IEEE Int. Conf. Ind. Technol.*, Mar. 2015, pp. 920-925.

[86] W. Lu, X. Lu, J. Han, Z. Zhao, and X. Du, "Online estimation of ESR for DC-Link capacitor of boost PFC converter using wavelet transform based time–frequency analysis method," *IEEE Trans. on Power Electronics*, vol. 35, no. 8, pp. 7755-7764, Aug. 2020.

[87] X.S. Pu, T.H. Nguyen, D.C. Lee, K.B. Lee, and J.M. Kim, "Fault diagnosis of dc-link capacitors in three-phase ac/dc PWM converters by online estimation of equivalent series resistance," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 4118-4127, Sep. 2013.

[88] X. Pu, T.H. Nguyen, D.C. Lee, and S.G. Lee, "Capacitance estimation of dc-link capacitors for single-phase PWM converters," *Proc. IEEE 6th Int. Power Electron. Motion Control Conf.*, May 2009, pp. 1656-1661.

[89] T.H. Nguyen and D.C. Lee, "Deterioration monitoring of dc-link capacitors in ac machine drives by current injection," *IEEE Trans. Power Electron.*, vol. 30, no. 3, pp. 1126-1130, Mar. 2015.

[90] Y.J. Jo, T.H. Nguyen, and D.C. Lee, "Condition monitoring of submodule capacitors in modular multilevel converters," *Proc. IEEE Energy Convers. Congr. Expo.*, Sep. 2014, pp. 2121-2126.

[91] M. Kim, S.K. Sul, and J. Lee, "Condition monitoring of dc-link capacitors in drive system for electric vehicles," *Proc. IEEE Veh. Power Prop. Conf.*, Oct. 2012, pp. 633-637.

[92] H. Soliman, H. Wang, B. Gadalla, and F. Blaabjerg, "Condition monitoring of dc-link capacitors based on artificial neural network algorithm," *Proc. IEEE 5th Int. Conf. Power Eng., Energy Elect. Drives*, May 2015, pp. 587-591.

[93] H. Soliman, P. Davari, H. Wang, and F. Blaabjerg, "Capacitance estimation algorithm based on DC-link voltage harmonics using artificial neural network in three-phase motor drive systems," *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, Cincinnati, OH, 2017, pp. 5795-5802.

[94] T. Kamel, Y. Biletskiy, and L. Chang, "Capacitor aging detection for the dc filters in the power electronic converters using ANFIS algorithm," *Proc. 28th Can. Conf. Elect. Comput. Eng.*, May 2015, pp. 663-668.

[95] A. M. Imam, T. G. Habetler, R. G. Harley and D. Divan, "Failure prediction of electrolytic capacitor using DSP methods," *Twentieth Annual IEEE Applied Power Electronics Conference and Exposition*, *2005. APEC 2005.*, Austin, TX, 2005, vol. 2, pp. 965-970.

[96] D. Morgan, "A handbook for EMC testing and measurement," The Institution of Engineering and Technology, London, United Kingdom, 2007.

[97] V. Roederstein, "Metallized Polypropylene Film Capacitors," Rep. no. MKP1848C65090JY5 Datasheet, Mar. 21, 2016.

[98] M. Gebbia, "Introduction to Film Capacitors," Illinois Capacitor, Inc. 2013.

[99] "Everything you wanted to know about capacitors but were afraid to ask," *PSMA Capacitor Committee and IEEE PELS Capacitor Workshop*, San Antonio, Texas, Mar. 2018.

[100] P. Mach and M. Horak, "Examination of influence of electrical stress on parameters of PP film capacitors," *IEEE 20th International Symposium for Design and Technology in Electronic Packaging*, pp. 51-54. Oct. 2014.

[101] M. Markdessi, A. Sari, P. Venet, P. Bevilacqua, and C. Joubert, "Accelerated aging of metallized film capacitors under high ripple currents combined with a DC voltage," *IEEE Trans. Power Electron.*, vol. 30, no. 5, pp. 2435-2444, May 2015.

[102] J. H. Hayes and T. Hubing, "Preliminary investigation of the current path and circuit parameters associated with the characteristic ringing in a MOSFET power inverter," *Clemson Vehicular Electronics Laboratory Technical Report: CVEL-13-041*, Jan. 16, 2013.

[103] J. Hayes and T. Hubing, "Effects of the lower body diode on the ringing characteristics of a power inverter," *Clemson Vehicular Electronics Laboratory Technical Report: CVEL-13-046*, Aug. 14, 2013.

[104] B. Rehman, M. Ahmad and J. Hussain, "Analysis of power system harmonics using singular value decomposition, least square estimation and FFT," *2014 International Conference on Energy Systems and Policies (ICESP)*, Islamabad, 2014, pp. 1-5.

[105] P. Kowalski and R. Smyk, "Review and comparison of smoothing algorithms for one-dimensional data noise reduction," *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, Swinoujście, 2018, pp. 277-281.

[106] W. Wettayaprasit, N. Laosen, and S. Chevakidagarn, "Data filtering technique for neural networks forecasting," *Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization*, Beijing, China, Sept. 2007, pp. 225-230.

[107] J. F. Kenney and E. S. Keeping, "Mathematics of statistics," 3rd ed., Van Nostrand, Princeton, NJ, USA, pp. 221-223, 1962.

[108] J. Hann, "Handbook of climatology. Part I: general climatology," Macmillan, New York, 1903.

[109] D. M. Hawkins and E. M. Maboudou-Tchao, "Smoothed linear modeling for smooth spectral data," *International Journal of Spectroscopy*, vol. 2013, June 2013.

[110] W. S. Cleveland, “Robust locally weighted regression and smoothing scatterplots,” *Journal of the American Statistical Association*, vol. 74, no. 368, pp. 829-836, 1979.

[111] A. Savitzky and M.J.E. Golay, “Smoothing and differentiation of data by simplified least squares procedures,” *Analytical Chemistry*, vol. 36, no. 8, pp. 1627-1639, 1964.

[112] W. H. Press, S. A. Teukolsky, “Savitzky-Golay Smoothing Filters,” *Computers in Physics*, pp. 669-672, Nov.-Dec. 1990.

[113] A. Burkov, The Hundred-Page Machine Learning Book, Kindle Direct Publishing, 2019.

[114] R. Sathya and A. Abraham, “Comparison of supervised and unsupervised learning algorithms for pattern classification,” *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 34-38, 2013.

[115] A. Belousov, S. Verzakov, and J. von Frese, “Applicational aspects of Support Vector Machines,” *J. of Chemometrics*, vol. 16, no. 8-10, pp. 482-489, Aug. 2002.

[116] P. Vateekul, S. Dendamrongvit, and M. Kubat, “Improving SVM performance in multi-label domains: threshold adjustment,” *Int. J. Artif. Intell. T.*, vol. 22, no.1, Mar. 2013.

