DOCTORAL THESIS

IoT based Tools and Methods for Electrical Machine Diagnostics

Hadi Ashraf Raja

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY
TALLINN 2023

TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
20/2023

IoT based Tools and Methods for Electrical Machine Diagnostics

HADI ASHRAF RAJA

TAL TECH PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
School of Engineering
Department of Electrical Power Engineering and Mechatronics
This dissertation was accepted for the defence of the degree on 12/05/2023

Supervisor: Dr. Toomas Vaimann
Dept. of Electrical Power Engineering and Mechatronics
School of Engineering
Tallinn University of Technology
Estonia

Co-Supervisor: Dr. Bilal Asad
Dept. of Electrical Power Engineering and Mechatronics
School of Engineering
Tallinn University of Technology
Estonia

Opponents: Doc. Ing. Roman Pechánek, Ph.D.
Vice-Dean for Educational Activity
University of West Bohemia
Czech Republic

Dr. Floran Martin
Staff Scientist
Aalto University
Finland

Defence of the thesis: 19/06/2023, Tallinn

Declaration:
Hereby I declare that this doctoral thesis, my original investigation, and achievement, submitted for the doctoral degree at Tallinn University of Technology has not been submitted for doctoral or equivalent academic degree.

Hadi Ashraf Raja

----
signature

European Union
European Regional
Development Fund

Investing
in your future

ISSN 2585-6898 (publication)
ISBN 978-9949-83-981-0 (publication)
ISSN 2585-6901 (PDF)
ISBN 978-9949-83-982-7 (PDF)
Printed by Koopia Niini & Rauam

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
20/2023

Asjade interneti põhised tööriistad ja meetodid elektrimasinate diagnostikaks

HADI ASHRAF RAJA

TALTECH
KIRJASTUS

Contents

List of Publications

The list of author's publications, on the basis of which the thesis has been prepared:

I **Raja, H. A**.; Kudelina, K.; Asad, B.; Vaimann, T.; Kallaste, A.; Rassõlkin, A.; Khang, H. V. (2022). Signal Spectrum-Based Machine Learning Approach for Fault Prediction and Maintenance of Electrical Machines. Energies, 15 (24), #9507. DOI: 10.3390/en15249507.

II Asad, B.; **Raja, H. A.**; Vaimann, T.; Kallaste, A.; Pomarnacki, R.; Hyunh, V.K. A Current Spectrum-Based Algorithm for Fault Detection of Electrical Machines Using Low-Power Data Acquisition Devices. Electronics 2023, 12, 1746. DOI: 10.3390/electronics12071746

III **Raja, H**. **A.**; Raval, H.; Vaimann, T.; Rassõlkin, A.; Kallaste, A.; Belahcen A. (2022). Cost-efficient real-time condition monitoring and fault diagnostics system for BLDC motor using IoT and Machine learning. International Conference on Diagnostics in Electrical Engineering (Diagnostika), 6. - 8. 9. 20226. - 8. 9. 2022, Pilsen, Czech Republic. IEEE.

IV **Raja, H. A.**; Asad, B.; Vaimann, T.; Rassõlkin, A.; Kallaste, A.; Belahcen A. (2022). Custom Simplified Machine Learning Algorithms for Fault Diagnosis in Electrical Machines. International Conference on Diagnostics in Electrical Engineering (Diagnostika), 6. - 8. 9. 2022, Pilsen, Czech Republic. IEEE.

V **Raja, H. A.**; Vaimann, T.; Rassõlkin, A.; Kallaste, A. (2022). Condition Monitoring and Fault Detection for Electrical Machines using IOT. Proceedings of the Future Technologies Conference (FTC) 2022, Volume 2. Lecture Notes in Networks and Systems, vol 560. Springer, Cham. https://doi.org/10.1007/978-3-031-18458-1_12.

VI **Raja, H. A.**; Vaimann, T.; Rassõlkin, A.; Kallaste, A.; Belahcen, A. (2021). IoT Based Tools for Data Acquisition in Electrical Machines and Robotics. 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC). Poland: IEEE, 737−742. DOI: 10.1109/PEMC48073.2021.9432553.

VII **Raja, H. A.**; Kudelina, K.; Asad, B.; Vaimann, T. (2022). Fault Detection and Predictive Maintenance of Electrical Machines. In: New Trends in Electric Machines - Technology and Applications IntechOpen. DOI: 10.5772/intechopen.107167.

Other Publications:

VIII Kudelina, K.; **Raja, H. A.**; Autsou, S.; Asad, B.; Vaimann, T.; Rassõlkin, A.; Kallaste, A. (2022). Preliminary Analysis of Global Parameters of Induction Machine for Fault Prediction in Rotor Bars. International Power Electronics and Motion Control Conference (PEMC): International Power Electronics and Motion Control Conference (PEMC), Brasov, Romania., 25-28 September 2022. IEEE.

IX Kudelina, K.; **Raja, H. A.**; Autsou, S.; Asad, B.; Vaimann, T.; Rassõlkin, A.; Kallaste, A.; Shabbir, N. (2022). The Impact of Load on Global Parameters of Electrical Machines in Case of Healthy and Broken Rotor Bars. Baltic Electronics Conference (BEC): Baltic Electronics Conference 2022, Tallinn, Estonia, October 4-6, 2022. IEEE.

X Kudelina, K.; **Raja, H. A.**; Autsou, S.; Asad, B.; Vaimann, T.; Rassõlkin, A.; Kallaste, A. (2022). The Impact of Control Environments on Global Parameters of Electrical Machines in Case of Broken Rotor Bars. In: International Conference on Diagnostics in Electrical Engineering (Diagnostika) IEEE.

XI Shabbir, N.; Kütt, L.; **Raja, H. A.**; Jawad, M.; Allik, A.; Husev, O. (2022). Techno-Economic Analysis and Energy Forecasting Study of Domestic and Commercial Photovoltaic System Installations in Estonia. Energy, 253, 124156. DOI: 10.1016/j.energy.2022.124156.

XII Shabbir, N.; Kütt, L; Daniel, K; Astapov, V; **Raja, H. A.**; Iqbal, M. N; Husev, O;(2022). Feasibility Investigation of Residential Battery Sizing Considering EV Charging Demand. Sustainability, 14 (3), #1079. DOI: 10.3390/su14031079.

XIII Shabbir, N.; Kütt, L.; **Raja, H. A.**; Ahmadiahangar, R.; Rosin, A.; Husev, O.; (2021). Machine Learning and Deep Learning Techniques for Residential Load Forecasting: A Comparative Analysis. 2021 IEEE 62nd International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON): 2021 IEEE 62nd International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON). Riga, Latvia: IEEE. DOI: 10.1109/RTUCON53541.2021.9711741.

XIV Shabbir, N.; Ahmadiahangar, R.; **Raja, H. A.**; Kütt, L.; Rosin, A. (2020). Residential Load Forecasting using Recurrent Neural Networks. IEEE 14th International Conference on Compatibility, Power Electronics and Power Engineering (CPE - POWERENG 2020). Portugal: IEEE.

Author’s Contribution to the Publications

Contribution to the papers in this thesis are:

I Hadi Ashraf Raja is the primary author of this article. He designed and implemented the signal spectrum based technique and validated the results. He also wrote the initial draft of the paper.

II Hadi Ashraf Raja is the secondary author of this article. He helped with the implementation of the experimental setup in the laboratory for network losses and also helped in writing the manuscript.

III Hadi Ashraf Raja is the primary author of the article. He designed the experiment and helped in gathering the data in the laboratory and also wrote the initial draft of the paper.

IV Hadi Ashraf Raja is the primary author of the article. He implemented a custom machine learning algorithm with different activation functions and carried out a comparative analysis. He also wrote the initial draft of the article.

V Hadi Ashraf Raja is the primary author of the article. He implemented the experimental setup in the laboratory, gathered data and also connected it to the web portal. He also wrote the initial draft of the article.

VI Hadi Ashraf Raja is the primary author of the article. He performed laboratory experiments, gathered data and also wrote the initial draft of the article.

VII Hadi Ashraf Raja is the primary author of the article. He performed laboratory experiments and ran simulations. He also wrote the initial draft of the chapter.

Abbreviations

6LoWPAN	Internet protocol version 6 (IPv6) over low-power wireless networks
AC	Alternating current
ADC	Analog-to-digital converter
AES	Advanced encryption standard
AI	Artificial intelligence
ANN	Artificial neural network
ARM	Advanced risc machine
BLE	Bluetooth low energy
BRB	Broken rotor bars
CAN	Controller area network
CNN	Convolutional neural networks
CoAP	Constrained application protocol
CPA	Principal component analysis
DBN	Deep belief networks
DC	Direct current
DTLS	Datagram transport layer security
ESP32	Espressif 32 bit module
FEM	Finite element method
FFT	Fast Fourier transform
GAN	Generative adversarial network
GMM	Gaussian mixture models
GPIO	General purpose input/output
GUI	Graphic user interface
HDMI	High-definition multimedia interface
I/O Pins	Input/output pins
I2C	Inter-integrated circuit
IEEE	Institute of electrical and electronics engineers
INOC	Integral number of cycles
IoT	Internet of things
IP	Internet protocol address
KNN	K-nearest neighbors
LAN	Local area network
LoRaWAN	Low powered, wide area networking protocol
LSTM	Long short-term memory
LTE-M	Long-term evolution
MCUs	Microcontroller units
ML	Machine Learning
MQTT	Message queuing telemetry transport

MSE	Mean square error
NB-IoT	Narrowband Internet of Things
NFC	Near-field communication
PCA	Principal component analysis
PLC	Programmable logic controller
RFID	Radio frequency identification
RMSE	Root mean square error
RNN	Recurrent neural network
SCADA	Supervisory control and data acquisition
SD Card	Secure digital card
SoC	System-on-a-chip
SPI	Serial peripheral interface
SRAM	Static random access memory
SSL	Secure sockets layer
SVM	Support vector machines
SVR	Support vector regression
TCP	Transmission control protocol
TLS	Transport layer security
UART	Universal asynchronous receiver/transmitter
USB	Universal serial bus
Wi-Fi	Wireless fidelity
WPA2	Wireless fidelity protected access 2
XMPP	Extensible messaging and presence protocol
XGBoost	Extreme Gradient Boosting

Symbols

b_d	Bearing ball's diameter
f_{asym}	Rotor winding asymmetries frequencies
f_{bb}	Bearing fault frequencies
f_{BR1}	Broken bar frequencies
f_{ecce}	Eccentricity frequency
$f_{(i,o)}$	Characteristic vibration frequencies
f_p	Probability of fault occurrence
f_r	Rotor frequency
f_s	Supply frequency
k	Harmonic order
m	Positive integer
n_b	Number of rotor bars
n_{bb}	Number of bearing balls
n_d	Dynamic eccentricity (0 for static and 1,2,3…. for dynamic)
p	Number of poles
P	Number of pole pairs
p_d	Bearing pitch
s	Slip
v	Supply fed harmonics
x	Input data
y	Output data
y_f	Maximum amplitude of the frequency component at faulty state
y_h	Maximum amplitude of the frequency component at healthy state
y_t	Higher average amplitude of the frequency component
α	Learning rate
θ	Parameters of the machine learning model
$\mathrm{f}(\mathrm{x};\theta))$	Predicted output
$\mathrm{L}(\mathrm{y},\mathrm{f}(\mathrm{x};\theta))$	Loss function for machine learning model
$\mathrm{L}(\mathrm{W}(\mathrm{t}))$	Loss function for neural network
$\mathrm{W}(\mathrm{t})$	Weights and biases of the network at iteration t
$\nabla\mathrm{L}(\mathrm{W}(\mathrm{t}))$	Gradient of the loss function with respect to weights and biases of the neural network

1 Introduction

1.1 Electrical Machines

Electrical machines are one of the most crucial components of the current era. Their wide range of residential and industrial applications has made life simpler, safer and more convenient. With the introduction of Industry 4.0 [1], there is a further increase in the importance of electrical machines due to their reliability, speed range, efficiency, power density, low cost, etc. They are also one of the critical components of renewable energy resources, as in the case of wind turbines [2], which are essential for a sustainable future. There are several types of electrical machines with unique characteristics and applications. Some of the most commonly used electrical machines are listed below in Table 1.1.

Table 1.1. Common types of electrical machines

Type of Electrical Machine	Description	Applications
DC motors	DC motors convert electrical energy into mechanical energy through a rotating shaft. They can provide precise speed control and better efficiency.	Robotics, automation, electric vehicles, industrial machinery
AC motors	AC motors use alternating current to create a rotating magnetic field, which generates mechanical energy. As a result, they are highly efficient and reliable, with low maintenance requirements.	Industrial machinery, pumps, fans, compressors, HVAC systems
Transformers	Transformers are used to transfer electrical energy from one circuit to another without changing the frequency of the current.	Power generation and distribution, electrical equipment
Generators	Generators convert mechanical energy into electrical energy through a rotating shaft.	Power generation, backup power supplies
Alternators	Alternators are similar to generators but produce alternating current instead of direct current.	Automotive, industrial machinery
Stepper motors	Stepper motors provide precise control over position and speed, making them ideal for robotics and automation systems. They operate by moving in small, precise steps rather than rotating continuously.	Robotics, automation, industrial machinery
Servo motors	Servo motors provide precise control over position and speed. They are designed to provide smooth, accurate movement.	Robotics, automation, industrial machinery

Besides these standard electrical machines, specialized electrical machines are used for specific applications, such as linear motors, magnetic bearings, and piezoelectric motors. Therefore, for designing and optimizing systems for various industrial and commercial applications, it is necessary to understand the unique characteristics of each electrical machine. The importance of electrical machines in modern society cannot be overstated. They are essential components of daily life, powering various applications, from small appliances like fans and refrigerators to large-scale industrial machines like pumps, compressors, and turbines. Moreover, many systems and devices usually relied on for industrial and residential applications would be impossible without electrical machines.

The industrial applications of electrical machines range from power distribution to transportation, manufacturing, automation and many more because they are highly efficient and reliable. Electrical machines are designed to operate continuously and reliably over long periods, making them ideal for industrial applications where downtime can be costly and disruptive. Hence, the downtime of electrical machines in an industrial environment due to stress and load can result in economic losses, wastage of resources and even threats to human life in the worst case. Therefore, maintenance of these electrical machines is a big question, current practices of scheduled maintenance result in a lot of resources being used. That is why with the recent advancements in information technology, most of the industry is moving towards predictive maintenance.

1.2 Internet of Things

The Internet of Things (IoT) [3] is a rapidly growing technological paradigm transforming how we interact with the world around us. At its core, the IoT refers to the network of interconnected devices, sensors, and systems that collect and exchange data, often in real-time, without requiring direct human input. The concept of the IoT has been around for several decades, but it is only in recent years that it has gained widespread attention and adoption. Advances in wireless communication, data analytics, and cloud computing have created highly interconnected and intelligent systems that can respond to real-time data inputs and adapt to changing conditions.

The Internet of Things (IoT) is the communication of different intelligent devices over the internet. Technological advancement has not only made our life easier but has also paved new ways to be more efficient. The data from devices can be used to predict results, diagnose the machine and predict faults. Hence, cutting time on the maintenance of a single machine and making it cost-efficient to remove unnecessary maintenance checks. IoT has a lot of applications in the industrial area, including predictive maintenance of industrial equipment [1]. As manufacturing has been advancing, IoT applications related to industrial development and monitoring have been increasing rapidly too. The inclusion of data collection from machines to monitor them and run predictive analytics for maintenance [2] is becoming a norm in the industrial field.

IoT architecture [4–6] refers to the structure of an IoT system [7,8], including the various components and how they interact with each other to collect, process, and transmit data. There are several layers to an IoT architecture, each with its own set of functions and responsibilities. The most commonly used IoT architecture model is the 5-layer architecture shown in Figure 1.1, which includes the following layers:

Perception layer: The perception layer is the bottom layer of the IoT architecture and is responsible for sensing the physical world. It includes various sensors and devices that collect environmental data, such as temperature, humidity, and motion. The data collected at this layer is raw and unprocessed.

Network layer: The network layer transmits the data collected at the perception layer to the cloud or the edge computing layer. It includes various communication protocols such as Wi-Fi, Bluetooth, and Zigbee. In addition, the network layer ensures that the data is transmitted securely and efficiently.

Processing layer: The processing layer is responsible for processing the data collected at the perception layer. It includes various edge computing devices such as gateways, routers, and servers. In addition, the processing layer is responsible for performing tasks such as data filtering, aggregation, and analysis.

Application layer: The application layer is responsible for providing end-users with various applications that utilize the data collected by the IoT system. It includes various applications such as monitoring applications, control applications, and analytics applications. The application layer makes the data collected by the IoT system meaningful and valuable to end users.

Business layer: The business layer manages and controls the IoT system. It includes various management and control applications such as billing, security, and device management. In addition, the business layer ensures that the IoT system is secure, efficient, and cost-effective.

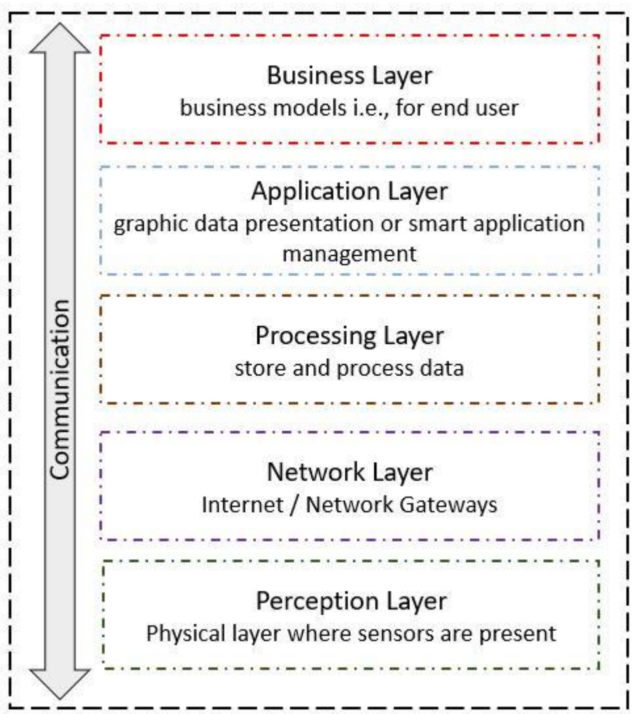


Figure 1.1. Architecture of IoT

The 5-layer IoT architecture provides a framework for building scalable and efficient IoT systems. However, other IoT architecture models have been proposed, such as the 4-layer architecture, which excludes the processing layer, and the 3-layer architecture, which excludes the perception layer. The choice of architecture model depends on the specific requirements of the IoT system and the available resources.

The IoT has enormous potential to transform various industries and applications, from healthcare [9,10] and agriculture to transportation and manufacturing. By enabling real-time data collection and analysis, IoT systems can help organizations optimize their operations, reduce costs, and improve the quality of products and services. One of the key benefits of the IoT is its ability to enable more efficient and sustainable use of resources. For example, intelligent energy systems that use IoT sensors and data analytics can help organizations optimize their energy usage, reduce waste, and minimize their carbon footprint [11].

In addition, the IoT is playing an increasingly important role in the development of smart cities [12,13], which use interconnected systems to improve the quality of life for residents and make urban environments more sustainable and efficient[14]. From smart transportation systems that use real-time data to optimize traffic flow and reduce congestion to intelligent buildings that adjust lighting and heating based on occupancy patterns, the IoT is enabling a new era of urban design and management.

Despite its many benefits, the IoT presents several challenges and risks, particularly regarding data privacy and security. With so many interconnected devices and systems exchanging data, sensitive information can fall into the wrong hands or be used for malicious purposes. As such, organizations must take proactive steps to ensure that their IoT systems are secure and that data privacy is maintained. Some of the most common IoT protocols are listed in Table 1.2.

MQTT: MQTT [15,16] is a lightweight messaging protocol widely used in IoT applications for telemetry and messaging. It uses TCP/IP for communication and can provide data transfer rates of up to 256 Kbps. In addition, it has low power consumption and supports SSL/TLS encryption for security.

XMPP: XMPP [17] is a custom protocol commonly used in industrial automation. It is designed for high-speed communication and can provide up to 1 Gbps data transfer rates. In addition, it has low power consumption and supports custom encryption methods for security.

CoAP: CoAP [18] is a lightweight protocol commonly used for IoT messaging and RESTful web services. It uses UDP for communication and can provide data transfer rates of up to 20 Kbps. In addition, it has low power consumption and supports DTLS encryption for security.

The choice of protocol for an IoT application depends on the specific requirements of the application, including data transfer rates, power consumption, and security needs [19]. Table 1.2 provides a brief comparison of some popular IoT protocols, but there are many other protocols available that may be better suited to specific applications.