[117] R.Sangeetha, Dr.B.Kalpana, “Identifying efficient kernel function in multiclass Support Vector Machines,” *International Journal of Computer Applications*, vol. 28, no. 8, pp. 18-23, Aug. 2011.

[118] A. Rocha and S. K. Goldenstein, “Multiclass from binary: expanding one-versus-all, one-versus-one and ECOC-based approaches,” *IEEE Trans. on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 289-302, Feb. 2014.

[119] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, London, 1995.

[120] S. Knerr, L. Personnaz, and G. Dreyfus, “Single-layer learning revisited: a stepwise procedure for building and training a neural network,” *NATO Neurocomputing*, 1989.

[121] J.H. Friedman, “Another approach to polychotomous classification,” Technical report, Department of Statistics, Stanford University, 1996.

[122] C. Hsu and C. Lin, “A comparison of methods for multi-class Support Vector Machines,” *IEEE Trans Neural Netw*, vol.13, no.2, pp. 415-425, Mar. 2002.

[123] X. Gu, L. Liu, J. Li, Y. Huang and J. Lin, “Data classification based on artificial neural networks,” *2008 International Conference on Apperceiving Computing and Intelligence Analysis*, Chengdu, 2008, pp. 223-226.

[124] G. Singh and M. Sachan, “Multi-layer perceptron (MLP) neural network technique for offline handwritten Gurmukhi character recognition,” *2014 IEEE International Conference on Computational Intelligence and Computing Research*, Coimbatore, 2014, pp. 1-5.

[125] D. Lo, C. Wei, and E. Tsai, “Parameter automatic calibration approach for Neural-Network-based cyclonic precipitation forecast models,” *Water*, vol.7, pp. 3963-3977, Jul. 2015.

[126] L. R. Rabiner, “A tutorial on Hidden Markov Models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, Feb. 1989.

[127] M. Aymen, A. Abdelaziz, S. Halim, and H. Maaref, “Hidden Markov Models for automatic speech recognition,” *2011 International Conference on Communications, Computing and Control Applications (CCCA)*, Hammamet, 2011, pp. 1-6.

[128] A. Benkessirat and N. Benblidia, “Fundamentals of feature selection: an overview and comparison,” *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, Abu Dhabi, United Arab Emirates, 2019, pp. 1-6.

[129] V. Sysoeva, “HMM-supported ML for DC-link capacitor CM,” 2020, [Online] Available: https://github.com/sysoevv/HMM-supported_ML_for_DC-link_capacitor_CM/blob/master/CODE.m

[130] Kemet, “C4AE Series, Radial, 2 or 4 Leads, 450 - 1100 VDC, for DC Link,” Rep. no. C4AEJBW5450A3MJ Datasheet, Mar. 3, 2017.

[131] “RF current probe CLCE-400,” Com-Power Corporation Datasheet, rev. D05.30.

[132] H. Liu, M. Cocea, “Semi-random partitioning of data into training and test sets in granular computing context,” *Granular Computing*, vol. 2, no. 4, pp. 357-386, Dec. 2017.

[133] L. J. Qi and N. Alias, “Comparison of ANN and SVM for classification of eye movements in EOG signals,” *J. Phys.: Conf. Ser.*, vol. 971, 2018.

[134] R. Samsudin, A. Shabri and P. Saad, “A comparison of time series forecasting using Support Vector Machine and Artificial Neural Network model,” *Journal of Applied Sciences*, vol. 10, no. 11, pp. 950-958, 2010.

[135] P. Santos, L. F. Villa, A. Reñones, A. Bustillo, J. Maudes, “An SVM-based solution for fault detection in wind turbines,” *Sensors*, vol. 15, no. 3, pp. 5627-5648, 2015.

[136] D. Olson and D. Delen, “Advanced data mining techniques,” Springer, Berlin, Germany, 2008.