Table 1.2. IoT protocols

Protocol Name	Application	Communication Protocol	Data Transfer Rate	Security	Power Consumption
Wi-Fi	Smart homes, offices	TCP/IP	Up to 866 Mbps	WPA2 encryption	High power consumption
Bluetooth	Wearables, smart homes	Bluetooth Low Energy (BLE)	Up to 1 Mbps	AES encryption	Low power consumption
Zigbee	Smart homes, industrial automation	IEEE 802.15.4	Up to 250 Kbps	AES-128 encryption	Low power consumption
Z-Wave	Smart homes, home automation	Z-Wave	Up to 100 Kbps	AES-128 encryption	Low power consumption
LoRaWAN	Smart cities, agriculture	LoRa	Up to 27 Kbps	AES-128 encryption	Ultra-low power consumption
MQTT	IoT messaging, telemetry	TCP/IP	Up to 256 Kbps	SSL/TLS encryption	Low power consumption
XMPP	Industrial automation	Custom	Up to 1 Gbps	Custom encryption	Low power consumption

1.3 Fault Diagnostics using Artificial Intelligence

Artificial Intelligence (AI) and Machine Learning (ML) are two rapidly growing fields in computer science that have gained significant attention in recent years. AI is the simulation of human intelligence in machines, while ML is a subset of AI that involves the development of algorithms that enable machines to learn from data without being explicitly programmed. These technologies have enormous potential to revolutionize many industries and change the way we live and work.

AI and ML have made significant progress in recent years, thanks to advances in hardware and software technology, as well as the availability of large datasets. AI systems can now perform a wide range of tasks, including speech recognition, image recognition, natural language processing, and decision-making. ML algorithms can learn from data to recognize patterns and make predictions, which has led to breakthroughs in fields such as healthcare, finance, and transportation.

One of the critical advantages of AI and ML is their ability to automate tasks previously performed by humans, leading to greater efficiency and productivity. For example, in healthcare, AI and ML are being used to develop diagnostic tools that can detect diseases such as cancer at an early stage, leading to better patient outcomes [9,10]. In finance, AI and ML are used to develop trading algorithms to make faster and more accurate investment decisions. In transportation, AI and ML are being used to develop self-driving cars that can reduce accidents and traffic congestion.

Fault diagnosis is a critical aspect of many engineering systems, as it is essential for ensuring their reliable and safe operation. However, the conventional methods of fault diagnosis involve manual inspection and testing, which can be time-consuming and prone to errors. To overcome these limitations, artificial intelligence (AI) techniques are increasingly being used for fault diagnosis.

AI-based fault diagnosis systems [20] can automatically detect faults and provide recommendations for maintenance and repair. These systems are typically based on machine learning (ML) algorithms that learn from the data collected from sensors and other sources. First, the data is pre-processed to remove noise and irrelevant information, and then the ML algorithms are trained on this data to recognize patterns and anomalies that indicate faults.

Another key advantage of AI-based fault diagnosis systems is their ability to learn and adapt to new situations. As a result, they can detect faults that may not have been identified before and provide more accurate and timely recommendations for maintenance and repair. AI-based systems can also analyze large amounts of data in real-time, which is particularly important for systems operating in complex and dynamic environments.

There are several AI techniques that are commonly used for fault diagnosis, including neural networks, fuzzy logic, genetic algorithms, and support vector machines. Neural networks are particularly useful for fault diagnosis, as they can model complex relationships between inputs and outputs and learn from examples. Fuzzy logic is another technique that is commonly used for fault diagnosis, as it can deal with imprecise and uncertain information. Finally, genetic algorithms can be used to optimize the parameters of a model for fault diagnosis, while support vector machines can be used for binary classification problems.

1.4 Hypotheses

The research field of fault diagnostics in electrical machines have taken a turn with the integration of information technology. This has paved way for development of condition monitoring and data acquisition systems, yet it is still lacking in multiple aspects like low sampling rates, mobility, complex etc. Most prominently, is the lack of a cost-effective system that can monitor and detect faults in real-time. Based on the current available data acquisition systems and diagnostic algorithms, we propose the following hypothesis:

- Microcontroller cards can be utilized as an alternative to PLC and SCADA based systems for a stable data acquisition system which can be flexible, scalable and reliable.
- A combination of Raspberry Pi with micro-controller card can help overcome the limitation of network connectivity and scalability for future implementations.
- Synthetic data can be generated using statistical equations to compensate the lack of data for training the machine learning models.
- The reliability of incoming data can be improved by analyzing the frequency spectrum.
- It would be possible to predict the probability of fault occurrence based on the frequency spectrum of an electrical machine.
- Machine learning techniques could be simplified to reduce training time and utilized for the detection and prediction of electrical machines faults in real-time.

1.5 Objectives of the Thesis

The main aim of the research work is to design and develop a cost-effective data acquisition tool that can be used to detect and predict faults in electrical machines utilizing machine learning methods. The system needs to be reliable, flexible, economical and can be scaled later on to run diagnostic algorithms in real-time. At the same time, the development of machine learning models and predictive algorithms for fault diagnostics is also important. Most of the currently available diagnostic algorithms do not have the capability to detect faults in real-time. The research goals for this work are:

- Development and implementation of a cost-effective data acquisition system based on micro-controller cards utilizing IoT.
- Fine-tuning the data acquisition setup so it can gather samples at high frequency which can also be used to train machine learning models.
- Deployment of custom machine learning methods to reduce training time.
- Develop and implement a method to generate synthetic signals using improvised statistical equations for the training of fault detection models.
- Development and implementation of fault prediction algorithm based on frequency spectrum for electrical machines.

1.6 Scientific Contributions

1.6.1 Scientific Novelty

- Designing the structure of the data acquisition system so that it is scalable and flexible for future implementations. In most data acquisition systems, only one of the microcontroller cards is considered and there is nothing about a local backup/node or anything about the hardware on-premise implementation of fault detection.
- The proposed data acquisition system is made compatible with analog sensors by utilizing a level shift circuit and calibration of sensors.
- The method was developed to derive improvised statistical equations from signature fault frequency equations of electrical machine faults and the generation of synthetic signals for training machine learning models to compensate for the lack of available data regarding faults.
- Development of custom machine learning algorithms with the simplest form and different activation functions to reduce training time and focus solely on electrical machine faults.
- Implementation of the method to recover missing data points and improve spectrum resolution of the current spectrum gathered through low-power signal processing devices.
- A signal spectrum-based machine learning algorithm was developed for fault prediction of electrical machines. As there are not many predictive algorithms present, the proposed technique can help reduce shutdowns and fault occurrences in electrical machines.

1.6.2 Practical Novelty

- Development and implementation of the data acquisition system in the lab with different test benches.
- Development of the test rig to gather and test the data acquisition system for different faults.
- Development of a web-based panel for displaying and analyzing the stored data in the cloud for end-user.
- Implementation of synthetic signals based trained model to endorse their accuracy and validation against the trained models using measured signals.
- Comparative analysis of custom machine learning trained models based on different variants.
- Implementation of the signal spectrum-based machine learning technique to check the accuracy of fault prediction using signals gathered from physical setup for BRB and bearing faults.

1.7 Outline of the thesis

The thesis is structured into five subsections which are as follows.

Chapter 2 focuses on the survey of related work. This mainly focuses on electrical machines faults, IoT devices, microcontroller boards, IoT communication protocols and machine learning techniques.

Chapter 3 covers the design and implementation of condition monitoring and data acquisition system for electrical machines. It includes the measurement setup and the data acquisition system with its validation and fault detection in real-time using machine learning based trained models.

Chapter 4 describes details regarding to the improvement of data sets which can be used for training of machine learning models. The first half of the chapter covers generation of synthetic signals to compensate for lack of measured data related to different faults. The middle half covers the improvement technique for frequency spectrum resolution which can help cover components lost due to discontinuity or network interruptions. The chapter closes with a comparative analysis between two different variants of a custom machine learning algorithm which is solely focused on faults of electrical machines.

Chapter 5 covers the details regarding fault prediction algorithm based on frequency spectrum. The first half of the chapter gives an overview of the algorithm and the second half validates the algorithm with trained models on measured readings.

Chapter 6 presents the conclusion and future work of this study.

2 Literature Review

2.1 Electrical Machines Faults

Electrical machine faults [21] can be broadly classified into two categories: electrical and mechanical faults. Electrical faults are related to the electrical components of the machine, while mechanical faults are related to the mechanical components of the machine. Electrical faults can be further subdivided into stator faults and rotor faults. Stator faults are faults that occur in the stationary part of the machine, while rotor faults are faults that occur in the rotating part of the machine. Some common types of stator faults include:

- **Short-circuits:** A short-circuit occurs when the winding insulation breaks down, causing the winding to become electrically connected to another winding or machine frame. Short-circuits can cause overheating, which can lead to insulation breakdown and failure of the winding.
- **Open-circuits:** An open-circuit occurs when the winding is physically broken, causing a break in the electrical path. Open-circuits can result in reduced machine performance, overheating, and insulation breakdown.
- **Ground faults:** A ground fault occurs when one of the winding's insulation breaks down, causing the winding to become electrically connected to the machine frame. Ground faults can cause overheating, as well as safety hazards, such as electric shocks.

Rotor faults can be further subdivided into broken rotor bars and rotor eccentricity.

- **Broken rotor bars:** They occur when one or more rotor bars break, leading to unbalanced currents and increased vibration.
- **Rotor eccentricity:** It occurs when the rotor's centerline is not aligned with the stator's centerline, leading to increased vibration and reduced performance.

Mechanical faults can be further subdivided into bearing faults and coupling faults.

- **Bearing faults:** These occur when the bearings that support the rotating components of the machine fail, leading to increased vibration and potential catastrophic failure.
- **Coupling faults:** They occur when the coupling between the motor and the driven load fails, leading to increased vibration and potential catastrophic failure.

Various factors, including environmental conditions, manufacturing defects, and operating conditions, can cause electrical machine faults. Some common causes of electrical machine faults include:

- **Environmental conditions:** Environmental conditions, such as temperature, humidity, and dust, can affect the insulation of the winding and the bearings, leading to insulation breakdown and bearing failure.
- **Manufacturing defects:** Manufacturing defects, such as poor quality control and material defects, can lead to insulation breakdown, bearing failure, and other faults.

- **Operating conditions:** Operating conditions, such as overloading and unbalanced currents, can lead to increased vibration, insulation breakdown, and other faults.

Diagnosis and prevention of electrical machine faults are critical to ensuring machine performance and reliability. There are various methods for diagnosing electrical machine faults, including visual inspection, vibration analysis, and electrical testing. An overview of different types of electrical faults with their causes and detection techniques is given in Table 2.1.

Table 2.1. Different types of electrical faults with causes and detection techniques

Type of Electrical Fault	Description	Causes	Detection Techniques	References
Stator winding faults	Shorted turns	Turn-to-turn insulation breakdown caused by thermal or electrical stress, leading to a short circuit between two or more turns	Park's vector approach, high frequency signal injection, motor current signature analysis	[1], [2], [3]
	Open circuits	Breakage or disconnection of a turn, resulting in a break in the current flow and loss of motor function	Voltage and current signature analysis	[2], [4], [5]
	Ground faults	Occurs when one or more turns of the stator winding come in contact with the ground or the motor housing, leading to a short circuit	Current signature analysis, leakage reactance measurement	[1], [3], [6]
Rotor faults	Broken rotor bars	Breakage or cracking of rotor bars due to mechanical stresses, leading to an imbalance in the rotor and vibration in the motor	Motor current signature analysis, vibration analysis	[2], [4], [7]
	Rotor eccentricity	Misalignment of the rotor with respect to the stator, leading to an uneven air gap between the rotor and stator and resulting in vibration and increased losses	Motor current signature analysis, magnetic flux analysis	[2], [8], [9]

Table 2.1. Different types of electrical faults with causes and detection techniques (continued)

Type of Electrical Fault	Description	Causes	Detection Techniques	References
Rotor faults	Rotor rubbing	Contact between the rotor and stator due to misalignment or bearing problems, leading to increased friction and vibration in the motor	Motor current signature analysis, vibration analysis, eddy current testing	[2], [10], [11]
	Shaft misalignment	Misalignment between the motor shaft and the driven equipment, resulting in increased vibration and wear on the bearings	Motor current signature analysis, vibration analysis, shaft alignment measurement	[12], [13], [14]
	Load misalignment	Misalignment between the driven equipment and the load, resulting in increased vibration and wear on the bearings	Motor current signature analysis, vibration analysis, load alignment measurement	[12], [13], [14]
Electrical supply faults	Voltage sags and interruptions	Momentary voltage drops or complete interruptions in the power supply, leading to fluctuations in the motor's performance and possible damage	Power quality analysis, voltage monitoring	[1], [3], [15]
	Harmonic distortions	Non-sinusoidal waveforms superimposed on the power supply, leading to additional losses and overheating of the motor	Power quality analysis, harmonic analysis	[1], [16], [17]
	Transient voltages	Voltage spikes or surges due to lightning strikes, switching operations or other causes, leading to insulation breakdown and damage to the motor	Voltage monitoring, insulation resistance testing	[18], [19], [20]

2.2 IoT Devices

The Internet of Things (IoT) [22,23] has become an increasingly popular technology in recent years. It involves connecting physical devices, appliances, and even vehicles to the internet, allowing them to communicate and exchange data with each other. IoT devices have the potential to revolutionize industries such as healthcare, transportation, and manufacturing by enabling real-time monitoring and data analysis.

One of the key advantages of IoT devices is their ability to collect and analyze vast amounts of data in real-time. This data can be used to improve efficiency, reduce costs, and enhance safety. For example, IoT-enabled sensors can monitor temperature and humidity levels in a manufacturing facility, allowing workers to identify potential issues and prevent equipment failure before it occurs [24]. Similarly, IoT devices can be used in healthcare to monitor patients remotely, providing doctors with real-time data on their condition [25].

However, the widespread adoption of IoT devices has also raised concerns about privacy and security [26,27]. As more and more devices are connected to the internet, there is an increased risk of cyberattacks and data breaches. In 2020, the FBI issued a warning about the security risks associated with IoT devices, highlighting the importance of implementing strong security measures to protect sensitive data. Another challenge facing the widespread adoption of IoT devices is interoperability. With so many different devices and communication protocols, ensuring that they can all work together seamlessly can be a significant challenge. This has led to the development of standardization efforts such as the Open Connectivity Foundation and the Thread Group, which aim to create a common framework for IoT devices to operate within.

Despite these challenges [28], the adoption of IoT devices is expected to continue to grow in the coming years. According to a report by Statista, the number of connected IoT devices is expected to reach 75.44 billion by 2025. This growth is driven by the increasing availability of low-cost sensors and wireless connectivity, as well as the growing demand for real-time data analysis in a variety of industries.

IoT devices have the potential to transform industries by enabling real-time monitoring and data analysis. However, challenges such as security risks and interoperability must be addressed to ensure their widespread adoption. As the technology continues to evolve, it will be important to balance the potential benefits of IoT devices with the need for robust security and privacy measures.

The infrastructure of IoT devices also involves various supporting technologies, such as cloud computing, big data analytics, and artificial intelligence (AI). Cloud computing provides a scalable and cost-effective platform for data storage, processing, and analysis. Big data analytics enables the processing and analysis of large and complex datasets generated by IoT devices. AI techniques, such as machine learning and deep learning, enable intelligent decision-making based on the data collected by IoT devices.

However, the infrastructure of IoT devices also presents several challenges, such as data security and privacy, interoperability, and scalability. The massive volume of data generated by IoT devices requires robust security measures to protect sensitive information and prevent unauthorized access. Interoperability refers to the ability of different IoT devices and systems to communicate and work together seamlessly, which requires standardized protocols and interfaces. Scalability refers to the ability of the infrastructure to handle a growing number of devices and data without performance degradation. The infrastructure of IoT devices is complex and involves multiple layers and supporting technologies. The infrastructure presents both opportunities and challenges for the development and deployment of IoT devices. It is crucial to address the challenges and ensure a secure, interoperable, and scalable infrastructure to fully realize the potential of IoT devices.

2.3 Microcontroller Boards

Microcontroller cards, also known as microcontroller development boards or microcontroller units (MCUs), are electronic devices that consist of a microcontroller, a clock oscillator, input/output pins, and various other components [29]. These devices have become increasingly important in the field of electrical machines due to their ability to control and monitor machine operations in real-time. They are particularly useful in applications where precise control of electrical machines is necessary, such as in motor control, power electronics, and automation systems [30].

One of the key advantages of using microcontroller cards in electrical machines is their ability to provide real-time monitoring and control of machine operations. By integrating sensors with the microcontroller card, it is possible to monitor machine parameters such as speed, temperature, and vibration and adjust the machine's operation accordingly. This allows for more precise and efficient control of the machine, which can lead to improved performance, reduced energy consumption, and increased reliability [31].

Another advantage of microcontroller cards is their flexibility and versatility. These devices can be programmed using a variety of programming languages and development environments, allowing for customized control and monitoring of machine operations [32]. Additionally, many microcontroller cards are designed to be modular, with various add-on modules available to expand their capabilities. This allows for easy customization of the device to meet specific application requirements.

Despite the advantages of microcontroller cards, there are also some challenges associated with their use in electrical machines. One of the key challenges is the need for specialized knowledge and skills to program and use these devices effectively. This can be a barrier for some users, particularly those without a strong background in electronics or programming.

Overall, microcontroller cards are an important and efficient tool for controlling and monitoring electrical machines [33]. They offer real-time monitoring and control capabilities, flexibility, and versatility and can lead to improved performance, reduced energy consumption, and increased reliability. As the field of electrical machines continues to evolve and become more advanced, the use of microcontroller cards is likely to become even more widespread. A comparison of different microcontroller cards is shown in Table 2.2.

Table 2.2. Comparison of microcontroller cards

Board	Processor	Memory	I/O Pins	Operating Voltage	Price	Reference
Arduino uno	ATmega328P	32KB flash	20	5V	$23	[34]
Arduino mega	ATmega2560	256KB flash	54	5V	$38	[35]
Teensy 4.1	ARM Cortex-M7	2MB flash	46	3.3V	$27	[36]
Raspberry Pi pico	RP2040	2MB flash	26	3.3V	$4	[37]
ESP32 devkit V1	ESP32	4MB flash	38	3.3V	$8	[38]
STM32 nucleo-64	STM32F401RE	512KB flash	64	3.3V	$13	[39]
Raspberry Pi 4 model B	Broadcom BCM2711	2GB RAM	40	5V	$35	[40]
BeagleBone black	TI Sitara AM3358BZCZ 100	512MB RAM	92	5V	$55	[41]
ESP8266 nodeMCU V3	ESP8266EX	4MB flash	11	3.3V	$7	[42]
Adafruit feather M4 express	ATSAMD51J19	2MB flash	21	3.3V	$22	[43]
Particle argon	Nordic nRF52840	1MB flash	18	3.3V	$15	[44]

2.4 IoT Communication Protocols

There are several IoT communication protocols available, which can be divided into two categories: wired and wireless. Wired protocols use cables or wires to connect IoT devices, whereas wireless protocols use electromagnetic waves to communicate wirelessly.

Wired IoT communication protocols are used when the devices are located near each other or when the devices are connected to the same network. Some of the popular wired IoT communication protocols are:

- **Ethernet:** Ethernet is a widely used wired communication protocol that is used for local area networks (LANs). It provides high data rates and low latency, making it suitable for real-time applications [45].
- **RS-485:** RS-485 is a differential serial communication protocol that is used for connecting multiple devices over long distances. It is commonly used in industrial automation and control systems [46].
- **CAN:** Controller Area Network (CAN) is a serial communication protocol that is used in vehicles and industrial automation. It provides high reliability, fault tolerance, and real-time performance [47].

Wireless IoT communication protocols are used when the devices are located far from each other or when it is difficult to install wires. Some of the popular wireless IoT communication protocols are:

- **Wi-Fi:** Wi-Fi is a widely used wireless communication protocol that provides high data rates and long-range connectivity. It is commonly used in home and office networks.
- **Bluetooth:** Bluetooth is a wireless communication protocol that is used for short-range communication between devices. It is commonly used in wireless headphones, speakers, and smartwatches [48].
- **Zigbee:** Zigbee is a low-power wireless communication protocol that is used for home automation, smart lighting, and industrial automation. It provides low data rates and low power consumption [49].
- **Z-Wave:** Z-Wave is a wireless communication protocol that is used for home automation and smart homes. It provides low data rates and low power consumption [48].
- **LoRaWAN:** LoRaWAN is a long-range, low-power wireless communication protocol that is used for IoT devices. It provides long-range connectivity and low power consumption, making it suitable for battery-powered devices.
- **Sigfox:** Sigfox is a low-power, wide-area wireless communication protocol that is used for IoT devices. It provides long-range connectivity and low power consumption.
- **NB-IoT:** Narrowband IoT (NB-IoT) is a low-power, wide-area wireless communication protocol that is used for IoT devices. It provides long-range connectivity and low power consumption.
- **LTE-M:** LTE-M is a low-power, wide-area wireless communication protocol that is used for IoT devices. It provides long-range connectivity and low power consumption.
- **RFID:** Radio Frequency Identification (RFID) is a wireless communication protocol that is used for tracking and identifying objects. It provides low data rates and short-range connectivity.
- **NFC:** Near Field Communication (NFC) is a wireless communication protocol that is used for contactless payments and data transfer. It provides short-range connectivity and low data rates [50].
- **Thread:** Thread is a wireless communication protocol that is used for home automation and smart homes. It provides low data rates and low power consumption.
- **6LoWPAN:** IPv6 over Low-Power Wireless Personal Area Networks (6 LoWPAN) is a wireless communication protocol that is used for IoT devices. It provides low power consumption and low data rates[51].
- **MQTT:** Message Queuing Telemetry Transport (MQTT) is a messaging protocol that is used for IoT devices. It provides lightweight, low bandwidth, and reliable communication between devices [52–55].
- **CoAP:** Constrained Application Protocol (CoAP) is a protocol that is used for resource-constrained IoT devices. It provides lightweight and efficient communication between devices [56,57].
- **XMPP:** Extensible Messaging and Presence Protocol (XMPP) is a messaging protocol that is used for IoT devices. It provides real-time communication and messaging between devices.

2.5 Machine Learning Techniques

Machine learning (ML) is a subfield of artificial intelligence (AI) that deals with the design and development of algorithms that can learn from data. These algorithms can be broadly classified into three categories: supervised learning, unsupervised learning, and reinforcement learning. In addition to these, there are also deep learning and semi-supervised learning techniques. An overview of machine learning techniques is shown in Figure 2.1.

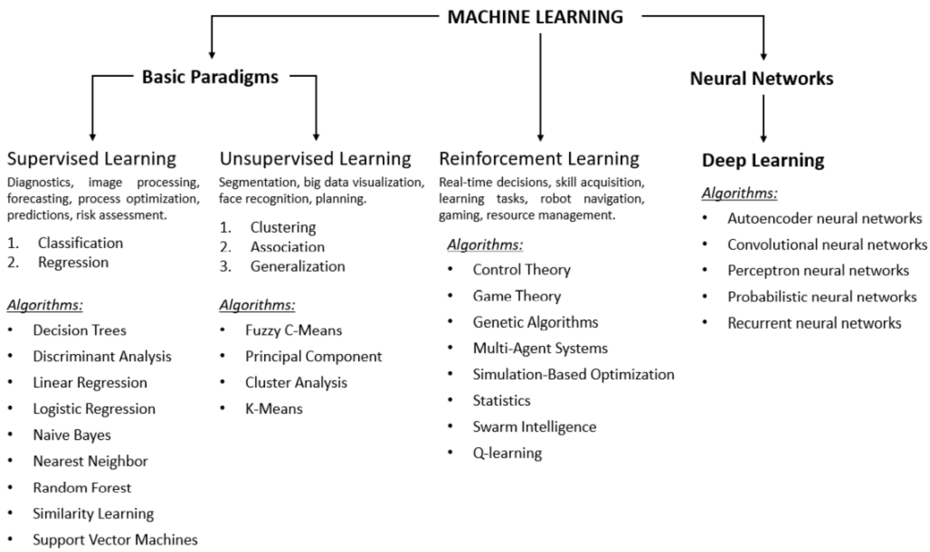


Figure 2.1. Machine learning algorithms [58]

Supervised learning is a type of machine learning in which an algorithm is trained on labeled data [59,60]. The labeled data consists of input variables (features) and an output variable (label or target) that the algorithm tries to predict. The goal of supervised learning is to learn a mapping from input variables to output variables that can generalize to new, unseen data. Linear regression is a popular technique used in supervised learning for predicting a continuous variable based on one or more input variables [61]. Logistic regression is used for predicting a binary or categorical variable based on one or more input variables [62]. Decision trees are a popular technique used for classification and regression problems [63]. Random forests are an ensemble learning technique that uses multiple decision trees to improve accuracy [64]. Support vector machines are used for classification and regression problems, and they work by finding the hyperplane that maximizes the margin between different classes [65]. Neural networks are a powerful technique used for a wide range of supervised learning tasks, including image recognition, natural language processing, and speech recognition [65,66].

Unsupervised learning is a type of machine learning in which an algorithm is trained on unlabeled data[67], [68]. The algorithm tries to find patterns or structures in the data without any prior knowledge of the output variable. Clustering is a popular technique used in unsupervised learning, where the goal is to group similar data points together [69]. K-means clustering is a simple and widely used technique that partitions data into K clusters based on their similarity [70]. Hierarchical clustering is another clustering technique that organizes data into a hierarchy of clusters [71]. Principal component analysis (PCA) is a technique used for dimensionality reduction, where the goal is to reduce the number of input variables while retaining as much information as possible [72].

Deep learning is a subfield of machine learning that uses neural networks with multiple layers to learn hierarchical representations of data [73]. Deep learning has achieved state-of-the-art performance on many complex tasks, including image and speech recognition, natural language processing, and playing games [74]. Convolutional neural networks (CNNs) are a type of neural network commonly used for image recognition [75]. Recurrent neural networks (RNNs) are another type of neural network commonly used for natural language processing [76]. Generative adversarial networks (GANs) are a type of neural network used for generating realistic images [77].

Reinforcement learning is a type of machine learning in which an agent learns to take actions in an environment to maximize a reward signal. The goal of reinforcement learning is to find an optimal policy that maps states to actions [78,79]. Q-learning is a popular reinforcement learning technique that uses a Q-function to estimate the expected reward of taking action in a given state [80]. Deep reinforcement learning combines reinforcement learning with deep learning techniques to learn complex policies from high-dimensional inputs, such as images [79].

Semi-supervised learning is a type of machine learning in which an algorithm is trained on a combination of labeled and unlabeled data. The goal of semi-supervised learning is to use the unlabeled data to improve the performance of the model on the labeled data [81]. Label propagation is a popular semi-supervised learning technique that propagates labels from labeled data to unlabeled data based on their similarity [81]. Self-training is another semi-supervised learning technique that involves using a model trained on labeled data to make predictions on unlabeled data and then adding the confident predictions to the labeled data to train a better model [82].

Machine learning has many techniques that can be used for a wide range of applications. Supervised learning is useful when labeled data is available, while unsupervised learning is useful when only unlabeled data is available. Deep learning is a powerful technique that can learn hierarchical representations of data, and reinforcement learning can be used to learn optimal policies for decision-making tasks. Semi-supervised learning is useful when there is limited labeled data available and can improve the performance of models trained on labeled data. Understanding the different types of machine learning techniques and their applications can help researchers and practitioners choose the most appropriate technique for their specific task. Table 2.3 lists a comprehensive overview of different machine learning techniques.

Table 2.3. Comparison of different machine learning techniques

Machine Learning Technique	Pros	Cons	References
Linear regression	Easy to interpret and implement, efficient in terms of training and inference time.	Limited flexibility in modelling complex relationships can be sensitive to outliers.	[60,61,83]
Logistic regression	Simple and easy to interpret, it works well for binary classification tasks.	Limited ability to capture complex relationships, prone to overfitting in high-dimensional data.	[60,62,84]
Decision trees	Easy to interpret and visualize, can handle both categorical and numerical data.	Prone to overfitting, sensitive to small changes in the data, can produce biased trees if the data is imbalanced.	[63,85–87]
Random forests	Robust against overfitting, can handle both categorical and numerical data, it works well with high-dimensional data.	Less interpretable than decision trees, it can be computationally expensive.	[64,88,89]
Support vector machines (SVM)	Effective in high-dimensional spaces, can handle non-linear data, can be regularized to avoid overfitting.	Computationally expensive for large datasets, sensitive to the choice of the kernel function.	[90–92]
Naive bayes	Simple and easy to implement, works well with high-dimensional data, computationally efficient.	Assumes that all features are independent, can be sensitive to irrelevant features.	[93–95]
K-nearest neighbors (KNN)	Easy to implement, works well with small datasets and non-linear data, can handle both classification and regression tasks.	Computationally expensive for large datasets, sensitive to the choice of distance metric, requires careful pre-processing of the data.	[65,96,97]
Gradient boosting	Can handle heterogeneous data, produces accurate predictions, works well with imbalanced data.	Can be sensitive to noisy data, computationally expensive, difficult to interpret.	[98–100]
Neural networks	Can handle complex relationships and high-dimensional data, can learn features automatically, can be regularized to avoid overfitting.	Requires large amounts of data to train, can be computationally expensive, can be difficult to interpret.	[66,101,102]

Table 2.3. Comparison of different machine learning techniques (continued)

Machine Learning Technique	Pros	Cons	References
Recurrent neural networks (RNN)	Effective in modelling sequential data, can handle variable-length inputs and outputs.	Can be sensitive to vanishing gradients, can be computationally expensive, may require careful initialization.	[76,101]
Convolutional neural networks (CNN)	Effective in modelling images and other types of spatial data, can learn hierarchical representations.	Can be computationally expensive, may require careful initialization and regularization.	[73,101,103]
Principal component analysis (PCA)	Reduces the dimensionality of the data while preserving most of the variability, computationally efficient.	Information loss due to dimensionality reduction, can be sensitive to outliers.	[64,67]
Clustering	Can discover underlying patterns and structure in the data, easy to implement.	Sensitive to initialization, can be difficult to determine the optimal number of clusters, can be sensitive to outliers.	[64,69]
Association rule mining	Can discover frequent co-occurrences and associations among items in a dataset, useful in market basket analysis.	Easy to implement, works well	[104]
Long short-term memory (LSTM)	Effective in modelling sequential data with long-term dependencies, can handle variable-length inputs and outputs.	Can be computationally expensive, may require careful initialization and regularization.	[76,92]

2.6 Chapter Summary

This chapter provides an in-depth literature review regarding electrical machine faults, IoT devices, microcontroller cards, communication protocols and machine learning algorithms. It can be concluded from the chapter that although the integration of information technology with electrical machines is not new, it still has a long way to go in terms of development and research. IoT devices are becoming more and more common in the industry to monitor the health of electrical machines, but it is still lacking in terms of sampling frequency, data analysis, scalability, etc. Machine learning based trained models are also useful tools for fault detection of electrical machines. However, these required a large number of data sets and there is a lack of predictive algorithms in terms of electrical machines.

3 Condition Monitoring of Electrical Machines

Fault diagnostics and condition monitoring of electrical machines have always been one of the primary concerns in the industry. With the advent of Industry 4.0, condition monitoring and predictive maintenance have become the forefront of current research areas. Electrical machines play a significant role in multiple domestic and industrial applications, so keeping track of their health is essential. Due to the environment, constant running, external and internal stresses and other parameters constantly affecting electrical machines, there is a high chance of fault occurrence.

With the recent advancements in information technology, industries are moving towards predictive and proactive maintenance from periodic and reactive maintenance. As periodic and reactive maintenance costs time and money due to the shutdown of the equipment, more and more research is being done into finding new ways for predictive and proactive maintenance. This has seen a recent boost with the integration of information technology with physical devices, also known as smart devices.

This helped monitor the electrical equipment in real-time and gather data from the equipment to develop algorithms for fault prediction. As most of the fault detection algorithms were running offline and it takes much more time to be able to implement them in real-time, like Finite Element Method (FEM) Analysis [105,106]. This combination of information technology with electrical devices changed the landscape of fault detection and prediction in electrical machines. It not only gives way to real-time fault detection but also further opens up research areas to enhance it into fault prediction, which helps reduce cost and time for maintenance.

The implementation of smart devices for industrial applications has further streamlined the industrial process and made it easier for end users to determine the issues with the monitored electrical equipment, if any. These smart devices can communicate with each other and implement logical decisions based on incoming data. The setup of these smart devices over a network (e.g., the internet) is termed as Internet of Things (IoT) [7]. The monitoring of electrical equipment with the help of IoT and cloud in real-time is usually known as remote sensing or monitoring. The general implementation of a condition monitoring system in an industrial application is shown in Figure 3.1.

Here the data is read from the electrical machine through sensors which are then forwarded onto the control center or cloud using a data acquiring system. The cloud then runs any analysis that is needed to be done on the incoming data for the detection of faults and the processed data is then shown on a GUI for the end user. This makes it easier for the user or technician to monitor the health of the electrical machines and to identify if those machines need maintenance or not. This is a general example of a condition monitoring system in terms of an industrial application.

3.1 Data Acquisition System

A data acquisition system is one of the most fundamental and crucial parts of a condition monitoring system as it helps collect data from the electrical machine. Most of the condition monitoring system that has been developed are either using SCADA or PLC, which are expensive, harder to transport and complex. With the recent advancements in the technology of microcontroller cards, researchers are trying to develop condition monitoring systems based on them, but these systems are still being improved and not much analysis is being done on the collected data.

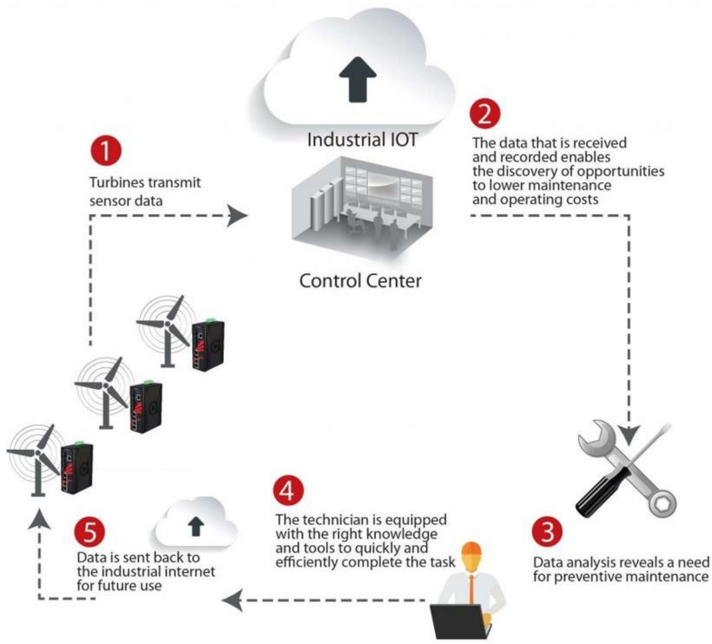


Figure 3.1. Industrial application of condition monitoring system [107]

These microcontroller boards are not only cost-efficient but also flexible and scalable for future development. The overview of the proposed data acquisition system is shown in Figure 3.2.

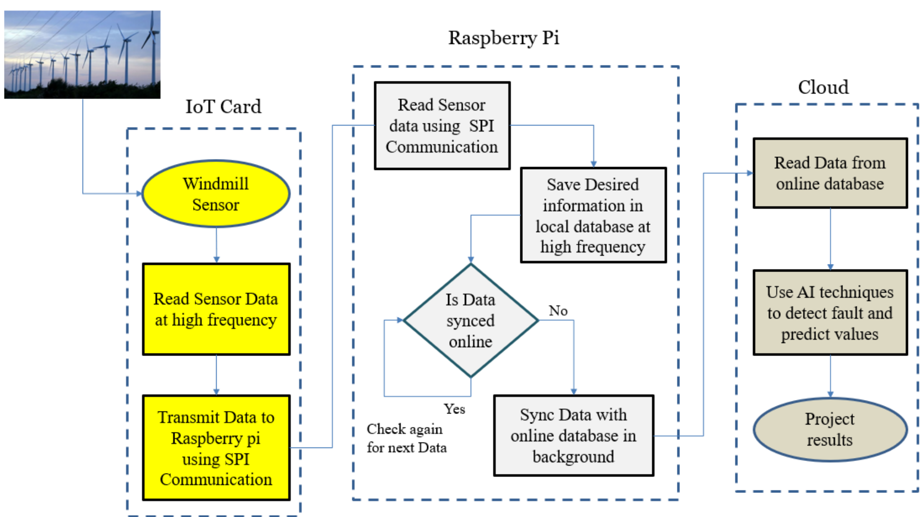
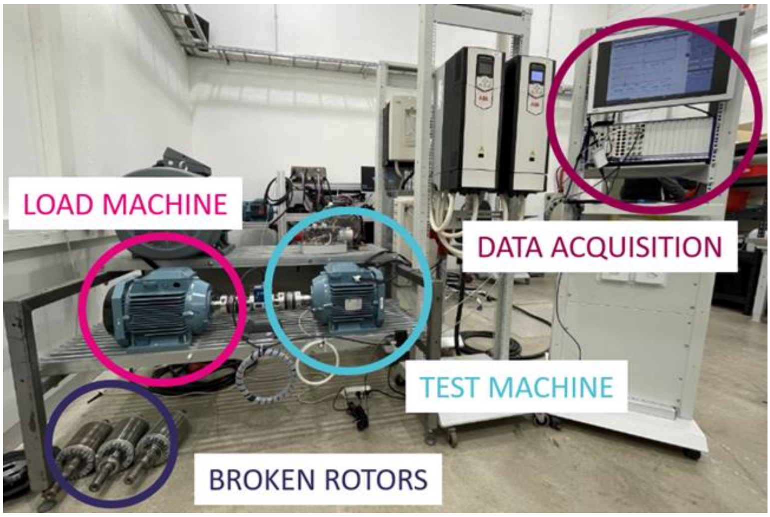


Figure 3.2. Overview of the data acquisition system (previously published in article VI)

3.1.1 Measurement Setup

For experiments and gathering of data from electrical machines, a test rig is setup in the lab. The setup consists of two motors where one acts as the loading motor and the other is the test motor with different faults in it. The detailed specifications for the motors are given in Table 3.1. The experimental setup for the test bench to collect data for different faults is shown in Figure 3.3.



(a)

(b)

Figure 3.3. Test rigs for data collection and testing of faults (previously published in article V)

The data is collected from the electrical machine in a healthy state using both Dewetron and the proposed data acquisition system. This is for verification of data collected through the data acquisition system to make sure there are no errors. The data for training was mostly collected through Dewetron, whereas the data acquisition system was used in between to verify the transfer of data to the cloud. The data collected is at different loads for both cases so as to diversify the collected data. The loading machine is fed power through an inverter for better control at different load levels. The data is collected at a sampling frequency of 20 kHz and the measurement time is for 5-6 minutes. The data collected through Dewetron consists of values of voltage, current, speed, torque and vibration.

Table 3.1. Specifications of motor

Parameter	Value
Number of poles	4
Number of phases	3
Connection	Delta
Stator slots	36; non-skewed
Rotor bars	28; skewed
Rated voltage	400V @50 Hz
Rated current	8.8 A
Rated power	7.5 kW @ 50 Hz

3.1.2 Components

The proposed data acquisition system consists of a microcontroller card coupled with raspberry Pi which will act as a local node and database backup. It will also sync data with the online database in the cloud for further analysis and computations. There is a level shifter in between the microcontroller card and raspberry pi as the output pins of the microcontroller card give a 3.3 V output. Whereas, for Pi the input should be at 5 V. The components of the data acquisition system are shown in Figure 3.4.

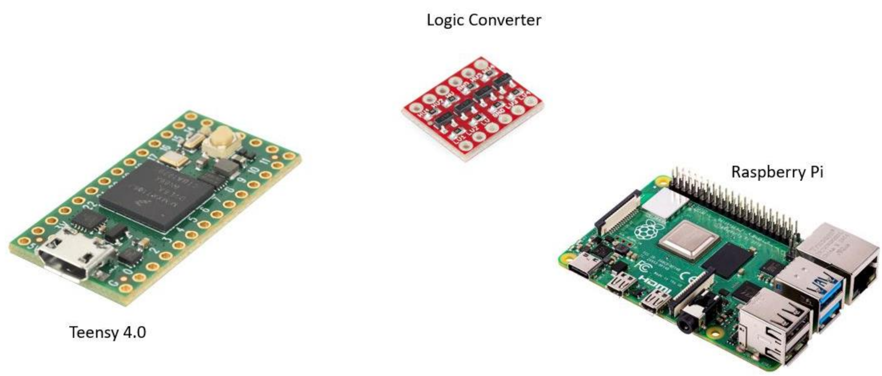


Figure 3.4. Components of data acquisition system

Technical specifications for the Teensy 4.0 (microcontroller card) and Raspberry Pi are given in Table 3.2 and Table 3.3, respectively. Whereas the microcontroller card selected is due to more processing power and less size.

Table 3.2. Technical specifications of Teensy 4.0

Feature	Specification
Microcontroller	ARM Cortex-M7 @ 600 MHz
Flash memory	2 MB
SRAM	512 KB
Digital I/O pins	40 (including 31 PWN outputs)
Analog inputs	14
CAN bus	3 (1 with CAN FD)
Serial	7
SPI	3
I2C	3
Operating voltage	3.3V
Dimensions	62 mm x 18 mm x 4 mm

Table 3.3. Specifications of Raspberry Pi

Feature	Specification
Processor	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
Memory	8 GB LPDDR4-3200 SDRAM
Connectivity	2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
LAN	Gigabit ethernet
USB ports	2 USB 3.0 ports; 2 USB 2.0 ports
HDMI	2 × micro-HDMI ports (up to 4kp60 supported)
I/O pins	Raspberry Pi standard 40 pin GPIO header
SD card	Micro-SD card slot for loading operating system and data storage
Operating voltage	5V DC via USB-C connector (minimum 3A*)

For the test purpose, two different microcontroller cards were used with different communication protocols and the setup was run for multiple days to make sure the data transmission was smooth and there was no loss of data. The comparison between Arduino and Teensy 4.0 is shown in Table 3.4, whereas the sample rate per second for different communication protocols is shown in Table 3.5, where there is no data loss at high frequency, even when the communication span is for weeks.