Appendix

```
%%                =|>- MIAMI POWER ELECTRONICS LABORATORY              %%
%  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  %
%  MATLAB CODE FOR HIDDEN MARKOV MODEL-SUPPORTED MACHINE LEARNING
%  FOR DC-LINK CAPACITOR AGE CLASSIFICATION
%  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  %
%  AUTHOR                                                          DATE %
%  Viktoriia Sysoeva                                         05/21/2020 %
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - %
%% CAPACITOR CLASSES     INPUT DATASETS
% - 1 - 6 New 0 old          X1
% - 2 - 5 New 1 Old          X2
% - 3 - 3 New 3 Old          X3
% - 4 - 1 New 5 Old          X4
% - 5 - 0 New 6 Old          X5
clear; close all; clc;

%% STAGE 1. PREPROCESSING
%% 1.1. Input Data & Processing Specification
nC=5;               % Number of Classes
tSample=1e-8;       % Sampling period, s

%Parameters for data that requires splitting
nAXS=2;             % Number of acquired signals for each class
split=100;          % Number of signals to be extracted from 1 acquired
signal
AXSlength=10000000; % Length of acquired signal, samples
tXS=100e-03;        % Signal record length, s

%Parameters for data that does not require splitting
nAXT=200;           % Number of acquired signals for each class
XTlength=100000;    % Length of acquired signal, samples
tX=1e-03;           % Signal record length, s

%% 1.2. Parameters Computation
%Parameters for data that requires splitting
XSlength=AXSlength/split;  % Length of extracted signal
sSplit=nAXS*split;         % Number of extracted signals for each class
nXS=nC*sSplit;             % Total Number of extracted signals for all
classes

%Parameters for data that does not require splitting
nXT=nC*nAXT;               % Total Number of acquired signals for all classes

%Parameters for dataset X
s=sSplit+nAXT;             % Number of signals for each class
s_data=nXS+nXT;            % Total number of signals
time=0:tSample:tX-tSample; % Time vector for 1 signal
fs=1./diff(time);
Fs=fs(1);                  % Sampling Rate

%% 1.3. Datasets XS of split data
XS1=Dsplit('X1/',XSlength,sSplit,split);
XS2=Dsplit('X2/',XSlength,sSplit,split);
XS3=Dsplit('X3/',XSlength,sSplit,split);
```

```
XS4=Dsplit('X4/',XSlength,sSplit,split);
XS5=Dsplit('X5/',XSlength,sSplit,split);

%% 1.4. Datasets XT of not split data
XT1=Dread('X1/',XTlength,nAXT);
XT2=Dread('X2/',XTlength,nAXT);
XT3=Dread('X3/',XTlength,nAXT);
XT4=Dread('X4/',XTlength,nAXT);
XT5=Dread('X5/',XTlength,nAXT);

%% 1.5. Dataset of type X - time-domain data
X1(:,1:sSplit)=XS1;
X2(:,1:sSplit)=XS2;
X3(:,1:sSplit)=XS3;
X4(:,1:sSplit)=XS4;
X5(:,1:sSplit)=XS5;

X1(:,sSplit+1:s)=XT1;
X2(:,sSplit+1:s)=XT2;
X3(:,sSplit+1:s)=XT3;
X4(:,sSplit+1:s)=XT4;
X5(:,sSplit+1:s)=XT5;

% Plot Data X
figure
plot(time,X1(:,1),'k');
hold on
plot(time,X2(:,1),'r');
hold on
plot(time,X3(:,1),'g');
hold on
plot(time,X4(:,1),'m');
hold on
plot(time,X5(:,1),'c');
title('Data X, Experimental Results');
xlabel('Time (s)')
ylabel('Amplitude (A)')
legend({'N^{6}A^{0}','N^{5}A^{1}','N^{3}A^{3}','N^{1}A^{5}',...
'N^{0}A^{6}'},'Location','northeast')
hold off

%% 1.6. Datasets of type F, MAR, MAH, LR, SG
% F   - Single-sided FFT Spectrum
% MAR - Spectrum smoothed with moving average, rectangular window
% MAH - Spectrum smoothed with moving average, Hanning window
% LR  - Spectrum smoothed with locally weighted linear regression
% SG  - Spectrum smoothed with Savitzky-Golay filter

w=1000;   %window length for smoothing filters
SL=50000; %signal length

% Class 1
F_1=zeros(SL,s);
MAR_1=zeros(SL,s);
MAH_1=zeros(SL,s);
LR_1=zeros(SL,s);
SG_1=zeros(SL,s);
```

```
for i=1:s
    Y=X1(:,i);
    [f,F]=Spec(Y,Fs);
    [MAR,MAH,LR,SG]=SpecSm(F,w,SL);
    F_1(:,i)=F;
    MAR_1(:,i)=MAR;
    MAH_1(:,i)=MAH;
    LR_1(:,i)=LR;
    SG_1(:,i)=SG;
end

AvF_1=mean(F_1,2);
AvMAR_1=mean(MAR_1,2);
AvMAH_1=mean(MAH_1,2);
AvLR_1=mean(LR_1,2);
AvSG_1=mean(SG_1,2);

% Class 2
F_2=zeros(SL,s);
MAR_2=zeros(SL,s);
MAH_2=zeros(SL,s);
LR_2=zeros(SL,s);
SG_2=zeros(SL,s);

for i=1:s
    Y=X2(:,i);
    [f,F]=Spec(Y,Fs);
    [MAR,MAH,LR,SG]=SpecSm(F,w,SL);
    F_2(:,i)=F;
    MAR_2(:,i)=MAR;
    MAH_2(:,i)=MAH;
    LR_2(:,i)=LR;
    SG_2(:,i)=SG;
end

AvF_2=mean(F_2,2);
AvMAR_2=mean(MAR_2,2);
AvMAH_2=mean(MAH_2,2);
AvLR_2=mean(LR_2,2);
AvSG_2=mean(SG_2,2);

% Class 3
F_3=zeros(SL,s);
MAR_3=zeros(SL,s);
MAH_3=zeros(SL,s);
LR_3=zeros(SL,s);
SG_3=zeros(SL,s);

for i=1:s
    Y=X3(:,i);
    [f,F]=Spec(Y,Fs);
    [MAR,MAH,LR,SG]=SpecSm(F,w,SL);
    F_3(:,i)=F;
    MAR_3(:,i)=MAR;
    MAH_3(:,i)=MAH;
    LR_3(:,i)=LR;
```

```
    SG_3(:,i)=SG;
end

AvF_3=mean(F_3,2);
AvMAR_3=mean(MAR_3,2);
AvMAH_3=mean(MAH_3,2);
AvLR_3=mean(LR_3,2);
AvSG_3=mean(SG_3,2);

% Class 4
F_4=zeros(SL,s);
MAR_4=zeros(SL,s);
MAH_4=zeros(SL,s);
LR_4=zeros(SL,s);
SG_4=zeros(SL,s);

for i=1:s
    Y=X4(:,i);
    [f,F]=Spec(Y,Fs);
    [MAR,MAH,LR,SG]=SpecSm(F,w,SL);
    F_4(:,i)=F;
    MAR_4(:,i)=MAR;
    MAH_4(:,i)=MAH;
    LR_4(:,i)=LR;
    SG_4(:,i)=SG;
end

AvF_4=mean(F_4,2);
AvMAR_4=mean(MAR_4,2);
AvMAH_4=mean(MAH_4,2);
AvLR_4=mean(LR_4,2);
AvSG_4=mean(SG_4,2);

% Class 5
F_5=zeros(SL,s);
MAR_5=zeros(SL,s);
MAH_5=zeros(SL,s);
LR_5=zeros(SL,s);