Table 3.4. Comparison of sample rate for different communication methods (previously published in article V)

Communication Method	Sample Rate/s
UART	1800
I2C	2600
SPI interface	3600

Table 3.5. Time taken by IoT card for processing 10,000 samples data transmission without loss of data (previously published in article VI)

Card	Processing time for 10,000 samples in seconds
Arduino mega	4.5 s
Teensy 4.0	0.75 s

The steps of the algorithm running on teensy 4.0 (microcontroller card) are as follows:

- Set up the Arduino board and the current sensor.
  - This involves connecting the current sensor to an analog pin on the Arduino board and setting it up to measure current values.
- Initialize the SPI communication interface on the Arduino.
  - The Serial Peripheral Interface (SPI) is a synchronous communication protocol that allows devices to exchange data. In this case, the Arduino is set up to use SPI to communicate with the Raspberry Pi.
- Enter an infinite loop.
  - This ensures that the Arduino continues to read and send data until it is stopped or turned off.
- Read the current sensor values using an analog pin on the Arduino.
  - The analog pin on the Arduino is used to read the analog voltage output of the current sensor, which is proportional to the current being measured.
- Convert the sensor readings into a format that can be sent over the SPI interface to the Raspberry Pi.
  - The sensor readings are typically in analog voltage values, which need to be converted to digital values that can be transmitted over SPI. This is done by performing analog-to-digital conversion (ADC) on the analog voltage values.
- Send the sensor data over the SPI interface to the Raspberry Pi.
  - The ADC-converted digital values are sent over the SPI interface to the Raspberry Pi for storage and analysis.
- Wait for a short period of time before repeating the process.
  - This ensures that the Arduino does not send data too frequently and overload the Raspberry Pi, and also allows time for the Raspberry Pi to process the data before receiving more.

The data transmission between Teensy and Pi is done through Serial Peripheral Interface (SPI), which is faster than other communication protocols but is also fragile and needs to be set up with care. If the sample rate for data transmission is not much, other protocols like I2C or UART can also be used. The data is received on Pi and saved inside a local database synced with the cloud for further analysis and computation. The local database on Pi makes sure that no data is lost in between transmissions or due to network issues. The local node can also be further expanded into a cluster of nodes with multiple sensors attached to it and can act as a control unit. Different steps of the algorithm running on Pi are:

- Set up the Raspberry Pi and the local and online databases.
  - This involves installing any necessary software, configuring the databases, and setting up the Wi-Fi connection to enable communication with the online database.
- Initialize the SPI communication interface on the Raspberry Pi.
  - This allows the Raspberry Pi to receive data from the Arduino over SPI.
- Enter an infinite loop.
  - This ensures that the Raspberry Pi continues to receive and process data until it is stopped or turned off.
- Read the sensor data sent over SPI from the Arduino.
  - The Raspberry Pi reads the digital sensor data that was sent over SPI from the Arduino.
- Store the incoming sensor data in the local database.
  - The Raspberry Pi saves the incoming sensor data in a local database for storage and easy retrieval.
- Sync the local database with the online database using Wi-Fi.
  - The Raspberry Pi periodically sends the data stored in the local database to the online database using Wi-Fi to ensure that the data is backed up and accessible remotely.
- Wait for a short period of time before repeating the process.

3.2 Cloud Computation

Cloud computation has helped reduce resources and make setups more accessible, scalable and reliable. It allows the accessibility of data from anywhere at any time and cost-effective use of resources. The data saved in the online database is accessed here for pre-processing of signals. The data is further divided into samples which are then pre-processed to remove noise and converted into the frequency domain. An example of the sample data set frequency spectrum is shown in Figure 3.5. The frequency spectrum is then used for the detection of any faults based on trained machine learning models. If a fault is present, it is logged into the system, which is then shown to the end user, whereas the algorithm keeps on running to detect faults in any other incoming data.

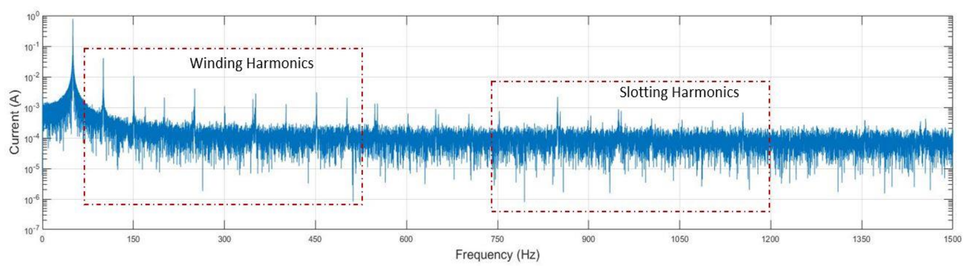


Figure 3.5. Frequency spectrum

After denoising the signal and converting it to the frequency domain, the resultant is then used for the detection of faults using machine learning trained models. The detailed steps of the algorithm running on the cloud for detection are as follows:

- Connect to the cloud database and retrieve the data to be processed.
  - This step involves establishing a connection to the cloud database and retrieving the data that needs to be processed. Depending on the cloud provider you are using, you might need to provide authentication details or other credentials to establish the connection.
- Pre-process the data as needed to prepare it for the machine learning model.
  - This step involves pre-processing the data to ensure that it is in a suitable format for the machine learning model. This might involve tasks such as cleaning, transforming, or normalizing the data. The exact pre-processing steps will depend on the type of data you are working with and the requirements of the machine learning model you are using.
- Load the trained machine learning model(s) into memory.
  - This step involves loading the trained machine learning model(s) into memory so that they can be applied to the data. Depending on the type of model you are using, you might need to use a specific library or framework to load the model.
- Apply the model(s) to the data to detect faults.
  - This step involves applying the machine learning model(s) to the pre-processed data to detect faults. Depending on the type of model you are using, this might involve running predictions or classifications on the data. The exact approach will depend on the requirements of the machine learning model you are using.
- If faults are detected, log the relevant information and take appropriate action (e.g., alert a human operator, reroute processing, etc.).
  - This step involves logging relevant information about any faults that are detected and taking appropriate action to mitigate the issue. Depending on the severity of the fault and the requirements of your system, you might need to alert a human operator, reroute processing to another system, or take other corrective measures.
- If no faults are detected, proceed with further processing as needed.
  - If no faults are detected, you can proceed with further processing of the data as needed. This might involve additional computations or transformations of the data, depending on the requirements of your system.
- Once all processing is complete, disconnect from the cloud database.
  - Once all processing is complete, you should disconnect from the cloud database to free up resources and ensure that your system is secure. Depending on the cloud provider you are using, you might need to use a specific method to disconnect from the database.

3.3 Machine Learning Trained Models for Detection

Machine learning models can be used for training models which are able to detect faults in incoming signals. This will help detect faults in real-time and reduce the costs overhead due to shutdowns and scheduled maintenance. There are different types of machine learning models available, but due to the training time and accuracy combination, here recurrent neural networks or deep learning techniques are not utilized. Machine learning algorithms try to map the data inputs to outputs based on patterns or different statistical equations. It can be expressed in general by Eq. (3.1).

$$\theta = \operatorname{argmin} \ 〚L(y, f(x; \theta))〛 \quad (3.1)$$

where, θ represents the parameters of the model. $〚L(y, f(x; \theta))〛$ is the loss function, which measures the difference between the predicted output $f(x; \theta))$ and the actual output *y*, *x* is the input data and argmin finds the values of θ that minimize the loss function.

The equation searches for the best fit value of θ by minimizing the difference between the original output and the predicted value of the output. Different techniques of machine learning used different optimization techniques to achieve this purpose. The most commonly used machine learning technique for the training of models is neural networks. Neural networks make use of the hidden layers to train complex models with high accuracy. Figure 3.6 shows the general schematics of the neural network training. Similarly, Neural networks can also be expressed using the generalized form in Eq. (3.2).

$$W(t+1) = W(t) - \alpha \nabla L(W(t)) \quad (3.2)$$

where, $W(t)$ represents the weights and biases of the network at iteration t. $L(W(t))$ is the loss function, which measures the difference between the predicted output of the network and the actual output. Whereas, α is the learning rate, which controls the step size taken in the direction of the negative gradient and $\nabla L(W(t))$ is the gradient of the loss function with respect to the weights and biases of the network, which represents the direction of the steepest descent.

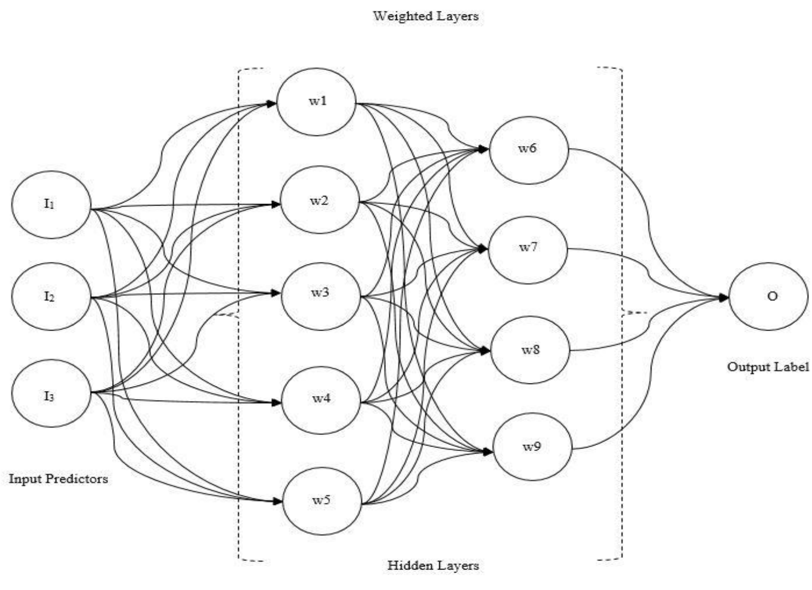


Figure 3.6. Neural network schematics (previously published in article VII)

The biases and weights of the network are updated with every iteration based on the gradient of the loss function. The process keeps on repeating itself until the total number of iterations is met or the loss function converges to a minimum value. Neural networks main aim is to get the most optimal value for the weights and biases so as to minimize the loss function and get higher accuracy for predictions.

In this case, different machine learning algorithms are used to train models for fault detection of the electrical machines. Here, the faults considered for the training of the machine learning models are broken rotor bars with one broken rotor bar, two broken rotor bars and three broken rotor bars. The trained model is implemented in the cloud for the detection of faults. Table 3.6 shows a comparison result of accuracy and time took for the training of machine learning models.

It can be seen that different machine learning models take different times for training and have different accuracies based on their technique to find the optimal values. Here, in this case, the wide neural network model performed best, but it might not be accurate for another data set. That is why there is a need to create custom machine learning models that are focused on electrical machine faults so as to get consistent results. Also, there is a lack of sufficient data for training these models for different faults, so it is also important to find an alternative way to get high quality data sets for different faults of electrical machines in quantity. Among all of the present machine learning models, neural network models give better results as compared with others.

Table 3.6. Comparison between different machine learning models

Machine Learning Algorithm	Accuracy (validation)	Time (s)
Fine gaussian SVM	86.00 %	21.81
Fine KNN	85.80 %	9.48
Coarse tree	75.60 %	0.85
Linear discriminant	92.60 %	1.73
Gaussian naive bayes	60.60 %	2.22
Kernel naïve bayes	80.80 %	291.94
Narrow neural network	95.90 %	47.11
Medium neural network	96.20 %	40.36
Wide neural network	96.60 %	27.61
Bilayered neural network	96.40 %	47.82
Trilayered neural network	95.50 %	49.74

3.4 Graphical User Interface (Dashboard)

Most of the techniques developed for the diagnosis of electrical machines are offline, like FEM and take some time to analyze and diagnose the electrical machines. In this case, the data is stored in the cloud and it is possible to detect faults in the incoming signals in real-time. To show the results or the incoming data to the end user, a graphical user interface (GUI) is designed. The GUI will not only present data but will also show different markings, including warnings and suggestions to the end user, depending upon the state of the motor and fault detection. An example of the GUI is shown in Figure 3.7.

Figure 3.7. GUI dashboard (previously published in article III)

3.5 Chapter Summary

IoT and cloud computing is becoming the new norm in the communication industry. This chapter covers a low-cost plug-and-play setup that can transmit data from remote locations or electrical machines. Furthermore, cloud computation can be used to detect faults in incoming data using models trained by machine learning. This will help contain the impact of faults on the machine and reduce overhead costs. The system is not only cost-efficient but is also easy to transport and implement, which gives it an edge when implemented in remote locations. This also helps the user to understand even a small fluctuation in values so that a major disaster can be avoided and identify the fault in a more systematic way.

4 Synthetic Signals & Custom Neural Network Algorithm

Machine learning algorithms need a large number of training datasets for an accurate and efficient trained model both in quality and quantity, but there is always a lack of data sets when it comes to the training of models for different kinds of faults. As it is difficult to generate data sets of faulty conditions of an electrical machine, researchers are looking into different areas to generate signals. Some are generating it through Simulink models does not consider external parameters on the motor like stress, deterioration over time, environmental effects, etc. Also, gathering data from actual setups not only takes time but is also expensive. This issue can be resolved using synthetic signals, which are generated through derived statistical equations of faults for electrical machines. Table 4.1 shows the fault signature frequencies for the induction motor.

Table 4.1. Fault signature frequencies for induction motor

<table>
<tr><th>Fault</th><th>Modulating Frequencies</th><th>where,</th></tr>
<tr><td>Rotor winding asymmetries</td><td>$$f_{asym} = f_s \pm 2ksf_s$$ k = 1,2,3, …</td><td rowspan="4">k: Harmonic order
v: Supply fed harmonics
f_s: Supply frequency
f_{asym}: Rotor winding asymmetries frequencies
f_{BR1}: Broken bar frequencies
f_{bb}: Bearing fault frequencies
$f_{(i,o)}$: Characteristic vibration frequencies
m: Positive integer
n_b: Number of rotor bars
n_{bb}: Number of bearing balls
n_d: Dynamic eccentricity (0 for static and 1,2,3…. for dynamic)
f_r: Rotor frequency
f_{ecce}: Eccentricity frequency
b_d: Bearing ball's diameter
p_d: Bearing pitch
ϑ: The angle between bearing ball and race
s: Slip
p: Number of poles
P: Number of pole pairs</td></tr>
<tr><td>Broken rotor bars</td><td>$$f_{BR1} = \left[\frac{k}{p}(1-s) \pm s\right] f_s$$ $\frac{k}{p} = 1,3,5, \ldots$</td></tr>
<tr><td>Bearing faults</td><td>$$f_{bb} = \left|f_s \pm m f_{i,o}\right|$$ $$f_{i,o} = \frac{n_{bb}}{2} f_r \left[1 \pm \frac{b_d}{p_d}\cos\theta\right]$$</td></tr>
<tr><td>Eccentricity & PSH</td><td>$$f_{ecce} = \left[(kn_b \pm n_d)\left(\frac{1-s}{P}\right) \pm v\right] f_s$$ For mixed eccentricity: $$f_{ecce} = f_s \pm kf_r$$</td></tr>
</table>

Similarly, another issue that occurs when training models using machine learning algorithms is that there is a chance that different algorithms can have different higher accuracies based on the data set and the number of faults. Although the training is not so complex but a different combination of faults can result in a different model performing better. So, there is a need to keep the results consistent and get an optimal trained model which is focused on electrical machine faults. As the future of this development is to have self-learning models which are able to better themselves based on new incoming data sets, machine learning algorithms are preferred for testing the deep learning ones.

4.1 Synthetic Signals

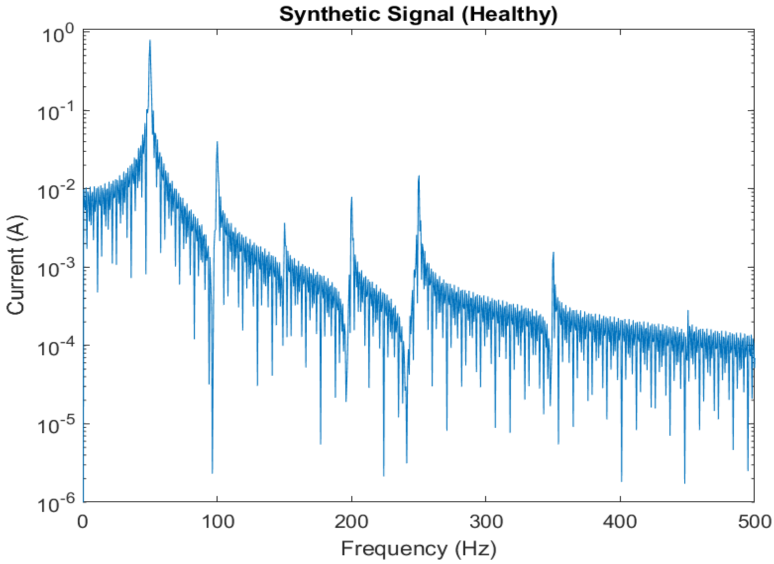
To overcome the shortfalls, data sets for faults are generated using the derived statistical equation for the fault signature frequencies. For testing purposes, the fault considered is that of broken rotor bars of an induction motor. It is also possible to get the derived statistical equations for different faults from the above table. The improvised equation for the generation of the signals based on broken rotor bars is shown in Eq. (4.1).

$$y = a * \sin(2\pi f_s t) + \sum b * \sin(2\pi f_{BR} t) \tag{4.1}$$

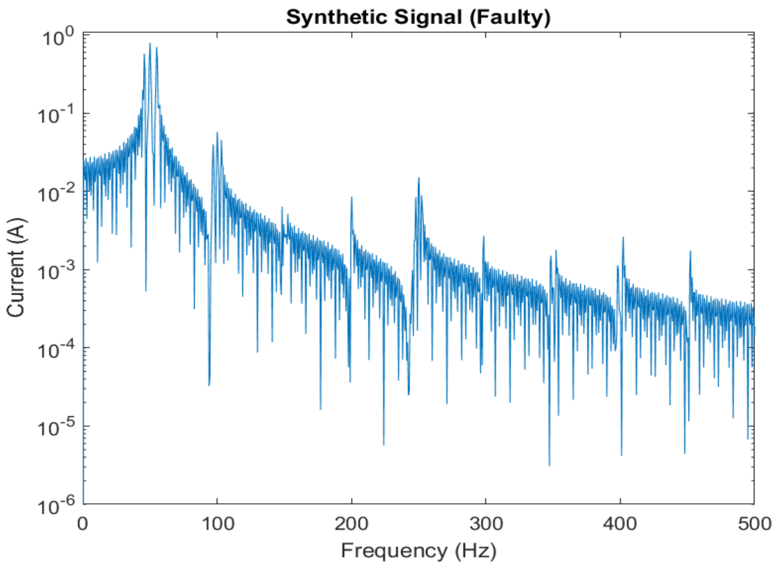
where, $f_{BR} = [kp\ (1-s) \pm s]\ f_s$
and kp = 1,3,5,….

In the above equation, fs is the supply frequency, whereas t is the time period. The variables a and b are multipliers for the generation of the amplitude of the generated faulty frequency components, which are categorized in ranges based on the number of broken bars. Whereas k is the harmonic order, s is the slip and p are the number of poles. The data sets generated from the equation are generic, with a random multiplier differentiating the amplitude of frequencies, especially the faulty frequencies inside the equation.

The generated signals are also compared with the real ones and found to be not exact but exhibit similar behavior. Although the synthetic signals need more processing afterward, at the moment, general pre-processing for smoothing out with RMS and normalization is taken into effect. A sample of generated synthetic signals in the frequency domain is shown in Figure 4.1.



(a)



(b)

Figure 4.1. (a) FFT of one of the healthy synthetic signal datasets (b) FFT of one of the faulty synthetic signal datasets used for training

4.1.1 Training of machine learning models for synthetic signals

The fault detection in real-time for incoming signals can also be divided into two parts. The first is the training of models and the second consists of the detection of those faults on the cloud. It is necessary to know about the distinctive features related to these faults, which will help identify them. For the identification of faults, the frequency spectrum is considered as there are distinctive frequency components present in the spectrum for faults that are not presented normally. The samples are recorded in the time domain, so first of all, they are converted to the frequency domain using a fast Fourier transform (FFT). The processed data still undergoes some pre-processing and normalization before it is passed onto the machine learning algorithm for training.

Here, the training data sets are divided into two categories: synthetic signals and real-time signals. Both consist of around 14.4 million training data points with a sampling frequency of 20k Hz and a blind validation set of 1.4 million. The validation set consists of real-time signals and is compared and used after conversion to the frequency domain and pre-processing. As this is still under consideration, here the training sets considered are divided into four classifications and labeled as ’0’ in case of healthy, ’1’ in case of one broken bar, ’2’ in case of two broken bars and ’3’ in case of three broken bars. Hence, the data consists of one healthy case and three faulty cases for both synthetic signals and signals collected from the experimental setup. The frequency spectrum for healthy and faulty cases in a logarithmic scale for the real signal is shown in Figure 4.2.