SG_5=zeros(SL,s);

for i=1:s
    Y=X5(:,i);
    [f,F]=Spec(Y,Fs);
    [MAR,MAH,LR,SG]=SpecSm(F,w,SL);
    F_5(:,i)=F;
    MAR_5(:,i)=MAR;
    MAH_5(:,i)=MAH;
    LR_5(:,i)=LR;
    SG_5(:,i)=SG;
end

AvF_5=mean(F_5,2);
AvMAR_5=mean(MAR_5,2);
AvMAH_5=mean(MAH_5,2);
AvLR_5=mean(LR_5,2);
AvSG_5=mean(SG_5,2);
```

```
% Plot Data F
figure
plot(f,AvF_1,'k');
hold on
plot(f,AvF_2,'r');
hold on
plot(f,AvF_3,'g');
hold on
plot(f,AvF_4,'m');
hold on
plot(f,AvF_5,'c');
title('Data F, Experimental Results');
xlabel('Frequency (Hz)')
ylabel('Amplitude (dBuA)')
legend({'N^{6}A^{0}','N^{5}A^{1}','N^{3}A^{3}','N^{1}A^{5}',...
'N^{0}A^{6}'},'Location','northeast')
hold off

% Plot Data MAR
figure
plot(f,AvMAR_1,'k');
hold on
plot(f,AvMAR_2,'r');
hold on
plot(f,AvMAR_3,'g');
hold on
plot(f,AvMAR_4,'m');
hold on
plot(f,AvMAR_5,'c');
title('Data MAR, Experimental Results');
xlabel('Frequency (Hz)')
ylabel('Amplitude (dBuA)')
legend({'N^{6}A^{0}','N^{5}A^{1}','N^{3}A^{3}','N^{1}A^{5}',...
'N^{0}A^{6}'},'Location','northeast')
hold off

% Plot Data MAH
figure
plot(f,AvMAH_1,'k');
hold on
plot(f,AvMAH_2,'r');
hold on
plot(f,AvMAH_3,'g');
hold on
plot(f,AvMAH_4,'m');
hold on
plot(f,AvMAH_5,'c');
title('Data MAH, Experimental Results');
xlabel('Frequency (Hz)')
ylabel('Amplitude (dBuA)')
legend({'N^{6}A^{0}','N^{5}A^{1}','N^{3}A^{3}','N^{1}A^{5}',...
'N^{0}A^{6}'},'Location','northeast')
hold off

% Plot Data LR
figure
plot(f,AvLR_1,'k');
hold on
```

```
plot(f,AvLR_2,'r');
hold on
plot(f,AvLR_3,'g');
hold on
plot(f,AvLR_4,'m');
hold on
plot(f,AvLR_5,'c');
title('Data LR, Experimental Results');
xlabel('Frequency (Hz)')
ylabel('Amplitude (dBuA)')
legend({'N^{6}A^{0}','N^{5}A^{1}','N^{3}A^{3}','N^{1}A^{5}',...
'N^{0}A^{6}'},'Location','northeast')
hold off

% Plot Data SG
figure
plot(f,AvSG_1,'k');
hold on
plot(f,AvSG_2,'r');
hold on
plot(f,AvSG_3,'g');
hold on
plot(f,AvSG_4,'m');
hold on
plot(f,AvSG_5,'c');
title('Data SG, Experimental Results');
xlabel('Frequency (Hz)')
ylabel('Amplitude (dBuA)')
legend({'N^{6}A^{0}','N^{5}A^{1}','N^{3}A^{3}','N^{1}A^{5}',...
'N^{0}A^{6}'},'Location','northeast')
hold off

%% 1.7. Datasets for ANN

% Matrices of features
ANN_X=zeros(XTlength,s_data);
ANN_F=zeros(SL,s_data);
ANN_MAR=zeros(SL,s_data);
ANN_MAH=zeros(SL,s_data);
ANN_LR=zeros(SL,s_data);
ANN_SG=zeros(SL,s_data);

ANN_X=ANNfeatures(X1,X2,X3,X4,X5,s);
ANN_F=ANNfeatures(F_1,F_2,F_3,F_4,F_5,s);
ANN_MAR=ANNfeatures(MAR_1,MAR_2,MAR_3,MAR_4,MAR_5,s);
ANN_MAH=ANNfeatures(MAH_1,MAH_2,MAH_3,MAH_4,MAH_5,s);
ANN_LR=ANNfeatures(LR_1,LR_2,LR_3,LR_4,LR_5,s);
ANN_SG=ANNfeatures(SG_1,SG_2,SG_3,SG_4,SG_5,s);

% Matrix of targets
ANN_Targets = zeros(nC,s_data);
for i=1:nC
ANN_Targets(i,((i-1)*s+1):(i*s)) = 1;
end

%% 1.8. Datasets for SVM
% Matrices of features
```

```
SVM_X=ANN_X.';
SVM_F=ANN_F.';
SVM_MAR=ANN_MAR.';
SVM_MAH=ANN_MAH.';
SVM_LR=ANN_LR.';
SVM_SG=ANN_SG.';

% Matrix of targets
SVM_Targets = zeros(s_data,1);
for i=1:nC
SVM_Targets((s*(i-1))+1:s*i)=i;
end

%% STAGE 2. MACHINE LEARNING CLASSIFICATION

%% 2.1. SVM
% SVM Parameters
total_trial=100; % total number of SVM trials
model=templateSVM('KernelFunction','linear','KernelScale','auto',...
'Standardize',true,'BoxConstraint',25);

% Training and testing division
N_train=[51:200, 351:400, 451:600, 751:800, 851:1000, 1151:1200,...
         1251:1400, 1551:1600, 1651:1800, 1951:2000];
N_test=[201:350, 1:50, 601:750, 401:450, 1001:1150, 801:850,...
        1401:1550, 1201:1250, 1801:1950, 1601:1650];

%% SVM for data type X
[SVM_OUT_X, SVM_Targets_test,...
SVM_accuracy_X, avg_SVM_accuracy_X, max_SVM_accuracy_X]=...
SVMclassification(SVM_X, SVM_Targets, total_trial, N_train, N_test, model);

% Print SVM accuracy
fprintf ('\n Average SVM accuracy, data type X: %f \n', avg_SVM_accuracy_X)
fprintf ('\n Maximum SVM accuracy, data type X: %f \n', max_SVM_accuracy_X)

% Plot confusion matrix
for num_trial=1:total_trial
figure
Testconfusion = confusionchart(SVM_Targets_test,SVM_OUT_X(num_trial,:));
Testconfusion.Title = 'X Confusion Matrix';
end

%% SVM for data type F
[SVM_OUT_F, SVM_Targets_test,...
SVM_accuracy_F, avg_SVM_accuracy_F, max_SVM_accuracy_F]=...
SVMclassification(SVM_F, SVM_Targets, total_trial, N_train, N_test, model);

% Print SVM accuracy
fprintf ('\n Average SVM accuracy, data type F: %f \n', avg_SVM_accuracy_F)
fprintf ('\n Maximum SVM accuracy, data type F: %f \n', max_SVM_accuracy_F)

% Plot confusion matrix
for num_trial=1:total_trial
figure
Testconfusion = confusionchart(SVM_Targets_test,SVM_OUT_F(num_trial,:));
```

```
Testconfusion.Title = 'F Confusion Matrix';
end

%% SVM for data type MAR
[SVM_OUT_MAR, SVM_Targets_test,...
SVM_accuracy_MAR ,avg_SVM_accuracy_MAR, max_SVM_accuracy_MAR]=...
SVMclassification(SVM_MAR, SVM_Targets, total_trial, N_train, N_test, model);

% Print SVM accuracy
fprintf ('\n Average SVM accuracy, data type MAR: %f \n',
avg_SVM_accuracy_MAR)
fprintf ('\n Maximum SVM accuracy, data type MAR: %f \n',
max_SVM_accuracy_MAR)

% Plot confusion matrix
for num_trial=1:total_trial
figure
Testconfusion = confusionchart(SVM_Targets_test,SVM_OUT_MAR(num_trial,:));
Testconfusion.Title = 'MAR Confusion Matrix';
end

%% SVM for data type MAH
[SVM_OUT_MAH, SVM_Targets_test,...