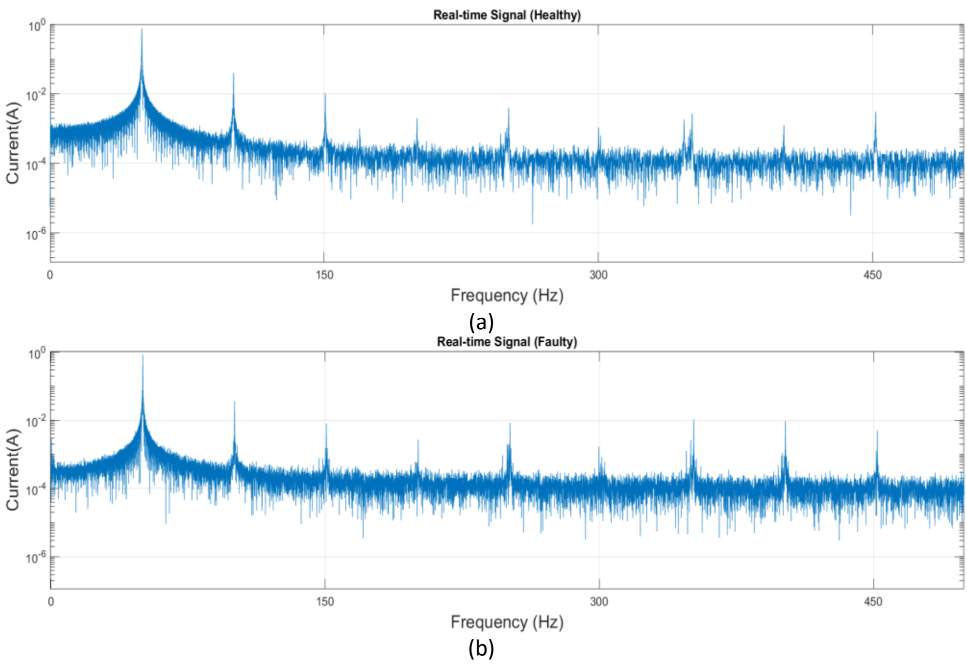


Figure 4.2. (a) FFT of one of the healthy real signal datasets (b) FFT of one of the faulty real signal datasets used for training

4.1.2 Synthetic Signal Results and Summary

As seen from the samples above, distinctive frequency components are present in the faulty spectrum, which are not usually present in the healthy one. The case is similar for synthetic signals, which are further smoothened out and processed with filters to only consider the most prominent frequency components for training purposes. Here, for comparison, five different types of neural network algorithms are considered to check the accuracy of both cases. Table 4.2 shows the accuracy of the synthetic signal approach when tested with real-time signals.

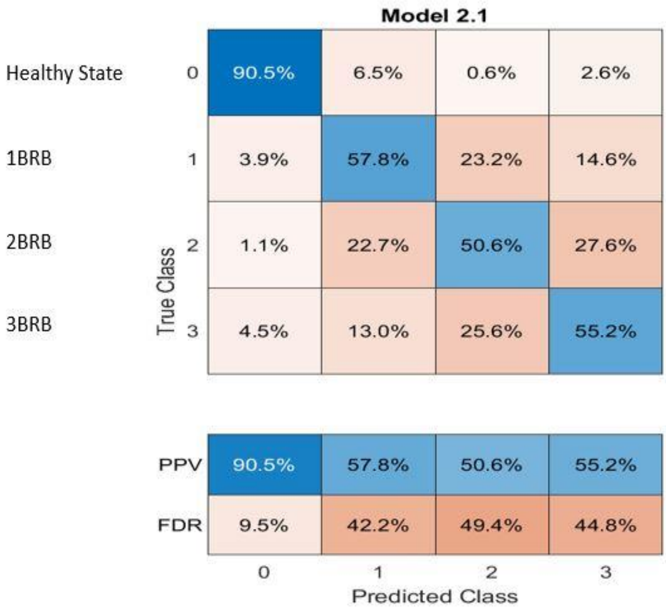
Table 4.2. Comparison between results of synthetic and original signals trained model

Machine Learning Model	Accuracy	
	Real Signals	Synthetic Signals
Narrow neural network	63.60 %	44.20 %
Medium neural network	74.20 %	59.80 %
Wide neural network	70.80 %	66.70 %
Bilayered neural network	69.90 %	64.70 %
Trilayered neural network	70.70 %	63.60 %

As it can be seen from Table 4.2 that the accuracy of the trained model is still lacking. To check the accuracy and working of the synthetic signals, the results are compared with the current approach, i.e., utilizing real-time signals for the training model. Table 4.2 summarizes the accuracy for both cases with the synthetic signal approach and the modern approach of using real-time signals. The accuracy can be further improved by considering slotting and inherent eccentricity-based harmonics.

Here, the training data sets are divided into two categories: the synthetic signals and the real-time signals. Both consist of around 14.4 million training data points with a sampling frequency of 20k Hz and a blind validation set of 1.4 million. For a detailed comparison between the two cases, validation results for both cases of Narrow Neural Network is shown in Figure 4.3. The healthy state is labeled as 0, 1BRB as 1, 2BRB as 2 and 3BRB as 3 for the training of neural network model.

It can be seen that for real signals, at least for healthy case the correct detection is 90%+, which falls for other cases mainly because of multiple label classification and insufficient dataset. Similarly, for synthetic signals, the detection is less than or around 50.0% in all cases except one, which is very low. This simplifies that if further improvement can be done, it is possible to get similar or better results than real signals. However, more research should be done by increasing the number of training datasets and gathering data from multiple machines for similar faults.



(a)

(b)

Figure 4.3. Validation results of narrow neural network in the case of (a) real signals (b) synthetic signals

4.2 Frequency Spectrum Resolution Improvement

The smart sensor-based low-power data acquisition and processing devices such as Arduino cards are increasing due to the increasing trend of the internet of things (IoT), cloud computation and other Industry 4.0 standards. For predictive maintenance, the fault representing frequencies at the incipient stage is very difficult to detect due to their small amplitude and spectral leakage of powerful frequency components. For this purpose, offline advanced signal processing techniques are used that are not possible in small signal processing devices due to the required computational time, complexity, and memory. This section an algorithm which can improve the spectrum resolution without complex advanced signal processing techniques and is suitable for low-power signal processing devices.

The accuracy and maturity of AI algorithms depend on the data size and its variety under different loading and faulty conditions. Thanks to the research in the field of mathematical modelling, data collection under different faulty and loading conditions for a variety of different machines is possible using simulations. Moreover, the data storage on the cloud can increase the training data set every day. The common point in all conventional and advanced techniques is the input signal. Mostly the global signals remain the same for all types of machines as the state variables of all machines are almost the same. Now the paradigm shift is the measurement of all those signals using low-cost data acquisition devices such as Arduino cards and sending the data in the database without loss or any additional infiltrations such as noise.

4.2.1 The Effect of Discontinuities in the Signal

Although FFT is a very powerful tool that is extensively used in the field of signal processing, for smooth, periodic, uniformly sampled points and long signals, FFT no doubt gives accurate results. However, the results become significantly erroneous if there are singularities or discontinuities in the signals. Thanks to the symmetrical and sinusoidal distributed design and performance parameters of electrical machines, almost all global signals such as current, voltage, and flux are periodic. The data discontinuities are however possible due to the limitations of the data acquisition devices, particularly if those are low power cards. This can be because of network limitations such as delay or loss of data transfer from the device to cloud. Because of the high sample rate, there is a high chance of data loss while data is being transferred from sensors to the low power cards. This is mostly because of the delay in the clearance of the buffers when data are being transmitted for a long time, i.e., a couple of days to weeks. The flow chart for such a setup is shown in Figure 4.4.

Data loss can occur in two scenarios, while the data are being transferred from sensors to the low powered cards and the other while the data are being transferred from the cards to cloud. The protocols used for data transmission have their own limitations too. The loss of data during transmission can be due to the limitation of network or delay/loss of network while transferring. Another reason might be due to the buffers being overloaded and not being properly cleared up before the next data come in, which can result in a loss of data while in transmission. These sharp changes in the signal are the potential cause of hiding the low power fault-based frequencies due to the increased spectral leakage of significant harmonics. It also decreases the computational time of FFT, decreases its efficiency, and increases the need for increased data length.

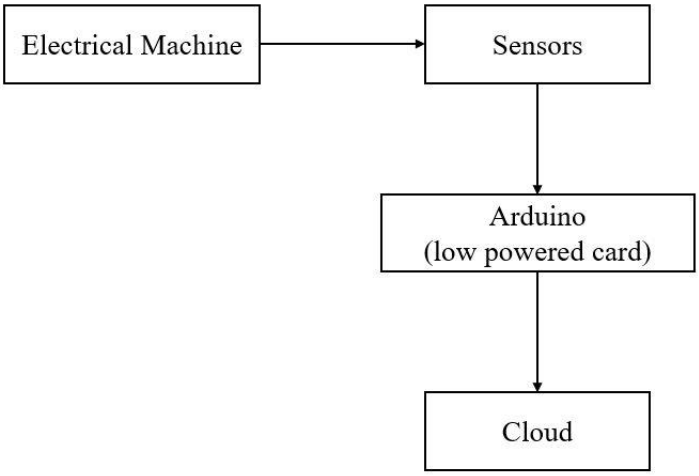


Figure 4.4. Flow chart for the data acquisition setup (previously published in article II)

The setup was run continuously for multiple days with different sampling rates to generate data losses. At higher sampling rates, the data losses occurred more often as the buffer became overloaded. Because of the limitation of the processing power of Arduino (low powered cards), data loss became inevitable in these cases. This is why the sampling rate tended to be on the lower side in most cases, but this also resulted in the samples being too low and similar data loss issues could occur if it kept running for a more extended period. The other scenario was also created by interrupting the network connection. In this case, wi-fi was used to transmit data from Arduino to the cloud database. Upon interruption of the network, as no data were transmitted, this resulted in data being lost. For some protocols, it could result in a delay at the receiving end, but this will still have components lost for the received signal. The setup was used to obtain signals with data discontinuity to check the result of the proposed algorithm.

The data discontinuities are detected by making a moving subtraction filter. The amplitude difference of every two consecutive samples defines the magnitude of discontinuity in them. For example, in the test case nine discontinuities along with their amplitude are discovered which needs correction.

$$diff = |x[n]| - |x[n-1]| \tag{4.2}$$

For correction the discontinuous sample is replaced with the average value of the samples x[n-1] and x[n+1];

$$\hat{x}[n] = \frac{|x[n+1]| + |x[n-1]|}{2} \tag{4.3}$$

The integer number of cycles can be calculated using zero cross detection as shown in Figure 4.5 below but in that case wrong computation can be occurred if there is any data discontinuity in the signal.

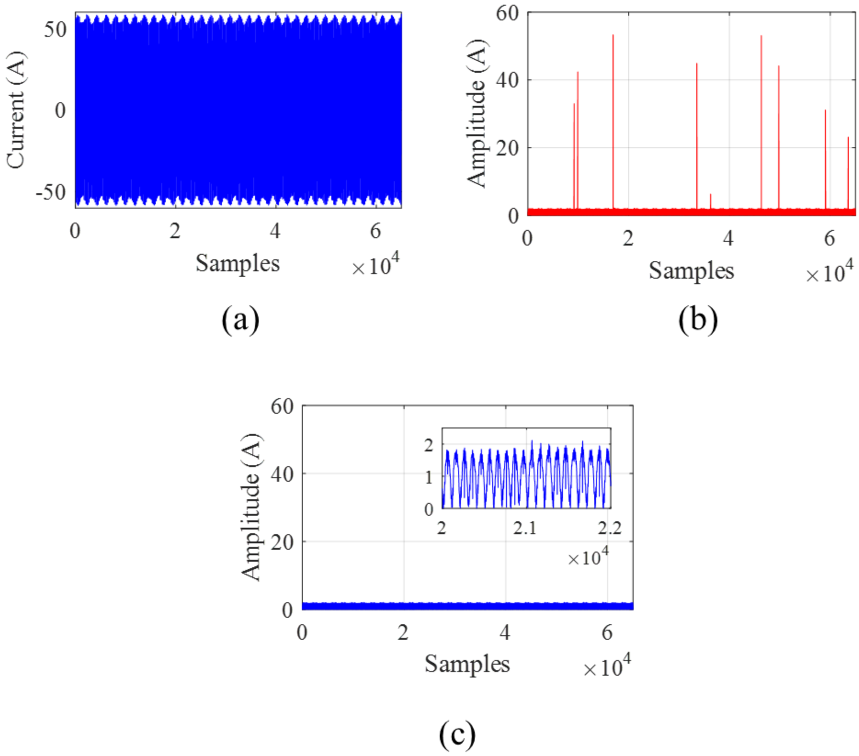


Figure 4.5. (a) The acquired stator current, (b) the result of moving subtraction filter for the detection of discontinuities, (c) after the correction of discontinuous samples (previously published in article II)

4.2.2 Algorithm for Spectrum Improvement

Data discontinuities are detected in the samples and replaced using mean value interpolation. After counting the integral number of cycles the sampling rate is improved and fractional cycles at both ends are discarded. This will give a signal from first zero crossing till last. The detailed steps for the implementation of the algorithm are shown in Figure 4.6.

4.2.3 Results & Summary

The algorithm is first implemented on the FEM based simulation signals with low sampling frequency. In Figure 4.7, it can be seen that even at high step size with a sampling frequency of 4kHz, the spectrum with counting the integral number of cycles increases the resolution significantly without the need of any truncating window. Moreover, the effect of communication channel-based data discontinuities and their correction is shown in Figure 4.8.

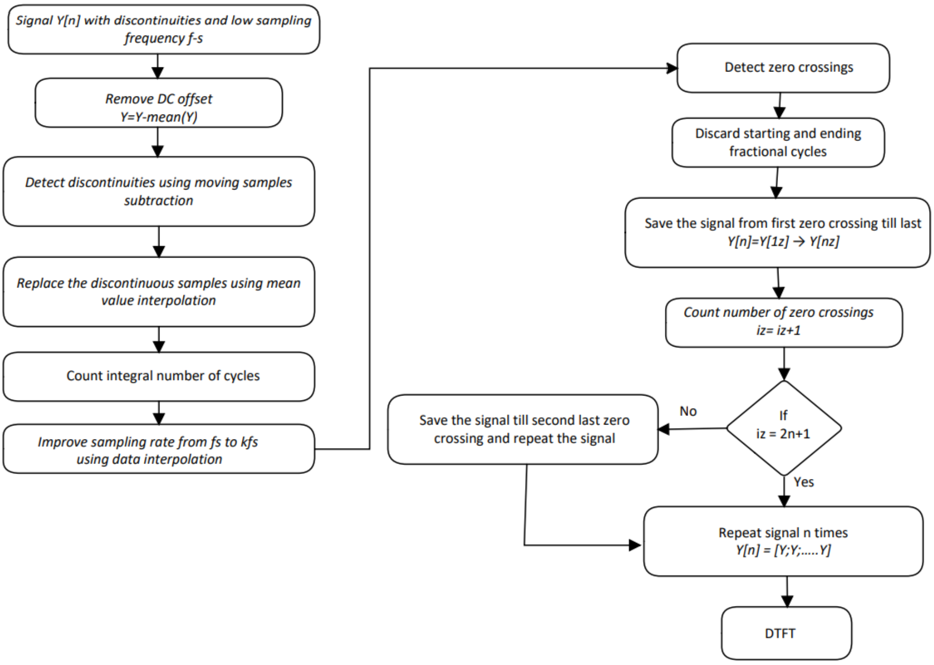


Figure 4.6. The algorithm for the counting the integral number of cycles, removal of signal discontinuities and fractional parts of the signal, data interpolation and repetition if necessary (previously published in article II)

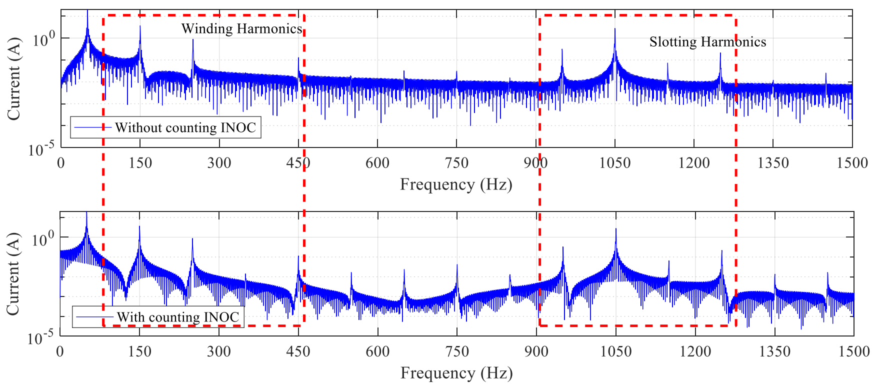


Figure 4.7. The simulated stator current spectrum showing stator winding and slotting harmonics before and after counting integral number of cycles (INOC) (previously published in article II)

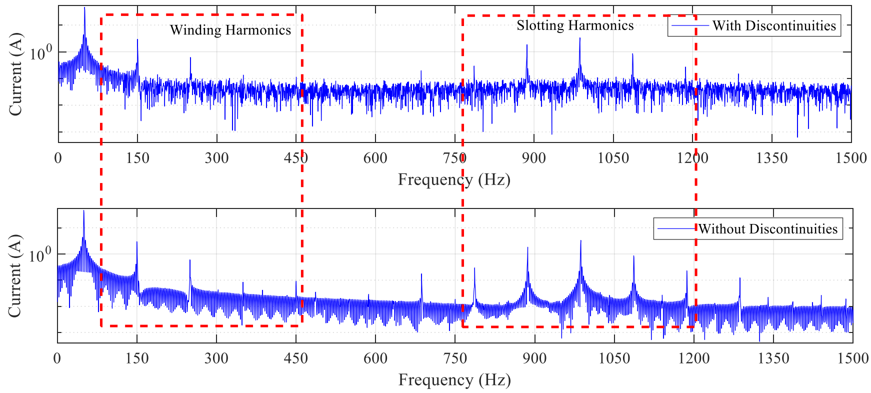


Figure 4.8. The effect of signal discontinuities on the spectrum resolution (previously published in article II)

For better comparison, it was then implemented on a practical setup consisting of investigations two similar machines are back-to-back connected. One machine works as loading machine while other is used as testing motor where the healthy and broken rotor bar carrying rotor are tested. Figure 4.9 and 4.10 shows the improvement in the spectrum resolution by removing the fractional parts of the signal and data discontinuities without any truncating window. The tiny broken rotor bar harmonics near the strong supply and spatial harmonics becomes well legible.

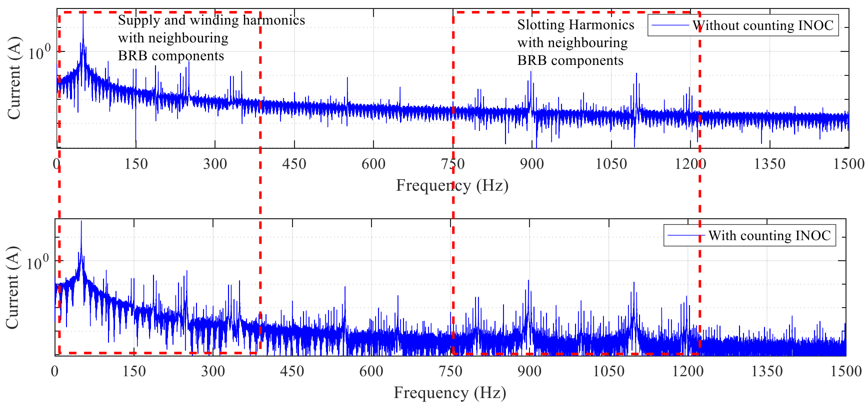


Figure 4.9. The practical stator current spectrum showing stator winding, slotting and broken rotor bar-based harmonics before and after counting integral number of cycles (INOC) (previously published in article II)

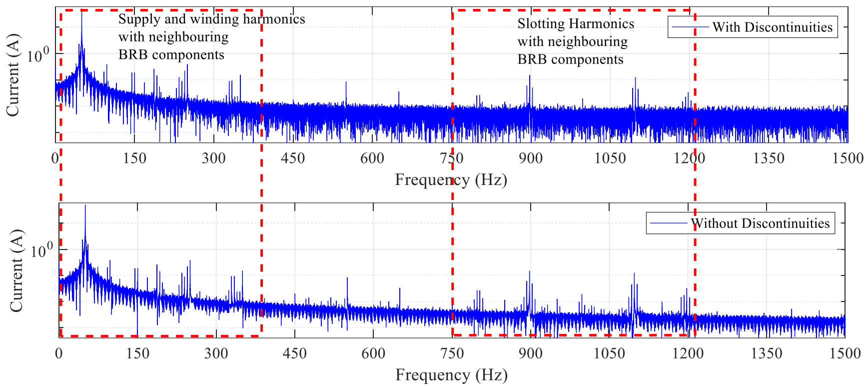


Figure 4.10. The practical stator current spectrum showing stator winding, slotting and broken rotor bar-based harmonics with and without discontinuities (previously published in article II)

4.3 Custom Neural Network Algorithms

This section presents a simplified machine learning approach to neural networks with two variations to work on the fault diagnostics of the electrical machine. This is a more straightforward approach to get the result in a minimum number of layers and with higher accuracy in the shortest time possible. This is also to check if the results can be generated with higher accuracy using a small number of datasets. The first layer of the neural network consists of a dot product of weights and the incoming inputs, whereas the second layer is different for both variations. One variation includes a sigmoid function on the second layer, whereas the second variation has a hyperbolic tangent function present. Both are given in Eq. 4.4 and Eq. 4.5, respectively as previously published in article IV.

$$f(x) = \frac{1}{1+e^{-x}} \tag{4.4}$$

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{4.5}$$

The variations are trained separately using 10,000 training data sets and then the results are validated with a different set of 500 validation data sets. The model trained is still under development to improve their results consistency as it may vary when the model is retrained. The models are trained using Python with a custom neural network defined class and without the use of tensor flow or any other third-party library. The flowchart for the working and training of the neural network algorithm is shown in Figure 4.11.