SVM_accuracy_MAH, avg_SVM_accuracy_MAH, max_SVM_accuracy_MAH]=...
SVMclassification(SVM_MAH, SVM_Targets, total_trial, N_train, N_test, model);

% Print SVM accuracy
fprintf ('\n Average SVM accuracy, data type MAH: %f \n',
avg_SVM_accuracy_MAH)
fprintf ('\n Maximum SVM accuracy, data type MAH: %f \n',
max_SVM_accuracy_MAH)

% Plot confusion matrix
for num_trial=1:total_trial
figure
Testconfusion = confusionchart(SVM_Targets_test,SVM_OUT_MAH(num_trial,:));
Testconfusion.Title = 'MAH Confusion Matrix';
end

%% SVM for data type LR
[SVM_OUT_LR, SVM_Targets_test,...
SVM_accuracy_LR ,avg_SVM_accuracy_LR, max_SVM_accuracy_LR]=...
SVMclassification(SVM_LR, SVM_Targets, total_trial, N_train, N_test, model);

% Print SVM accuracy
fprintf ('\n Average SVM accuracy, data type LR: %f \n', avg_SVM_accuracy_LR)
fprintf ('\n Maximum SVM accuracy, data type LR: %f \n', max_SVM_accuracy_LR)

% Plot confusion matrix
for num_trial=1:total_trial
figure
Testconfusion = confusionchart(SVM_Targets_test,SVM_OUT_LR(num_trial,:));
Testconfusion.Title = 'LR Confusion Matrix';
end
%% SVM for data type SG
[SVM_OUT_SG, SVM_Targets_test,...
```

```
SVM_accuracy_SG ,avg_SVM_accuracy_SG, max_SVM_accuracy_SG]=...
SVMclassification(SVM_SG, SVM_Targets, total_trial, N_train, N_test, model);

% Print SVM accuracy
fprintf ('\n Average SVM accuracy, data type SG: %f \n', avg_SVM_accuracy_SG)
fprintf ('\n Maximum SVM accuracy, data type SG: %f \n', max_SVM_accuracy_SG)

% Plot confusion matrix
for num_trial=1:total_trial
figure
Testconfusion = confusionchart(SVM_Targets_test,SVM_OUT_SG(num_trial,:));
Testconfusion.Title = 'SG Confusion Matrix';
end

%% 2.2. ANN
% ANN parameters
total_trial=2;      % total number of ANN trials
hiddenLayerSize=40;   % number of hidden neurons
trainFcn='trainscg';  % training function
net=patternnet(hiddenLayerSize, trainFcn); % ANN model
% input/output processing functions
net.input.processFcns={'removeconstantrows','mapminmax'};

% Training, validation, and testing division
net.divideFcn='divideind';  % divide data by index
net.divideMode='sample';    % divide samples
N_train=[51:140, 351:400, 451:540, 751:800, 851:940,...
    1151:1200, 1251:1340, 1551:1600, 1651:1740, 1951:2000];
N_valid=[141:200, 541:600, 941:1000, 1341:1400, 1741:1800];
N_test=[201:350, 1:50, 601:750, 401:450, 1001:1150,...
    801:850, 1401:1550, 1201:1250, 1801:1950, 1601:1650];

%% ANN for data type X
[ANN_OUT_X,ANN_Targets_test,...
ANN_accuracy_X,avg_ANN_accuracy_X, max_ANN_accuracy_X]=...
ANNclassification(ANN_X,ANN_Targets,total_trial,N_train, N_valid, N_test,
net);

% Print ANN accuracy
fprintf ('\n Average ANN accuracy, data type X: %f \n', avg_ANN_accuracy_X)
fprintf ('\n Maximum ANN accuracy, data type X: %f \n', max_ANN_accuracy_X)

% Plot confusion matrix
for num_trial=1:total_trial
figure
Testconfusion = confusionchart(ANN_Targets_test,ANN_OUT_X(num_trial,:));
Testconfusion.Title = 'X Confusion Matrix';
end

%% ANN for data type F
[ANN_OUT_F,ANN_Targets_test,...
ANN_accuracy_F,avg_ANN_accuracy_F, max_ANN_accuracy_F]=...
ANNclassification(ANN_F,ANN_Targets,total_trial,N_train, N_valid, N_test,
net);
```

```
% Print ANN accuracy
fprintf ('\n Average ANN accuracy, data type F: %f \n', avg_ANN_accuracy_F)
fprintf ('\n Maximum ANN accuracy, data type F: %f \n', max_ANN_accuracy_F)

% Plot confusion matrix
for num_trial=1:total_trial
figure
Testconfusion = confusionchart(ANN_Targets_test,ANN_OUT_F(num_trial,:));
Testconfusion.Title = 'F Confusion Matrix';
end

%% ANN for data type MAR
[ANN_OUT_MAR,ANN_Targets_test,...
ANN_accuracy_MAR,avg_ANN_accuracy_MAR, max_ANN_accuracy_MAR]=...
ANNclassification(ANN_MAR,ANN_Targets,total_trial,N_train, N_valid, N_test,
net);

% Print ANN accuracy
fprintf ('\n Average ANN accuracy, data type MAR: %f \n',
avg_ANN_accuracy_MAR)
fprintf ('\n Maximum ANN accuracy, data type MAR: %f \n',
max_ANN_accuracy_MAR)

% Plot confusion matrix
for num_trial=1:total_trial
figure
Testconfusion = confusionchart(ANN_Targets_test,ANN_OUT_MAR(num_trial,:));
Testconfusion.Title = 'MAR Confusion Matrix';
end

%% ANN for data type MAH
[ANN_OUT_MAH,ANN_Targets_test,...
ANN_accuracy_MAH,avg_ANN_accuracy_MAH, max_ANN_accuracy_MAH]=...
ANNclassification(ANN_MAH,ANN_Targets,total_trial,N_train, N_valid, N_test,
net);

% Print ANN accuracy
fprintf ('\n Average ANN accuracy, data type MAH: %f \n',
avg_ANN_accuracy_MAH)
fprintf ('\n Maximum ANN accuracy, data type MAH: %f \n',
max_ANN_accuracy_MAH)

% Plot confusion matrix
for num_trial=1:total_trial
figure
Testconfusion = confusionchart(ANN_Targets_test,ANN_OUT_MAH(num_trial,:));
Testconfusion.Title = 'MAH Confusion Matrix';
end

%% ANN for data type LR
[ANN_OUT_LR,ANN_Targets_test,...
ANN_accuracy_LR,avg_ANN_accuracy_LR, max_ANN_accuracy_LR]=...
ANNclassification(ANN_LR,ANN_Targets,total_trial,N_train, N_valid, N_test,
net);
```

```
% Print ANN accuracy
fprintf ('\n Average ANN accuracy, data type LR: %f \n', avg_ANN_accuracy_LR)
fprintf ('\n Maximum ANN accuracy, data type LR: %f \n', max_ANN_accuracy_LR)

% Plot confusion matrix
for num_trial=1:total_trial
figure
Testconfusion = confusionchart(ANN_Targets_test,ANN_OUT_LR(num_trial,:));
Testconfusion.Title = 'LR Confusion Matrix';