4.3.1 Custom Neural Network Results & Summary

The error is calculated after predicting the values after layer two and the weights are adjusted accordingly in layer 1 for the following input. The training undergoes 10,000 iterations for the test data set and a mean square error for the trained model is generated for both sigmoid and hyperbolic tangent variations. The results for the mean square error for both variations are shown in Figure 4.12.

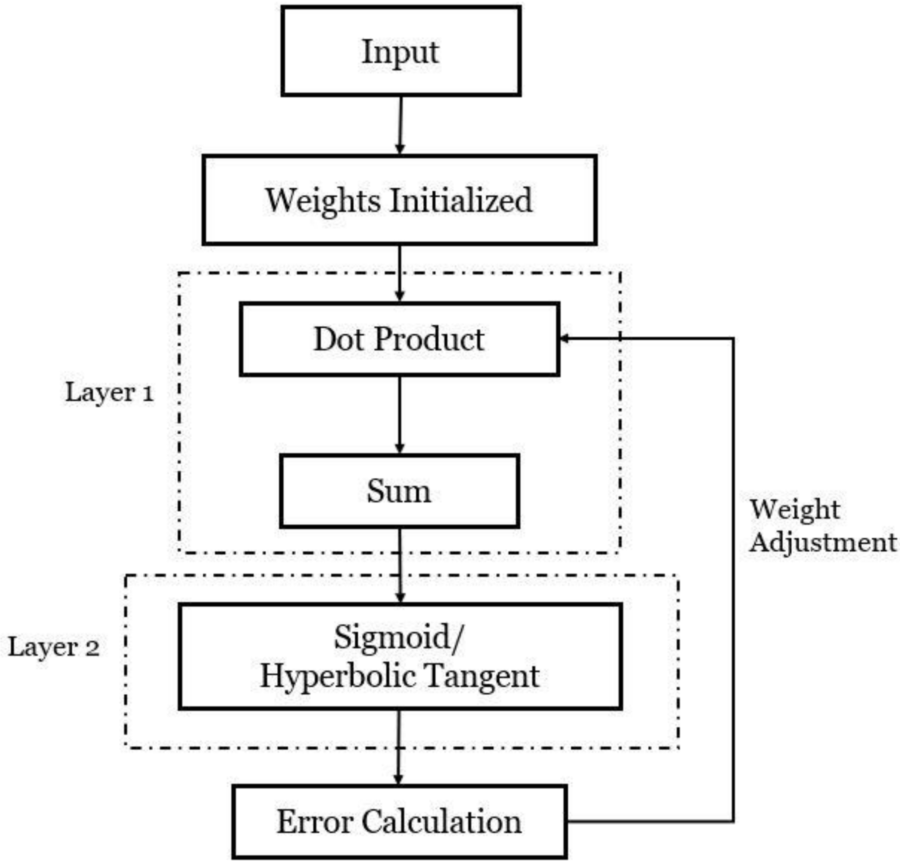
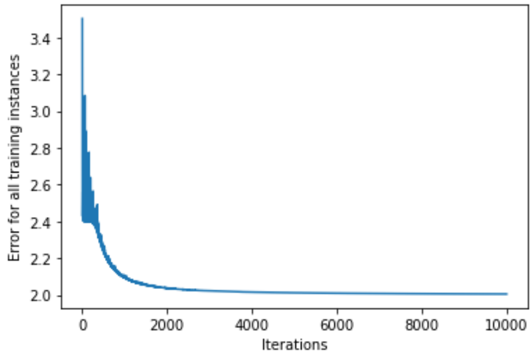
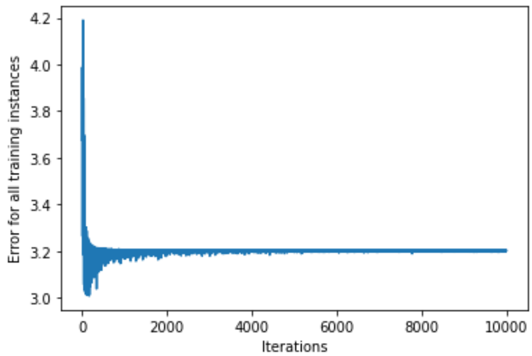


Figure 4.11. Overview of training of machine learning model (previously published in article IV)

From the above graphs, it can be seen that the model trained using the Sigmoid function has a low Mean Square Error (MSE) compared with the one trained using Hyperbolic Tangent Function. From the graphs of training algorithms, it is seen that the model with the Sigmoid function is better off than the one with Hyperbolic Tangent Function. For validation of the results, a validation data set for 500 sets was used and the comparison results for both are summarized in Table 4.3.



(a)



(b)

Figure 4.12. Mean square error (MSE) of model trained with (a) sigmoid function (b) hyperbolic tangent function (previously published in article IV)

Table 4.3. Comparison between sigmoid and hyperbolic tangent functions (previously published in article IV)

	Sigmoid	**Hyperbolic Tangent**
Mean square error (MSE)	1.8	3.17
Accuracy	80.41 %	65.48 %

Artificial Intelligence is taking its place in different research fields and is advancing at a rapid rate. Although there are a lot of machine learning algorithms present for classification, there is not a specific one that can be altered to one domain. This paper has presented two variations for a simplified machine learning algorithm with two layers that are a work in progress but target electrical machines' diagnostics. The proposed algorithms are simple and easy to train; they do not need a heavy system or wait for hours to get the trained model. Among the two variations of the presented approach, the one with the sigmoid function outperformed the hyperbolic tangent function in terms of performance.

The study is still a work in progress with future works, including the enhancement of neural networks to multi-layered algorithms with testing of different functions and having a consistent result in return. The future works also include further catering it towards fault diagnostics of electrical machines and adding some preprocessing, so it can detect and predict specific faults for electrical machines at the start. This will be further enhanced to make it a more generic approach.

4.4 Chapter Summary

This chapter covers the generation of synthetic signals using improvised statistical equations, which can be used to train models for fault detection. This can also help compensate for the lack of data as training of machine learning models requires big data sets. The research shows promising results, as the comparative analysis with models trained using measured signals is good. The accuracy of the trained models using synthetic signals can be further improved by generating data sets with more combinations. The chapter also covers the method for resolution improvement of the frequency spectrum because of discontinuities. This will compensate for the loss of data while transmission of signal and improve signal strength. Both of these methods will help generate data sets with improved quality for the training of machine learning models. The chapter also covers a comparative analysis of custom machine learning models based on different activation functions to check their validity. It can be seen from the results that it is possible to implement a simplified custom machine learning model that is solely focused on electrical machines faults and can help reduce training times and enhance accuracy for fault detection.

5 Predictive Fault Detection Algorithm for Electrical Machines

One of the most crucial parts of predictive maintenance or condition monitoring [108] is the implementation of machine learning based trained models. The accuracy of these models depends mainly on the quality of data that is being used for training and its diversity. Hence, it is always instructed to utilize high quality data samples with all possible scenarios and an optimal number of features to get better results. Artificial Neural Networks (ANNs) are the most commonly used models for fault detection classification. Although other offline models are available for the diagnostics of electrical machines like FEM [109,110], they are primarily for offline implementation and take much more time, which is unsuitable for real-time detection. ANNs models are usually trained with high processing power systems or cloud systems [111,112] to get minimum training time. However, there might be a chance that the model gets over-trained, so it is always good to check out the optimal number of samples needed and optimize the machine learning algorithm.

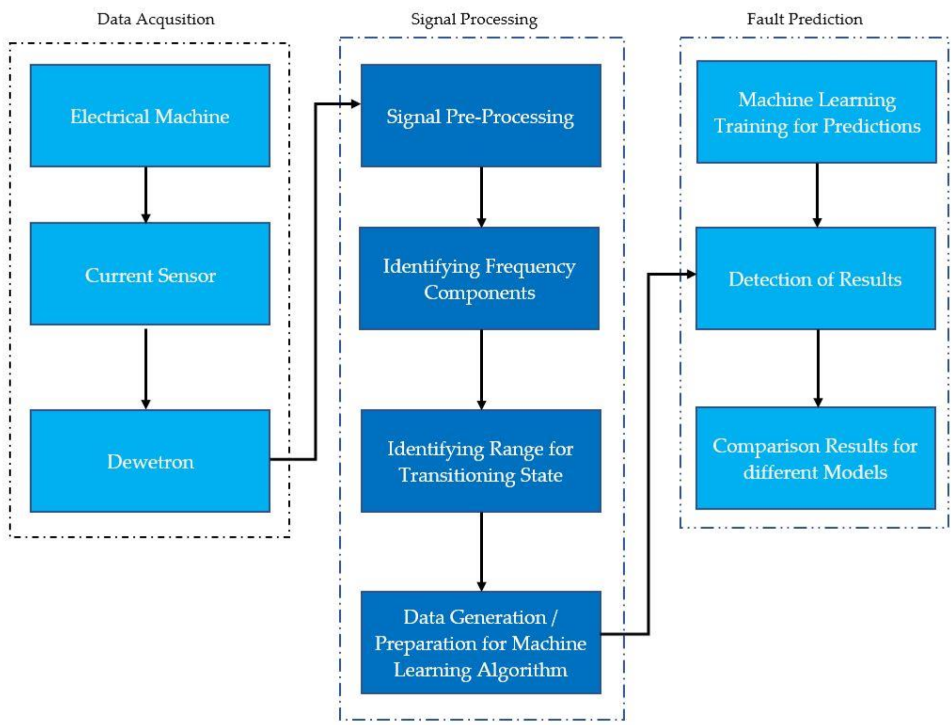


Figure 5.1. General overview of the proposed method (previously published in article I)

Much work is being done on the condition monitoring of electrical machines and fault detection in electrical machines and bearings. However, not much work is being done in the field of fault prediction in electrical machines. Some researchers are working in this domain, but some are utilizing the time domain while other systems are still being developed or are primarily for offline analysis and diagnostics. Some of the other systems used in this domain are only utilizing fault detection while gathering data for training, or there are some commercialized products available, but they are not only expensive, but their technology is still not disclosed. This section of the article can be divided into three parts: (i) Signal Processing, (ii) Data Preparation and (iii) Machine Learning Training and comparison between trained models. The general overview of the technique is shown in Figure 5.1.

5.1 Signal Processing

The first step is to process the gathered data and make it suitable for training. The data gathered through Dewetron is in the time domain, which is converted into the frequency domain using Fast Fourier Transform (FFT). The approach taken here is considering the current signature of the electrical machines and the effect of faults on them. A frequency spectrum of the current signal of electrical machine in logarithmic scale is shown in Figure 5.2.

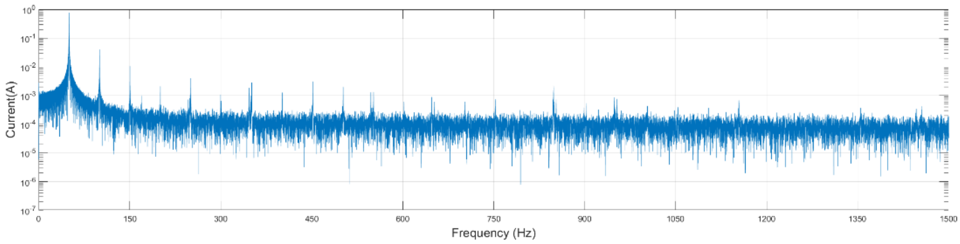


Figure 5.2. Frequency spectrum of current sample in logarithmic scale

As a wide range of frequency components are present in the spectrum, the first step is to identify the most prominent frequency components and filter out the negligible ones. This will give fewer frequency components and will make identifying distinct components easier. In this case, the frequency range was decided to be up to 500 Hz as, after this range, the frequency components amplitude is negligible and is not making any significant difference. The frequency spectrum after applying the low pass filter for up to 500 Hz is shown in Figure 5.3.

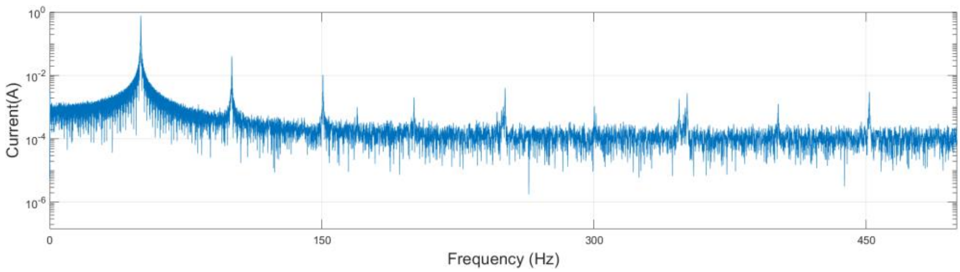
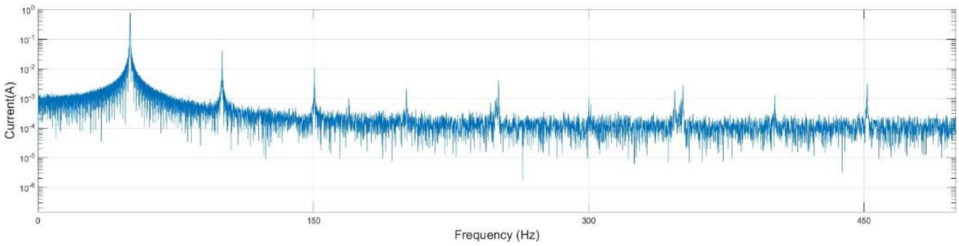
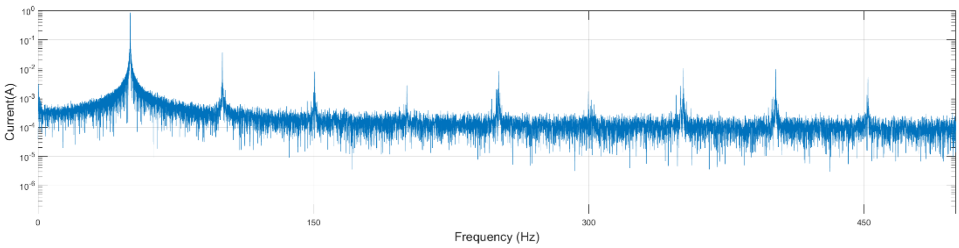


Figure 5.3. Narrowed down frequency spectrum example

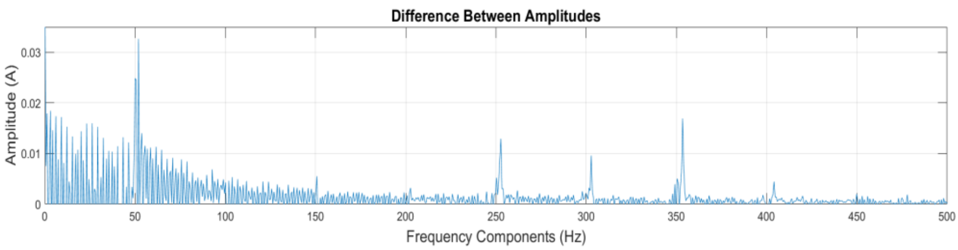
After narrowing down the frequency components used for training, a comparison is made between the healthy and faulty spectrums to narrow down the frequency components that make a difference. This comparison is carried out for multiple cases and data is collected from different induction motors to help identify the correct frequency components. General spectrums and their difference for one of the samples are shown in Figure 5.4. Once it is narrowed down, the most prominent components are selected to help determine the specific fault. This help simplifies the training more and these components will be used as one of the basics to complete combinations for the training of the predictive algorithm.



(a)



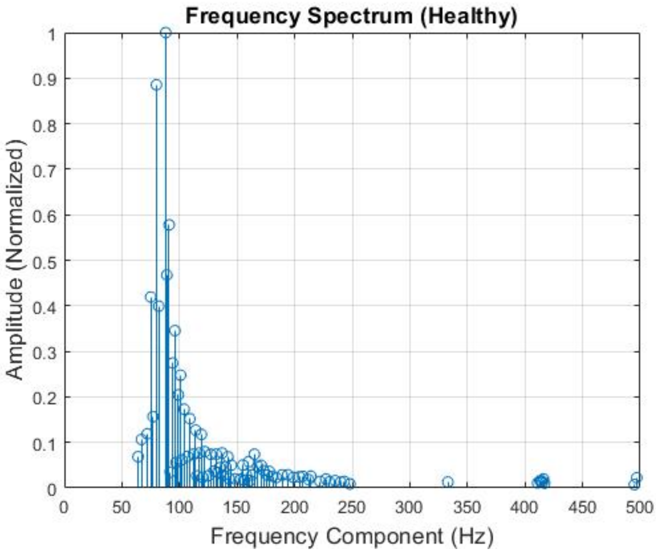
(b)



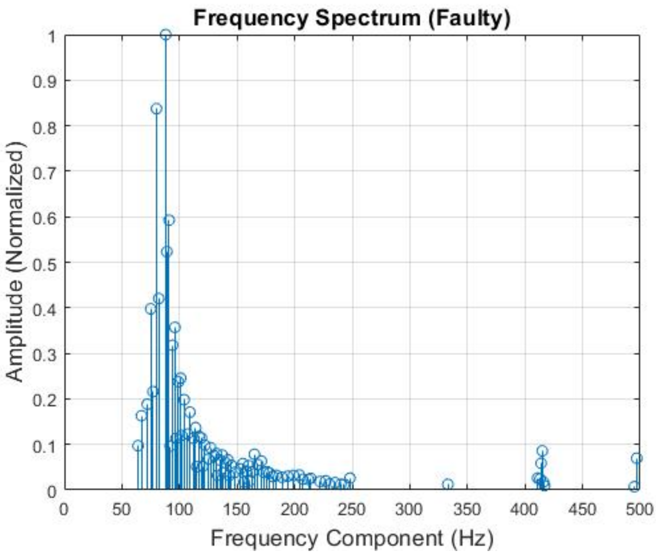
(c)

Figure 5.4. Frequency spectrum (a) healthy case (b) faulty case (c) difference to identify frequency components

The data is then further processed to check the amplitude of these frequency components in healthy and faulty cases. This process is carried out for multiple samples from different induction machines so that the identified components are universal for this specific fault. All amplitudes are normalized to lie between the range of 0 – 1 to get consistent results. Figure 5.5 shows an example of a frequency spectrum for both healthy and faulty cases with frequency components and amplitudes.



(a)



(b)

Figure 5.5. Frequency spectrum with normalized amplitudes (0 – 1) (a) healthy case (b) faulty case (previously published in article I)

After going through multiple samples, the changes in amplitude of the frequency components for both healthy and faulty are singled out. After careful analysis, the amplitude range for the prominent frequency components for fault occurrence is determined. Some of the frequency components with their amplitude range for the healthy case are shown in Table 5.1, whereas the faulty case is shown in Table 5.2. This range will help specify the fault occurrence probability for the predictive algorithm.

Table 5.1. Frequency amplitude range for fault occurrence (healthy signal) (previously published in article I)

Frequency component (Hz)	Minimum amplitude (A)	Maximum amplitude (A)
38.45 Hz	0.0031	0.0612
43.33 Hz	0.0022	0.0797
100 Hz	1.3004e-04	0.0460
125.73 Hz	7.5295e-05	0.0139
250.21 Hz	4.92e-05	0.0092
380.86 Hz	2.96e-06	8.56e-04
404.66 Hz	9.1363e-06	0.0049

Table 5.2. Frequency amplitude range for fault occurrence (faulty signal) (previously published in article I)

Frequency component (Hz)	Minimum amplitude (A)	Maximum amplitude (A)
38.45 Hz	4.72e-04	0.0633
43.33 Hz	6.8e-04	0.0841
100 Hz	8.6951e-05	0.0563
125.73 Hz	3.415e-04	0.0140
250.21 Hz	1.50e-04	0.0145
380.86 Hz	3.86e-05	0.0032
404.66 Hz	1.7432e-05	0.0083

After identifying the frequency components and their amplitude ranges for healthy and faulty cases, the trend of change in amplitude is noted. This will help to generate data and form combinations for the training of machine learning mode for fault prediction.

5.2 Faults Considered for Training

The faults considered for training of machine learning models are based on current signatures. As there are multiple phases of data being defined. After identifying the ranges and combinations, the next step is to prepare the data for training the machine learning model. In this case, the data gathered from the electrical machine is either for the healthy or faulty case. The combinations present are for either of the cases and there are no such data samples at this point that can predict the movement before fault occurrence or chances of fault occurrence. To compensate for this lack of data points in between, we will be using average to get the range value of range between the healthy and faulty cases. Eq. 5.1 depicts the calculation of frequency amplitudes for the case between healthy and faulty states.

$$y_t = \frac{(y_f + y_h)}{2} \tag{5.1}$$

here, y_t is the higher average amplitude of the frequency component when it is transitioning from a healthy to a faulty state. Let's say this is the transition state of the motor. Whereas y_h and y_f represents the maximum amplitude of the frequency component at healthy and faulty states, respectively. This will give the range of values for the frequency component amplitude between the transitioning state, which can be used further to determine which combinations can identify the faulty frequency components. A general overview of the amplitude range of different frequency components during the transitioning state is shown in Table 5.3.