end

%% ANN for data type SG
[ANN_OUT_SG,ANN_Targets_test,...
ANN_accuracy_SG,avg_ANN_accuracy_SG, max_ANN_accuracy_SG]=...
ANNclassification(ANN_SG,ANN_Targets,total_trial,N_train, N_valid, N_test,
net);

% Print ANN accuracy
fprintf ('\n Average ANN accuracy, data type SG: %f \n', avg_ANN_accuracy_SG)
fprintf ('\n Maximum ANN accuracy, data type SG: %f \n', max_ANN_accuracy_SG)

% Plot confusion matrix
for num_trial=1:total_trial
figure
Testconfusion = confusionchart(ANN_Targets_test,ANN_OUT_SG(num_trial,:));
Testconfusion.Title = 'SG Confusion Matrix';
end

%% STAGE 3. HIDDEN MARKOV MODEL-BASED OUTPUT CORRECTION
%% 3.1. SVM ouput correction
total_trial=100;
%State sequence
SVM_state_seq=flip(SVM_Targets_test.');

%% SVM output correction for data type MAR
% Observation sequence
SVM_obs_seq=flip(SVM_OUT_MAR,2);

[SVM_OUT_MAR_corrected,SVM_accuracy_MAR_corrected,avg_SVM_accuracy_MAR_correc
ted,...
max_SVM_accuracy_MAR_corrected]=HHM_SVM_corr(SVM_obs_seq,SVM_state_seq,total_
trial);

% Print corrected accuracy
fprintf ('\n Average corrected SVM accuracy, data type MAR: %f \n',
avg_SVM_accuracy_MAR_corrected);
fprintf ('\n Maximum corrected SVM accuracy, data type MAR: %f \n',
max_SVM_accuracy_MAR_corrected);

% Plot original and corrected accuracy
figure
plot(1:total_trial,SVM_accuracy_MAR,'k',1:total_trial,SVM_accuracy_MAR_correc
ted,'m')
ylim([0.8 1])
title('MAR Average Accuracy');
xlabel('Trial');
```

```
ylabel('Accuracy')
legend({'Original','Corrected'},'Location','southeast')

% Plot corrected confusion matrix
for num_trial=1:total_trial
figure
Testconfusion =
confusionchart(SVM_state_seq,SVM_OUT_MAR_corrected(num_trial,:));
Testconfusion.Title = 'Corrected MAR Confusion Matrix';
end

%% SVM output correction for data type MAH
% Observation sequence
SVM_obs_seq=flip(SVM_OUT_MAH,2);

[SVM_OUT_MAH_corrected,SVM_accuracy_MAH_corrected,avg_SVM_accuracy_MAH_correc
ted,...
max_SVM_accuracy_MAH_corrected]=HHM_SVM_corr(SVM_obs_seq,SVM_state_seq,total_
trial);

% Print corrected accuracy
fprintf ('\n Average corrected SVM accuracy, data type MAH: %f \n',
avg_SVM_accuracy_MAH_corrected);
fprintf ('\n Maximum corrected SVM accuracy, data type MAH: %f \n',
max_SVM_accuracy_MAH_corrected);

% Plot original and corrected accuracy
figure
plot(1:total_trial,SVM_accuracy_MAH,'k',1:total_trial,SVM_accuracy_MAH_correc
ted,'m')
ylim([0.8 1])
title('MAH Average Accuracy');
xlabel('Trial');
ylabel('Accuracy')
legend({'Original','Corrected'},'Location','southeast')

% Plot corrected confusion matrix
for num_trial=1:total_trial
figure
Testconfusion =
confusionchart(SVM_state_seq,SVM_OUT_MAH_corrected(num_trial,:));
Testconfusion.Title = 'Corrected MAH Confusion Matrix';
end

%% SVM output correction for data type LR
% Observation sequence
SVM_obs_seq=flip(SVM_OUT_LR,2);

[SVM_OUT_LR_corrected,SVM_accuracy_LR_corrected,avg_SVM_accuracy_LR_corrected
,...
max_SVM_accuracy_LR_corrected]=HHM_SVM_corr(SVM_obs_seq,SVM_state_seq,total_t
rial);

% Print corrected accuracy
fprintf ('\n Average corrected SVM accuracy, data type LR: %f \n',
avg_SVM_accuracy_LR_corrected);
```

```
fprintf ('\n Maximum corrected SVM accuracy, data type LR: %f \n',
max_SVM_accuracy_LR_corrected);

% Plot original and corrected accuracy
figure
plot(1:total_trial,SVM_accuracy_LR,'k',1:total_trial,SVM_accuracy_LR_correcte
d,'m')
ylim([0.8 1])
title('LR Average Accuracy');
xlabel('Trial');
ylabel('Accuracy')
legend({'Original','Corrected'},'Location','southeast')

% Plot corrected confusion matrix
for num_trial=1:total_trial
figure
Testconfusion =
confusionchart(SVM_state_seq,SVM_OUT_LR_corrected(num_trial,:));
Testconfusion.Title = 'Corrected LR Confusion Matrix';
end

%% SVM output correction for data type SG
% Observation sequence
SVM_obs_seq=flip(SVM_OUT_SG,2);

[SVM_OUT_SG_corrected,SVM_accuracy_SG_corrected,avg_SVM_accuracy_SG_corrected
,...
max_SVM_accuracy_SG_corrected]=HHM_SVM_corr(SVM_obs_seq,SVM_state_seq,total_t
rial);

% Print corrected accuracy
fprintf ('\n Average corrected SVM accuracy, data type SG: %f \n',
avg_SVM_accuracy_SG_corrected);
fprintf ('\n Maximum corrected SVM accuracy, data type SG: %f \n',
max_SVM_accuracy_SG_corrected);

% Plot original and corrected accuracy
figure
plot(1:total_trial,SVM_accuracy_SG,'k',1:total_trial,SVM_accuracy_SG_correcte
d,'m')
ylim([0.8 1])
title('SG Average Accuracy');
xlabel('Trial');
ylabel('Accuracy')
legend({'Original','Corrected'},'Location','southeast')

% Plot corrected confusion matrix
for num_trial=1:total_trial
figure
Testconfusion =
confusionchart(SVM_state_seq,SVM_OUT_SG_corrected(num_trial,:));
Testconfusion.Title = 'Corrected SG Confusion Matrix';
end

%% 3.2. ANN ouput correction
total_trial=100;
%State sequence
```

```
ANN_state_seq=flip(ANN_Targets_test);

%% ANN output correction for data type MAR
% Observation sequence
ANN_obs_seq=flip(ANN_OUT_MAR,2);

[ANN_OUT_MAR_corrected,ANN_accuracy_MAR_corrected,avg_ANN_accuracy_MAR_correc
ted,...
max_ANN_accuracy_MAR_corrected]=HHM_ANN_corr(ANN_obs_seq,ANN_state_seq,total_
trial);

% Print corrected accuracy
fprintf ('\n Average corrected ANN accuracy, data type MAR: %f \n',
avg_ANN_accuracy_MAR_corrected);
fprintf ('\n Maximum corrected ANN accuracy, data type MAR: %f \n',
max_ANN_accuracy_MAR_corrected);

% Plot original and corrected accuracy
figure
plot(1:total_trial,ANN_accuracy_MAR,'k',1:total_trial,ANN_accuracy_MAR_correc
ted,'m')
ylim([0.8 1])
title('MAR Average Accuracy');
xlabel('Trial');
ylabel('Accuracy')
legend({'Original','Corrected'},'Location','southeast')

% Plot corrected confusion matrix
for num_trial=1:total_trial
figure
Testconfusion =
confusionchart(ANN_state_seq,ANN_OUT_MAR_corrected(num_trial,:));
Testconfusion.Title = 'Corrected MAR Confusion Matrix';
end

%% ANN output correction for data type MAH
% Observation sequence
ANN_obs_seq=flip(ANN_OUT_MAH,2);

[ANN_OUT_MAH_corrected,ANN_accuracy_MAH_corrected,avg_ANN_accuracy_MAH_correc
ted,...
max_ANN_accuracy_MAH_corrected]=HHM_ANN_corr(ANN_obs_seq,ANN_state_seq,total_
trial);

% Print corrected accuracy
fprintf ('\n Average corrected ANN accuracy, data type MAH: %f \n',
avg_ANN_accuracy_MAH_corrected);
fprintf ('\n Maximum corrected ANN accuracy, data type MAH: %f \n',
max_ANN_accuracy_MAH_corrected);

% Plot original and corrected accuracy
figure
plot(1:total_trial,ANN_accuracy_MAH,'k',1:total_trial,ANN_accuracy_MAH_correc
ted,'m')
ylim([0.8 1])
title('MAH Average Accuracy');
```

```
xlabel('Trial');
ylabel('Accuracy')
legend({'Original','Corrected'},'Location','southeast')

% Plot corrected confusion matrix
for num_trial=1:total_trial
figure
Testconfusion =
confusionchart(ANN_state_seq,ANN_OUT_MAH_corrected(num_trial,:));
Testconfusion.Title = 'Corrected MAH Confusion Matrix';
end

%% ANN output correction for data type LR
% Observation sequence
ANN_obs_seq=flip(ANN_OUT_LR,2);

[ANN_OUT_LR_corrected,ANN_accuracy_LR_corrected,avg_ANN_accuracy_LR_corrected
,...
max_ANN_accuracy_LR_corrected]=HHM_ANN_corr(ANN_obs_seq,ANN_state_seq,total_t
rial);

% Print corrected accuracy
fprintf ('\n Average corrected ANN accuracy, data type LR: %f \n',
avg_ANN_accuracy_LR_corrected);
fprintf ('\n Maximum corrected ANN accuracy, data type LR: %f \n',
max_ANN_accuracy_LR_corrected);

% Plot original and corrected accuracy
figure
plot(1:total_trial,ANN_accuracy_LR,'k',1:total_trial,ANN_accuracy_LR_correcte
d,'m')
ylim([0.8 1])
title('LR Average Accuracy');
xlabel('Trial');
ylabel('Accuracy')
legend({'Original','Corrected'},'Location','southeast')

% Plot corrected confusion matrix
for num_trial=1:total_trial
figure
Testconfusion =
confusionchart(ANN_state_seq,ANN_OUT_LR_corrected(num_trial,:));
Testconfusion.Title = 'Corrected LR Confusion Matrix';
end

%% ANN output correction for data type SG
% Observation sequence
ANN_obs_seq=flip(ANN_OUT_SG,2);

[ANN_OUT_SG_corrected,ANN_accuracy_SG_corrected,avg_ANN_accuracy_SG_corrected
,...
max_ANN_accuracy_SG_corrected]=HHM_ANN_corr(ANN_obs_seq,ANN_state_seq,total_t
rial);
```

```matlab
% Print corrected accuracy
fprintf ('\n Average corrected ANN accuracy, data type SG: %f \n',
avg_ANN_accuracy_SG_corrected);
fprintf ('\n Maximum corrected ANN accuracy, data type SG: %f \n',
max_ANN_accuracy_SG_corrected);

% Plot original and corrected accuracy
figure
plot(1:total_trial,ANN_accuracy_SG,'k',1:total_trial,ANN_accuracy_SG_correcte
d,'m')
ylim([0.8 1])
title('SG Average Accuracy');
xlabel('Trial');
ylabel('Accuracy')
legend({'Original','Corrected'},'Location','southeast')

% Plot corrected confusion matrix
for num_trial=1:total_trial
figure
Testconfusion =
confusionchart(ANN_state_seq,ANN_OUT_SG_corrected(num_trial,:));
Testconfusion.Title = 'Corrected SG Confusion Matrix';
end

%% FUNCTIONS

%% Read CT data
function [M] = readCS(numberFile,localDirPath)
if numberFile(1)>=0 &&  numberFile(1) < 10
    folder = ([localDirPath ]);
    baseFileName =([ 'tek000' num2str( numberFile,'%d') 'CH1.csv']);
elseif numberFile(1)>=10 && numberFile(1) < 100
    folder = ([localDirPath ]);
    baseFileName =([ 'tek00' int2str(numberFile) 'CH1.csv']);
elseif numberFile(1)>=100 && numberFile(1) < 1000
    folder = ([localDirPath ]);
    baseFileName =([ 'tek0' int2str(numberFile) 'CH1.csv']);
else
folder = ([localDirPath ]);
    baseFileName =([ 'tek' int2str(numberFile) 'CH1.csv']);
end
fullFileName = fullfile(folder, baseFileName);
if exist(fullFileName,'file')
  M = csvread(fullFileName,21,0);
else
  warningMessage = sprintf('%s does not exist', fullFileName);
  uiwait(warndlg(warningMessage));
end
end

%% Splitting - Dataset XS
function Dout=Dsplit(Din,xs,sSplit,split)
Dout=zeros(xs,sSplit);
for numberFile=1:2
NF=numberFile-1;
    [M] = readCS(NF,Din);
```

```
    X1 = M(:,2);
    for i=1:split
    Dout(:,i+NF*split) = X1((i-1)*xs+1:i*xs);
    end
end
end

%% Dataset XT
function Dout=Dread(Din,Xlength,s)
Dout=zeros(Xlength,s);
for numberFile = 2:201
    [M] = readCS(numberFile,Din);
    X = M(:,2);
    Dout(:,numberFile-1) = X;
end
end

%% Perform FFT
function [f1,OUT]=Spec(IN,FS)
f1=round(0:FS/length(IN):FS/2);
xdft1=20*log10(abs(fft(IN)/length(IN))/1E-6)+120;
OUT=xdft1(1:length(IN)/2+1);
OUT=OUT(1:50000,1);
f1=f1(1,1:50000);
end

%% Spectral smoothing
function [MAR,MAH,LR,SG]=SpecSm(Y,w,SL)
MAR = smoothdata(Y,'movmean',w);
LR = smoothdata(Y,'lowess',w);
SG = smoothdata(Y,'sgolay',w);
wn = hann(w);
A=zeros(SL+w,1);
A(w/2+1:SL+w/2,1)=Y;
A(1:w/2)=Y(1);
A(SL+w/2+1:SL+w)=Y(SL);