Table 5.3. Frequency amplitude range for fault occurrence (transition state) (previously published in article I)

Frequency Component (Hz)	**Minimum Amplitude (A)**	**Maximum Amplitude (A)**
38.45 Hz	0.0612	0.0633
43.33 Hz	0.0797	0.0841
100 Hz	0.0460	0.0563
125.73 Hz	0.0139	0.0140
250.21 Hz	0.0092	0.0145
380.86 Hz	8.56e-04	0.0032

The difference in ranges for a specific frequency component is graphically shown in Figure 5.6. This gives an idea about the specific ranges needed to generate combinations for training a machine learning model as previously published in article I.

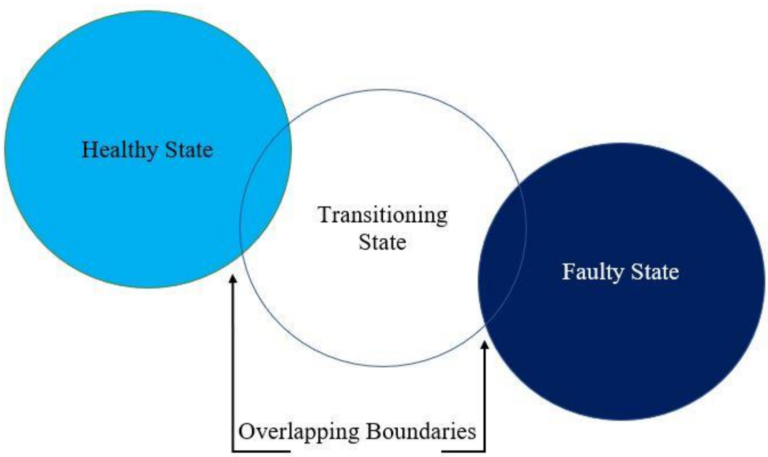


Figure 5.6. Overlapping Frequency Amplitude Range for Three states. (previously published in article I)

It can be seen that the ranges might overlap a bit, but it is either on the minimum side or maximum size of the transitioning state of the motor, which is to be expected. Once all the ranges of each considered frequency component are determined, the next step is to divide these ranges further into three parts, i.e., each part will be 30% of the range, excluding healthy and faulty areas. This will give us some idea about how much of a chance there is for the occurrence of a fault in the electrical machine. Table 5.4 shows the division of one of the frequency component amplitudes in the transition state.

Table 5.4. Division of transition state frequency component range for 250.21 Hz frequency component (previously published in article I)

Minimum Amplitude (A)	Maximum Amplitude (A)	Fault Occurrence Probability
0	0.00828	0%
0.00828	0.01052	30 %
0.01052	0.01185	60 %
0.01186	0.01317	90%
0.01317	-	100%

Table 5.4 shows the division of the range for one of the frequency component's amplitudes. This will help define the probability of fault occurrence in the incoming signal and will further enable to determine of the fault occurrence level. Once these are established, data points are generated based on these ranges, which will then be used for training the machine learning algorithm. Multiple combinations of these ranges are created to avoid missing out on any possible scenarios. The generated data is then combined with the data for healthy and faulty states for the specific frequency ranges and used for training the machine learning algorithm. The probability of fault occurrence is taken as an average with a weight of the ranges used in the combination. A weight is assigned based on the critical value of the frequency component amplitude and in which range it lies.

For example, for the initial combinations, the range values are the same, so similar weightage is applied to each range value to determine the probability of the fault occurrence and the average of those probabilities is taken. This will give us the same probability of the urgency of the fault that is occurring. The ranges are divided into five parts for simplicity, as shown in Table 5.4 above. The weightage assigned to each range is shown in Table 5.5. The final occurrence probability percentage is decided by taking the average for all frequency components, as shown in Eq. 5.2.

$$f_p = \frac{(a_1 + a_2 + a_3 + \cdots + a_n)}{n} \tag{5.2}$$

here, f_p is the probability of fault occurrence, whereas a_1 to a_n are the assigned probabilities to different frequency component amplitudes in the combination and n is the total number of frequency components present in the combination. This will give an average probability for the fault occurrence, which is then simplified on the basis of the division shown in Table 5.5.

Table 5.5. Division of transition state frequency component range for 250.21 Hz frequency component (previously published in article I)

Range Percentage of error	Weightage Assigned (0 – 1)	Range Percentage of error
0%	0	0%
30%	0.30	30%
60%	0.60	60%
90%	0.90	90%
100%	1.00	100%

This will determine the probability of fault occurrence in the electrical machine. However, for this research, the probability is rounded off to 0%, 30%, 60%, 90%, and 100% to classify the data for these 5 cases. However, they can be further divided into multiple options and a machine learning model can be trained based on it. Once the data is prepared, different machine learning models are used for training the sample data. The blind validation method was used to determine the accuracy of the trained model.

5.3 Predictive Model Results

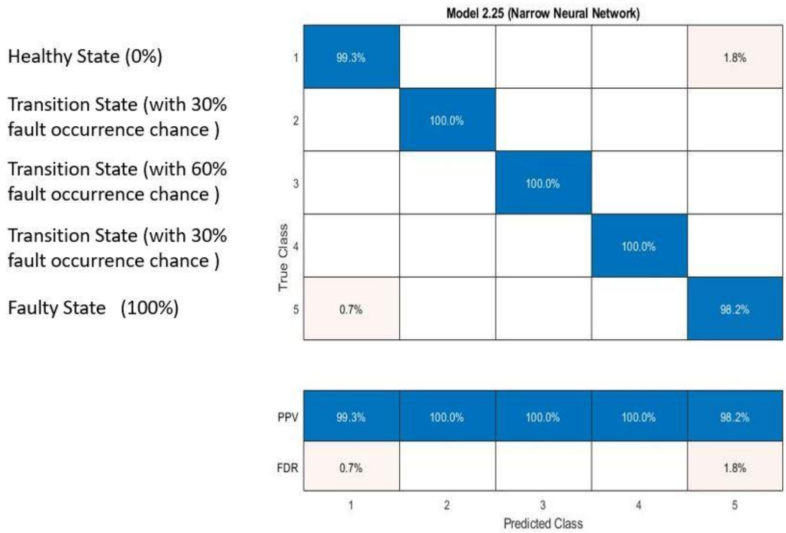
After completing the data points, the next step is to train machine learning models based on those data sets and validate the results to see if they predict the occurrence correctly or not. To ensure we cover all possible cases, healthy and faulty data points for validation were gathered from running electrical machines, whereas the data points for validation for transitioning state were randomly generated from the defined ranges. For comparison purposes, five different kinds of machine learning algorithms were selected and models were trained using those models.

For machine learning, different neural network algorithms were trained to compare purposes. As this is the initial stage of the proposed algorithm, the data points used for training the machine learning based models were around 68,000 data samples with a validation sample count of 6800 data samples. These initial tests were carried out on smaller data sets and might need to be tested for bigger data sample sets. Each range was assigned a classification label which is shown in Table 5.6.

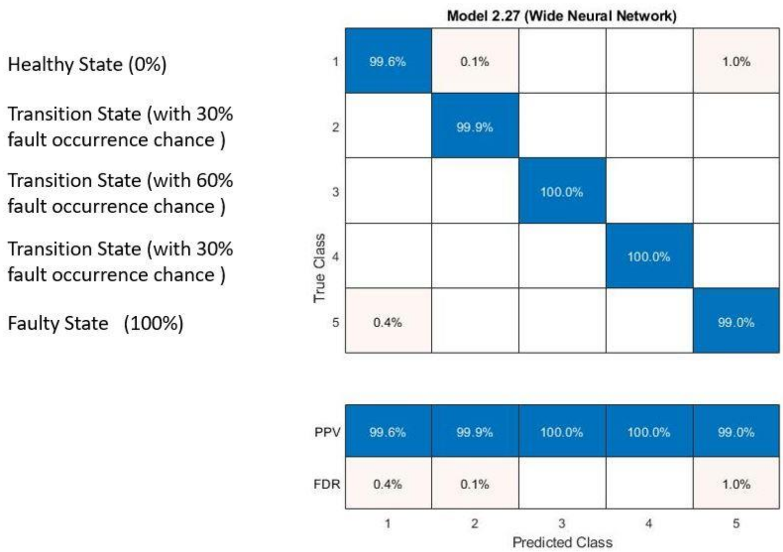
Table 5.6. Classification assigned per range of error (previously published in article I)

Range Percentage of error	Classification Label Assigned
0%	1
30%	2
60%	3
90%	4
100%	5

Machine Learning models were trained based on blind validation, i.e., the samples used for validation were not used for the training of the models. A total of eight models were considered, with the majority from neural networks for classification. A comprehensive comparison between the accuracies of these models is given in Table 5.7. The confusion matrix for the validation results of the two models is shown in Figure 5.7.



(a)



(b)

Figure 5.7. Machine learning results (previously published in article I) (a) narrow neural network (b) wide neural network

As can be seen from Table 5.7, all of the neural network techniques performed well, whereas the others were near. This might be because the data set is small and not too big; further experiments need to be done with bigger data sets to confirm results.

Table 5.7. Comparison results (previously published in article I)

Machine Learning Algorithm	Accuracy (validation)
Course tree	93.9 %
Gaussian naïve bayes	88.6 %
Fine KNN	97.1 %
Narrow neural network	99.3 %
Medium neural network	99.3 %
Wide neural network	99.6 %
Bilayered neural network	99.3 %
Trilayered neural network	99.1 %

5.4 Chapter Summary

This chapter proposes a novel technique for fault prediction of electrical machines. The technique is based on frequency spectrum fluctuations of the signal from electrical machines. As most of the electrical faults generate fluctuations in the current signature, this technique utilizes those fluctuations to train machine learning models for the prediction of electrical faults that occur in the incoming signals. This is a new concept and might need some more research to validate and improve the method further. The case study presents the result in the case of broken rotor bars; it can be seen from the results that it is indeed possible to predict the occurrence of faults in the electrical machine with higher accuracy based on the presented algorithm. The study shows promising results and predictive maintenance can be achieved using the proposed method. However, it would still be beneficial to test the algorithm with different complex combinations of faults and for different data sets to improve and validate its results.

6 Conclusion and Future Work

This chapter concludes the results of the research study based on the objectives. Moreover, some suggestions are added related to future work for the advancement of this study.

6.1 Conclusion & Summary

The main objective of this research was based on two points. The first was to develop a cost-efficient data acquisition system that can be utilized for remote locations and is easy to integrate. The second was to develop an algorithm for fault predictions in electrical machines based on current signatures.

The main emphasis on the data acquisition system was that it should be cost-efficient and reliable, with the ability to work in small places and remote locations. To achieve these conditions, a data acquisition system consisting of microcontroller cards along with the combination of Raspberry Pi was proposed so that there is a local backup of the collected data at Pi and it is possible to scale it easily in the future. The proposed system is also cost-efficient, scalable, reliable and flexible, as multiple microcontroller cards can be attached to the local node. Each microcontroller card can have multiple sensors attached to it; this can enable the formation of a cluster with one node acting as the control unit. Different IoT protocols were tested out to find the correct one and the data transmission protocol between the microcontroller card and Raspberry Pi was also verified over a period of weeks to make sure no data is being lost during transmission. For the research study, the data acquisition card is linked with the cloud and the data is synced online with the cloud database, which is then shown at the front end using a dashboard system.

The other part of the objective was to develop an algorithm for fault prediction of electrical machines, which does not take much time and could be implemented in real-time. To achieve this part, machine learning models were considered for training models to detect electrical machine faults. But as machine learning models require a large quantity of data to train more accurate models, the research was further expanded to see how the data can be generated using statistical equations. To reach this goal, statistical equations for different signatures frequencies of faults were considered, but as there was already broken rotor bars physical setup available in the lab, it was preferred over others. The study shows that models trained using synthetic signals give comparable results to that of models trained using real signals and if more data is generated covering all the possible values of prominent frequency components for fault, it might even be possible to give better results.

Another issue that came to light was the machine learning algorithm that was used for training, as different datasets with different fault combinations might give a different machine learning algorithm with higher accuracy. To resolve the issue, the study was taken for the development of a custom machine learning algorithm with different activation functions and different functions on hidden layers solely with a focus on electrical machine faults detection. The initial results of the study show promising results, with the accuracies still lower than that of the currently available algorithms, but the training time is lower or comparable. This study is still in its earlier phases and needs more time to develop.

On the signal processing side, for the development of an algorithm for the prediction of faults, the current signature spectrum was analyzed with the most prominent frequency components singled out that show fluctuations during fault occurrence. Those frequency components are then processed and utilized to determine a transit state between a healthy and faulty state which is used for training machine learning models to give a percentage prediction on the chances of fault occurrence in the incoming signal. The technique proposed gives promising results for multiple faults combination, but there is still a need to validate it further with more complex combinations and faults.

The test rig was used with broken rotor bars and bearing faults, where several experiments were performed with the motor running under different control environments and loads. The results for the data acquisition system were verified by using Dewetron in the test rig and for algorithms trained were compared with real signals that were needed for validation.

6.2 Future Work

The proposed model can be further validated with different combinations of faults and can be shifted from in lab setup to the industrial environment. Especially the data acquisition setup can be extended to implement hardware on-premises trained models for real-time detection and prediction of electrical machine faults.

The trained models also need to be validated for bigger data sets and more complex combinations of faults and in the industrial environment. The synthetic signals method can open up new ways for the generation of faulty data for different faults of electrical machines without much expense and further synthetic signals for different faults should be generated for validation of the method. It might open up a whole new way for the industry to get better trained models in less time and cut down the hassle of setting up a physical setup for data gathering.

The custom machine learning model should be studied and developed more as it can provide a consistent accuracy for faults involving electrical machines and yield a higher accuracy as it can be developed into a self-learning model. This will help the implementation of only one universal model in the industry with respect to fault detection and prediction in the electrical machine.

The prediction model based on spectrum analysis and machine learning can further be enhanced by validating it in the industrial environment and on larger datasets with multiple combinations of faults. It should also be tested out in real-time scenarios by implementing it on the node to see its performance for a longer period.

List of Figures

List of Tables

References

1. Ghobakhloo, M. Industry 4.0, Digitization, and Opportunities for Sustainability. *J. Clean. Prod.* 2020.
2. Wymore, M.L.; Van Dam, J.E.; Ceylan, H.; Qiao, D. A Survey of Health Monitoring Systems for Wind Turbines. *Renew. Sustain. Energy Rev.* **2015**, *52*, 976–990, doi:10.1016/j.rser.2015.07.110.
3. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Futur. Gener. Comput. Syst.* **2013**, *29*, 1645–1660, doi:10.1016/j.future.2013.01.010.
4. Khan, R.; Khan, S.U.; Zaheer, R.; Khan, S. Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges. In Proceedings of the Proceedings - 10th International Conference on Frontiers of Information Technology, FIT 2012; 2012.
5. Krco, S.; Pokric, B.; Carrez, F. Designing IoT Architecture(s): A European Perspective. *2014 IEEE World Forum Internet Things, WF-IoT 2014* **2014**, 79–84, doi:10.1109/WF-IoT.2014.6803124.
6. Shahinzadeh, H. IoT Architecture for Smart Grids. *2019 Int. Conf. Prot. Autom. Power Syst.* **2020**, 22–30.
7. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutorials* **2015**, *17*, 2347–2376, doi:10.1109/COMST.2015.2444095.
8. Divyashree, M. Internet of Things (IoT): A Survey. *2018 Int. Conf. Networking, Embed. Wirel. Syst.* **2020**, 1–6.
9. Wang, L.; Ann Alexander, C. Big Data Analytics in Biometrics and Healthcare. *J. Comput. Sci. Appl.* **2018**, *6*, 48–55, doi:10.12691/jcsa-6-1-7.
10. Sughasiny, M.; Rajeshwari, J. Application of Machine Learning Techniques, Big Data Analytics in Health Care Sector-A Literature Survey. *Proc. Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC 2018* **2019**, 741–749, doi:10.1109/I-SMAC.2018.8653654.
11. Rodriguez-Huerta, R.; Martinez-Castillo, J.; Morales-Gonzalez, E.; Herrera-May, A.L. Development of a Monitoring System for CO/CO2 with Android. *2019 IEEE Int. Conf. Eng. Veracruz, ICEV 2019* **2019**, doi:10.1109/ICEV.2019.8920673.
12. Jin, J.; Gubbi, J.; Marusic, S.; Palaniswami, M. An Information Framework for Creating a Smart City through Internet of Things. *IEEE Internet Things J.* **2014**, doi:10.1109/JIOT.2013.2296516.
13. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of Things for Smart Cities. *IEEE Internet Things J.* **2014**, doi:10.1109/JIOT.2014.2306328.
14. Potbhare, P.D.; Bhange, K.; Tembhare, G.; Agrawal, R.; Sorte, S.; Lokulwar, P. IoT Based Smart Air Pollution Monitoring System. *Proc. - Int. Conf. Appl. Artif. Intell. Comput. ICAAIC 2022* **2022**, 1829–1834, doi:10.1109/ICAAIC53929.2022.9792743.
15. MQTT Available online: https://mqtt.org/.
16. Yassein, M.B.; Shatnawi, M.Q.; Aljwarneh, S.; Al-Hatmi, R. Internet of Things: Survey and Open Issues of MQTT Protocol. In Proceedings of the Proceedings - 2017 International Conference on Engineering and MIS, ICEMIS 2017; 2018.

17. Hornsby, A.; Belimpasakis, P.; Defee, I. XMPP-Based Wireless Sensor Network and Its Integration into the Extended Home Environment. In Proceedings of the Digest of Technical Papers - IEEE International Conference on Consumer Electronics; 2009.
18. Iglesias-Urkia, M.; Orive, A.; Urbieta, A.; Casado-Mansilla, D. Analysis of CoAP Implementations for Industrial Internet of Things: A Survey. *J. Ambient Intell. Humaniz. Comput.* **2019**, doi:10.1007/s12652-018-0729-z.
19. Gregorio, F.; González, G.; Schmidt, C.; Cousseau, J. Internet of Things. In *Signals and Communication Technology*; 2020.
20. Raja, H.A.; Raval, H.; Vaimann, T.; Kallaste, A.; Rassolkin, A.; Belahcen, A. Cost-Efficient Real-Time Condition Monitoring and Fault Diagnostics System for BLDC Motor Using IoT and Machine Learning. *Diagnostika 2022 - 2022 Int. Conf. Diagnostics Electr. Eng. Proc.* **2022**, 21–24, doi:10.1109/Diagnostika55131.2022.9905102.
21. Hub, E. Types of Faults in Electrical Power Systems Available online: https://www.electronicshub.org/types-of-faults-in-electrical-power-systems/.
22. Chen, Y.K. Challenges and Opportunities of Internet of Things. In Proceedings of the Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC; 2012.
23. Wang, J.; Lim, M.K.; Wang, C.; Tseng, M.L. The Evolution of the Internet of Things (IoT) over the Past 20 Years. *Comput. Ind. Eng.* **2021**, *155*, 107174, doi:10.1016/j.cie.2021.107174.
24. Kwon, D.; Hodkiewicz, M.R.; Fan, J.; Shibutani, T.; Pecht, M.G. IoT-Based Prognostics and Systems Health Management for Industrial Applications. *IEEE Access* **2016**, *4*, 3659–3670, doi:10.1109/ACCESS.2016.2587754.
25. Sahoo, S.; Sahoo, J.; Kumar, S.; Lim, W.M.; Ameen, N. Distance Is No Longer a Barrier to Healthcare Services: Current State and Future Trends of Telehealth Research. *Internet Res.* 2023.
26. Li, S.; Tryfonas, T.; Li, H. The Internet of Things: A Security Point of View. *Internet Res.* **2016**, doi:10.1108/IntR-07-2014-0173.
27. Chen, D.; Bovornkeeratiroj, P.; Irwin, D.; Shenoy, P. Private Memoirs of IoT Devices: Safeguarding User Privacy in the IoT Era. *Proc. - Int. Conf. Distrib. Comput. Syst.* **2018**, *2018-July*, 1327–1336, doi:10.1109/ICDCS.2018.00133.
28. Compare, M.; Baraldi, P.; Zio, E. Challenges to IoT-Enabled Predictive Maintenance for Industry 4.0. *IEEE Internet Things J.* **2019**, *PP*, 1–1, doi:10.1109/jiot.2019.2957029.
29. Ayala, K. The 8051 Microcontroller : Architecture, Programming, and Applications , 2E. *Solutions* 1999.
30. Rajesh, A.S.; Prabhuswamy, M.S.; Krishnasamy, S. Smart Manufacturing through Machine Learning: A Review, Perspective, and Future Directions to the Machining Industry. *J. Eng. (United Kingdom)* **2022**, doi:10.1155/2022/9735862.
31. Wu, Z.; Qiu, K.; Zhang, J. A Smart Microcontroller Architecture for the Internet of Things. *Sensors (Switzerland)* **2020**, *20*, 1–17, doi:10.3390/s20071821.
32. Fran, M.; Anitha, S.; Mohan, R.R. IoT BASED WIND TURBINE MONITORING, FAULT DIAGNOSIS AND CONTROL USING UART. *Int. J. Adv. Res. Manag. Archit. Technol. Eng.* **2017**, *3*, 72–76.