MAH = conv(A, wn, 'valid')./5e2;
MAH=MAH(1:SL);
end

%% ANN matrix of features
function Dout=ANNfeatures(D1,D2,D3,D4,D5,s)
Dout(:,s*0+1:s*1)=D1;
Dout(:,s*1+1:s*2)=D2;
Dout(:,s*2+1:s*3)=D3;
Dout(:,s*3+1:s*4)=D4;
Dout(:,s*4+1:s*5)=D5;
end

%% SVM classification
function [SVM_OUT,SVM_Targets_test,SVM_accuracy,avg_SVM_accuracy,
max_SVM_accuracy]=SVMclassification(SVM_Features,SVM_Targets,total_trial,N_tr
ain, N_test,model)
SVM_Features_train=SVM_Features(N_train,:);
SVM_Features_test=SVM_Features(N_test,:);
SVM_Targets_train=SVM_Targets(N_train,:);
SVM_Targets_test=SVM_Targets(N_test,:);
```

```
for num_trial=1:total_trial
% SVM Training
SVMmodel=fitcecoc(SVM_Features_train,SVM_Targets_train,'Learners',model);
% SVM Validation
CVmodel=crossval(SVMmodel,'Holdout',0.3);
% SVM Testing
TrainedModel=CVmodel.Trained{1};
[Labels_predict,Score_predict]=predict(TrainedModel,SVM_Features_test);
% SVM output
SVM_OUT(num_trial,:)=Labels_predict.';
% SVM accuracy
SVM_accuracy(num_trial)=sum(Labels_predict==SVM_Targets_test)/length(Labels_p
redict);
end
% Average SVM accuracy
avg_SVM_accuracy=mean(SVM_accuracy);
% Maximum SVM accuracy
max_SVM_accuracy=max(SVM_accuracy);
end

%% ANN classification
function [ANN_OUT,ANN_Targets_test,ANN_accuracy,avg_ANN_accuracy,
max_ANN_accuracy]=ANNclassification(ANN_Features,ANN_Targets,total_trial,N_tr
ain,N_valid, N_test, net)
net.divideParam.trainInd=N_train;
net.divideParam.valInd=N_valid;
net.divideParam.testInd=N_test;
ANN_Targets_test=vec2ind(ANN_Targets(:,N_test));
for num_trial=1:total_trial
% ANN training
[ANNmodel,tr] = train(net,ANN_Features,ANN_Targets);
% ANN testing
y = ANNmodel(ANN_Features);
e = gsubtract(ANN_Targets,y);
yind = vec2ind(y);
OUT=yind(:,net.divideParam.testInd);
% ANN output
ANN_OUT(num_trial,:)=OUT;
% ANN accuracy
ANN_accuracy(num_trial)=sum(OUT==ANN_Targets_test)/length(ANN_OUT);
end
% Average ANN accuracy
avg_ANN_accuracy=mean(ANN_accuracy);
% Maximum ANN accuracy
max_ANN_accuracy=max(ANN_accuracy);
end

%% HMM-based SVM output correction
function
[SVM_OUT_corr,SVM_accuracy_corr,avg_SVM_accuracy_corr,max_SVM_accuracy_corr]=
HHM_SVM_corr(SVM_obs_seq,SVM_state_seq,total_trial)
for num_trial=1:total_trial

observed=SVM_obs_seq(num_trial,:);
corrected(1)=observed(1);
    tbd=[];
    for k=2:length(observed)
        temp=observed(k);
```

```
        if isempty(tbd)
            if temp>observed(k-1)
                corrected(k)=observed(k-1);
            elseif temp==observed(k-1)
                corrected(k-1)=temp;
                corrected(k)=temp;
            else
                if k<length(observed)
                tbd=temp;
                else
                    corrected(k)=corrected(k-1);
                end
            end
        else
            if temp>tbd
                corrected(k-1:k)=temp;
                tbd=[];
            elseif temp==tbd
                corrected(k-1:k)=tbd;
                tbd=[];
            else
                if k<length(observed)
                    corrected(k-1:k)=tbd;
                    tbd=temp;
                else
                    corrected(k-1:k)=tbd;
                    tbd=[];
                end
            end
        end
    end
% Corrected output
SVM_OUT_corr(num_trial,:)=corrected;
% Corrected SVM accuracy
SVM_accuracy_corr(num_trial)=sum(corrected==SVM_state_seq(1:length(SVM_OUT_co
rr)))/length(SVM_OUT_corr);
end
% Average corrected SVM accuracy
avg_SVM_accuracy_corr=mean(SVM_accuracy_corr);
% Maximum corrected SVM accuracy
max_SVM_accuracy_corr=max(SVM_accuracy_corr);
end

%% HMM-based ANN output correction
function
[ANN_OUT_corr,ANN_accuracy_corr,avg_ANN_accuracy_corr,max_ANN_accuracy_corr]=
HHM_ANN_corr(ANN_obs_seq,ANN_state_seq,total_trial)
for num_trial=1:total_trial

observed=ANN_obs_seq(num_trial,:);
corrected(1)=observed(1);
    tbd=[];
    for k=2:length(observed)
        temp=observed(k);
        if isempty(tbd)
            if temp>observed(k-1)
                corrected(k)=observed(k-1);
            elseif temp==observed(k-1)
```

```
                corrected(k-1)=temp;
                corrected(k)=temp;
            else
                if k<length(observed)
                tbd=temp;
                else
                    corrected(k)=corrected(k-1);
                end
            end
        else
            if temp>tbd
                if temp-tbd==1
               corrected(k-1:k)=temp;
                else
                    corrected(k-1:k)=temp-1;
                end
                tbd=[];
            elseif temp==tbd
                corrected(k-1:k)=tbd;
                tbd=[];
            else
                if k<length(observed)
                    corrected(k-1:k)=tbd;
                    tbd=temp;
                else
                    corrected(k-1:k)=tbd;
                    tbd=[];
                end
            end
        end
    end
% Corrected output
ANN_OUT_corr(num_trial,:)=corrected;
% Corrected ANN accuracy
ANN_accuracy_corr(num_trial)=sum(corrected==ANN_state_seq(1:length(ANN_OUT_co
rr)))/length(ANN_OUT_corr);
end
% Average corrected ANN accuracy
avg_ANN_accuracy_corr=mean(ANN_accuracy_corr);
% Maximum corrected SVM accuracy
max_ANN_accuracy_corr=max(ANN_accuracy_corr);
end
```