33. Memala, W.A.; Bhuvaneswari, C.; Mana, S.C.; Selvan, M.P.; Maniraj, M.; Kishore, S. An Approach to Remote Condition Monitoring of Electrical Machines Based on IOT. In Proceedings of the Journal of Physics: Conference Series; 2021.
34. Arduino Uno Available online: https://store.arduino.cc/products/arduino-uno-rev3 (accessed on 3 October 2021).
35. Arduino Arduino Mega 2560 Rev3 Available online: https://store.arduino.cc/products/arduino-mega-2560-rev3.
36. Teensy 4.1 Available online: https://www.pjrc.com/store/teensy41.html (accessed on 5 October 2021).
37. Raspberry Pi Pico Available online: https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html (accessed on 7 November 2021).
38. ESP32 Dev Kits Available online: https://www.espressif.com/en/products/devkits (accessed on 25 April 2022).
39. STM32 Available online: https://www.st.com/en/microcontrollers-microprocessors/stm32f401re.html (accessed on 4 March 2022).
40. Raspberry Pi 4 Model B Available online: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/ (accessed on 9 November 2021).
41. BeagleBone Black Available online: https://beagleboard.org/black (accessed on 10 November 2022).
42. ESP8266 Available online: https://esp8266-shop.com/product/nodemcu-esp8266-esp-12e/ (accessed on 5 November 2022).
43. Adafruit Feather Available online: https://www.adafruit.com/product/3857 (accessed on 15 September 2022).
44. Particle Aragon Available online: https://docs.particle.io/argon/ (accessed on 25 November 2022).
45. Kiravuo, T.; Sarela, M.; Manner, J. A Survey of Ethernet LAN Security. *IEEE Commun. Surv. Tutorials* 2013.
46. Gadre, D. V.; Gupta, S.; Gadre, D. V.; Gupta, S. Universal Asynchronous Receiver and Transmitter (UART). In *Getting Started with Tiva ARM Cortex M4 Microcontrollers*; 2018.
47. Sayyed, M.A.; Ansari, N.; Mobeen, S.A. A Can Bus Based Fault Monitoring and Diagnosis System in Wind Turbine. **2014**, 56–58.
48. Al-Sarawi, S.; Anbar, M.; Alieyan, K.; Alzubaidi, M. Internet of Things (IoT) Communication Protocols: Review. In Proceedings of the ICIT 2017 - 8th International Conference on Information Technology, Proceedings; 2017.
49. ZigBee What Is ZigBee? | The ZigBee Alliance.
50. Rajaraman, V. Radio Frequency Identification. *Resonance* **2017**, doi:10.1007/s12045-017-0498-6.
51. Kuzlu, M.; Pipattanasomporn, M.; Rahman, S. Review of Communication Technologies for Smart Homes/Building Applications. In Proceedings of the Proceedings of the 2015 IEEE Innovative Smart Grid Technologies - Asia, ISGT ASIA 2015; 2016.
52. OASIS OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0. *OASIS Stand.* **2012**.
53. Andrew Banks, Ed Briggs, Ken Borgendale, R.G. MQTT Version 5.0. *oasis-open* **2017**.

54. Singh, M.; Rajan, M.A.; Shivraj, V.L.; Balamuralidhar, P. Secure MQTT for Internet of Things (IoT). In Proceedings of the Proceedings - 2015 5th International Conference on Communication Systems and Network Technologies, CSNT 2015; 2015.
55. Thota, P.; Kim, Y. Implementation and Comparison of M2M Protocols for Internet of Things. In Proceedings of the Proceedings - 4th International Conference on Applied Computing and Information Technology, 3rd International Conference on Computational Science/Intelligence and Applied Informatics, 1st International Conference on Big Data, Cloud Computing, Data Science; 2017.
56. Naik, N. Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP. In Proceedings of the 2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings; 2017.
57. Corno, F.; De Russis, L.; Saénz, J.P. Easing IoT Development for Novice Programmers through Code Recipes. In Proceedings of the Proceedings - International Conference on Software Engineering; 2018.
58. Raja, H.A.; Kudelina, K.; Asad, B.; Vaimann, T. Fault Detection and Predictive Maintenance for Electrical Machines. In *New Trends in Electric Machines - Technology and Applications*; IntechOpen, 2022.
59. Jiang, T.; Gradus, J.L.; Rosellini, A.J. Supervised Machine Learning: A Brief Primer. *Behav. Ther.* **2020**, doi:10.1016/j.beth.2020.05.002.
60. Tyagi, K.; Rane, C.; Manry, M. Supervised Learning. In *Artificial Intelligence and Machine Learning for EDGE Computing*; 2022 ISBN 9780128240540.
61. wu, yilu Linear Regression in Machine Learning.; 2022.
62. Bowlee, J. Logistic Regression for Machine Learning. *Mach. Learn. Mastery* **2016**.
63. Shubham, J. Decision Trees in Machine Learning. *Becom. Hum. - Explor. Artif. Intell. What it Means to be Hum.* **2018**.
64. Hang, Q.; Yang, J.; Xing, L. Diagnosis of Rolling Bearing Based on Classification for High Dimensional Unbalanced Data. *IEEE Access* **2019**, *7*, 79159–79172, doi:10.1109/ACCESS.2019.2919406.
65. Gutiérrez, L.; Patiño, J.; Duque-Grisales, E. A Comparison of the Performance of Supervised Learning Algorithms for Solar Power Prediction. *Energies* **2021**, *14*, 1–16, doi:10.3390/en14154424.
66. Great Learning Team Neural Networks Available online: https://www.mygreatlearning.com/blog/types-of-neural-networks/.
67. Cersonsky, R.K.; De, S. Unsupervised Learning. In *Quantum Chemistry in the Age of Machine Learning*; 2022 ISBN 9780323900492.
68. Usama, M.; Qadir, J.; Raza, A.; Arif, H.; Yau, K.L.A.; Elkhatib, Y.; Hussain, A.; Al-Fuqaha, A. Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges. *IEEE Access* **2019**, doi:10.1109/ACCESS.2019.2916648.
69. Gittler, T.; Scholze, S.; Rupenyan, A.; Wegener, K. Machine Tool Component Health Identification with Unsupervised Learning. *J. Manuf. Mater. Process.* **2020**, *4*, 86, doi:10.3390/jmmp4030086.
70. Glowacz, A.; Glowacz, Z. Diagnosis of the Three-Phase Induction Motor Using Thermal Imaging. *Infrared Phys. Technol.* **2017**, *81*, 7–16, doi:10.1016/j.infrared.2016.12.003.

71. Muja, M.; Lowe, D.G. Scalable Nearest Neighbor Algorithms for High Dimensional Data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2227–2240, doi:10.1109/TPAMI.2014.2321376.
72. Atanasov, N.; Zhekov, Z.; Grigorov, I.; Alexandrova, M. Application of Principal Component Analysis for Fault Detection of DC Motor Parameters. *Adv. Intell. Syst. Comput.* **2018**, *680*, 312–322, doi:10.1007/978-3-319-68324-9_34.
73. Mayerich, D.; Sun, R.; Guo, J. Deep Learning. In *Microscope Image Processing, Second Edition*; 2022 ISBN 9780128210499.
74. Janiesch, C.; Zschech, P.; Heinrich, K. Machine Learning and Deep Learning. *Electron. Mark.* **2021**, doi:10.1007/s12525-021-00475-2.
75. Florkowski, M. Classification of Partial Discharge Images Using Deep Convolutional Neural Networks. *Energies* **2020**, *13*, doi:10.3390/en13205496.
76. Tsantekidis, A.; Passalis, N.; Tefas, A. Recurrent Neural Networks. In *Deep Learning for Robot Perception and Cognition*; 2022 ISBN 9780323857871.
77. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Commun. ACM* **2020**, doi:10.1145/3422622.
78. Joshi, D.J.; Kale, I.; Gandewar, S.; Korate, O.; Patwari, D.; Patil, S. Reinforcement Learning: A Survey. In Proceedings of the Advances in Intelligent Systems and Computing; 2021.
79. Paliwal, M. Deep Reinforcement Learning. In Proceedings of the Smart Innovation, Systems and Technologies; 2022.
80. Jang, B.; Kim, M.; Harerimana, G.; Kim, J.W. Q-Learning Algorithms: A Comprehensive Classification and Applications. *IEEE Access* **2019**, doi:10.1109/ACCESS.2019.2941229.
81. Devgan, M.; Malik, G.; Sharma, D.K. Semi-Supervised Learning. In *Machine Learning and Big Data: Concepts, Algorithms, Tools and Applications*; 2020 ISBN 9781119654834.
82. Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; Tang, J. Self-Supervised Learning: Generative or Contrastive. *IEEE Trans. Knowl. Data Eng.* **2023**, doi:10.1109/TKDE.2021.3090866.
83. Maulud, D.; Abdulazeez, A.M. A Review on Linear Regression Comprehensive in Machine Learning. *J. Appl. Sci. Technol. Trends* **2020**, doi:10.38094/jastt1457.
84. Ohsaki, M.; Wang, P.; Matsuda, K.; Katagiri, S.; Watanabe, H.; Ralescu, A. Confusion-Matrix-Based Kernel Logistic Regression for Imbalanced Data Classification. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 1806–1819, doi:10.1109/TKDE.2017.2682249.
85. Charbuty, B.; Abdulazeez, A. Classification Based on Decision Tree Algorithm for Machine Learning. *J. Appl. Sci. Technol. Trends* **2021**, doi:10.38094/jastt20165.
86. Hsu, J.Y.; Wang, Y.F.; Lin, K.C.; Chen, M.Y.; Hsu, J.H.Y. Wind Turbine Fault Diagnosis and Predictive Maintenance through Statistical Process Control and Machine Learning. *IEEE Access* **2020**, *8*, 23427–23439, doi:10.1109/ACCESS.2020.2968615.
87. Kaparthi, S.; Bumblauskas, D. Designing Predictive Maintenance Systems Using Decision Tree-Based Machine Learning Techniques. *Int. J. Qual. Reliab. Manag.* **2020**, doi:10.1108/IJQRM-04-2019-0131.

88. Saberi, A.N.; Sandirasegaram, S.; Belahcen, A.; Vaimann, T.; Sobra, J. Multi-Sensor Fault Diagnosis of Induction Motors Using Random Forests and Support Vector Machine. *Int. Conf. Electr. Mach.* **2020**.
89. Ristin, M.; Guillaumin, M.; Gall, J.; Van Gool, L. Incremental Learning of Random Forests for Large-Scale Image Classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 490–503, doi:10.1109/TPAMI.2015.2459678.
90. Rohith Gandhi Support Vector Machine — Introduction to Machine Learning Algorithms. *Towar. Data Sci.* **2018**.
91. Wang, Q. Support Vector Machine Algorithm in Machine Learning. In Proceedings of the 2022 IEEE International Conference on Artificial Intelligence and Computer Applications, ICAICA 2022; 2022.
92. Enshaei, N.; Naderkhani, F. Application of Deep Learning for Fault Diagnostic in Induction Machine’s Bearings. *2019 IEEE Int. Conf. Progn. Heal. Manag. ICPHM 2019* **2019**, 1–7, doi:10.1109/ICPHM.2019.8819421.
93. Sun, S.; Przystupa, K.; Wei, M.; Yu, H.; Ye, Z.; Kochan, O. Fast Bearing Fault Diagnosis of Rolling Element Using Lévy Moth-Flame Optimization Algorithm and Naive Bayes. *Maint. Reliab.* **2020**, *22*, 730–740, doi:10.17531/ein.2020.4.17.
94. Chauhan, N. Naïve Bayes Algorithm: Everything You Need to Know. *KDnuggets* **2022**.
95. Jason Brownlee Naive Bayes for Machine Learning. *Naive Bayes Mach. Learn.* **2019**.
96. Zhang, Z. Introduction to Machine Learning: K-Nearest Neighbors. *Ann. Transl. Med.* **2016**, doi:10.21037/atm.2016.03.37.
97. javapoint K-Nearest Neighbor(KNN) Algorithm for Machine Learning.
98. Brownlee, J. A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning. *Mach. Learn. Mastery* **2016**.
99. Si, M.; Du, K. Development of a Predictive Emissions Model Using a Gradient Boosting Machine Learning Method. *Environ. Technol. Innov.* **2020**, doi:10.1016/j.eti.2020.101028.
100. Natekin, A.; Knoll, A. Gradient Boosting Machines, a Tutorial. *Front. Neurorobot.* **2013**, doi:10.3389/fnbot.2013.00021.
101. Mehlig, B. *Machine Learning with Neural Networks*; 2021;
102. Lyon, R.F. Neural Networks for Machine Learning. In *Human and Machine Hearing*; 2017.
103. Ali, A. Convolutional Neural Network(CNN) with Practical Implementation Available online: https://medium.com/machine-learning-researcher/convlutional-neural-network-cnn-2fc4faa7bb63.
104. Rahman, M.M.; Watanobe, Y.; Matsumoto, T.; Kiran, R.U.; Nakamura, K. Educational Data Mining to Support Programming Learning Using Problem-Solving Data. *IEEE Access* **2022**, doi:10.1109/ACCESS.2022.3157288.
105. Asad, B.; Vaimann, T.; Belahcen, A.; Kallaste, A. Broken Rotor Bar Fault Diagnostic of Inverter Fed Induction Motor Using FFT, Hilbert and Park’s Vector Approach. *Proc. - 2018 23rd Int. Conf. Electr. Mach. ICEM 2018* **2018**, 2352–2358, doi:10.1109/ICELMACH.2018.8506957.
106. Asad, B.; Vaimann, T.; Belahcen, A.; Kallaste, A.; Rassõlkin, A.; Naveed Iqbal, M. The Cluster Computation-Based Hybrid FEM – Analytical Model of Induction Motor for Fault Diagnostics. *Appl. Sci.* **2020**, *10*, 1–15, doi:10.3390/app10217572.

107. Forese, M. Talking with Turbines through the Internet of Things Available online: https://www.windpowerengineering.com/talking-turbines-internet-things/.
108. Wang, Q.; Wang, H.; Hou, L.; Yi, S. Overview of Tool Wear Monitoring Methods Based on Convolutional Neural Network. *Appl. Sci.* **2021**, *11*, doi:10.3390/app112412041.
109. Asad, B.; Vaimann, T.; Belahcen, A.; Kallaste, A.; Rassolkin, A. Rotor Fault Diagnostic of Inverter Fed Induction Motor Using Frequency Analysis. *Proc. 2019 IEEE 12th Int. Symp. Diagnostics Electr. Mach. Power Electron. Drives, SDEMPED 2019* **2019**, 127–133, doi:10.1109/DEMPED.2019.8864903.
110. Shen, H.; Li, Y.; Zhao, W.; Zhu, C.; Kwon, B. Il Design and Analysis of a High-Performance Outer Rotor Brushless DC Motor Using Loading Distribution Method for Range Hood Applications. *ICEMS 2018 - 2018 21st Int. Conf. Electr. Mach. Syst.* **2018**, 206–210, doi:10.23919/ICEMS.2018.8549348.
111. Adhikaree, A.; Kim, T.; Vagdoda, J.; Ochoa, A.; Hernandez, P.J.; Lee, Y. Cloud-Based Battery Condition Monitoring Platform for Large-Scale Lithium-Ion Battery Energy Storage Systems Using Internet-of-Things (IoT). *2017 IEEE Energy Convers. Congr. Expo. ECCE 2017* **2017**, *2017-Janua*, 1004–1009, doi:10.1109/ECCE.2017.8095896.
112. Ayad, S.; Terrissa, L.S.; Zerhouni, N. An IoT Approach for a Smart Maintenance. *2018 Int. Conf. Adv. Syst. Electr. Technol. IC_ASET 2018* **2018**, 210–214, doi:10.1109/ASET.2018.8379861.

Acknowledgements

First, I would like to thank all my friends, colleagues, and the staff in the Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology, for giving me all the support and encouragement during my doctoral study. Moreover, I would like to say special gratitude to my supervisor Dr. Toomas Vaimann for his unvarying guidance, creative ideas, and motivation.

Moreover, I would like to say special thanks to my family for their patience and untiring support during this time especially my beloved wife Farhat-ul-Ain for keeping up with me. This would not have been possible without their support, encouragement and patience.

Abstract

IoT based tools and methods for electrical machine diagnostics

The thesis aims to study the development of different cost-efficient tools and methods based on the Internet of Things (IoT) that can be utilized for the predictive maintenance of electrical machines, which should allow the implementation of diagnostics and prognostics of the electrical machine in real-time. The purpose of doing so is to develop a diagnostic system for remote or off-shore electrical machines so that the maintenance of such electrical machines can be shifted from scheduled to predictive maintenance. The second prime objective is to study different faults in the electrical machine that generate unique current signatures and develop an algorithm using signal processing and machine learning to detect and predict those faults at an early stage. Another prime objective is to explore ways to generate or gather more data for different faults to train machine learning algorithms, as building an in-lab setup with the faults physically inflicted is expensive and it takes time to gather different faulty signals in such a way.

The data acquisition system based on microcontroller cards was considered as a starting point. As the data acquisition cards available on the market are expensive and they need a system separately to be able to extract data using them and also most of the setups readily available were using either PLC or SCADA, microcontroller cards were considered based on their flexibility, scalability and reliability. Different microcontroller cards were compared based on their processing power and the ability to transmit data to the cloud for a longer period of time using IoT protocols, but they had a limitation with respect to data transmission as loss of data was observed based on interruptions or delays in the network. The data collected through the data acquisition system is also compared with the one collected through Dewetron to validate data accuracy and to make sure there is no loss of data.

To address this shortcoming of the system, Raspberry Pi was attached as a separate node with the microcontroller card and the data is now transmitted to the Pi, where it is stored in a local database as a backup and synced with the cloud in real-time. So even if there is a network interrupt, the data can still be recovered from the local database and synced with the cloud without any issue. This setup also gives the flexibility and processing power to implement a machine learning model on-premises in the future and can further be expanded into a cluster with one Raspberry Pi acting as the control unit for different nodes.

Moreover, another scope was to train machine learning models for fault detection of electrical machines. For this purpose, broken rotor bars and bearings faults were considered for the case study. Test rigs were implemented in the lab with the faults physically inflicted to gather data for the faulty state of the motor. The collected signal was pre-processed to remove any noise that was present and fast Fourier transform (FFT) was used to study the fluctuations of frequency components between healthy and faulty states. The most prominent frequency components are extracted and then utilized for the training of machine learning models for fault detection. The blind validation of machine learning models gives different models with higher accuracy based on the combination of faults. Thus, limiting the options for simpler combinations, as for complex combinations, the neural network gives the best accuracy but takes more time for training, similarly for recurring neural networks. This gave way to implement a simple

custom machine learning algorithm focused on electrical machine faults to help compare the accuracy with present algorithms. This study is still in development and is only in its early stage.

Moreover, another issue that is faced is gathering enough data for the training of machine learning algorithms. As the physical implementation of faults for in lab setup takes time and is expensive so it is not possible to implement each fault of the electrical machine in the physical setup. To address this issue, the method for the generation of synthetic signals using statistical equations is proposed. The statistical equation for signature faults of electrical machines is utilized to derive an improvised version of the statistical equation to generate synthetic signals, including those faults. Broken rotor bars (BRB) faults are considered for this study, the improvised statistical equations are used to generate synthetic signals for one, two and three broken rotor bars along with a healthy state. The trained model based on these synthetic signals is then used for blind validation for real signals and a comparison is also made with a model trained with real signals gathered from physical setup having BRB faults.

Moreover, the spectrum of current was further analyzed to develop an algorithm with signal processing and machine learning to give a probable prediction of fault occurrence in the electrical machine. The proposed method focuses on the most prominent frequency components that show fluctuations in their current spectrum in the presence of a fault and extract those features to define a transit state between a healthy and faulty state. This transit state is further divided into three parts to define the probability of fault occurrence in the electric machine. The algorithm is a step towards developing a self-learning algorithm that can be deployed on nodes to implement fault prediction at the hardware on-premises.

Lühikokkuvõte

Asjade interneti põhised tööriistad ja meetodid elektrimasinate diagnostikaks

Lõputöö eesmärk on uurida erinevate kuluefektiivsete vahendite ja meetodite väljatöötamist, mis põhinevad asjade internetil (IoT), mida saab kasutada elektrimasinate ennetavaks hoolduseks, mis peaks võimaldama elektrimasinate diagnostika ja prognostika rakendamist reaalajas. Selle eesmärk on töötada välja diagnostikasüsteem keeruliselt ligipääsetavate elektrimasinate jaoks, et selliste elektrimasinate hooldus saaks minna üle plaanipäraselt hooldamiselt prognoositavale hooldusele. Teine peamine eesmärk on uurida erinevaid elektrimasinas esinevaid rikkeid, mis tekitavad unikaalseid voolusignaale, ning töötada välja signaalitöötluse ja masinõppe abil algoritm, et neid rikkeid varakult avastada ja prognoosida. Samuti uuritakse võimalusi, kuidas luua või koguda rohkem andmeid erinevate rikete kohta, et treenida masinõppe algoritme, sest füüsiliselt tekitatud riketega laborisüsteemi loomine on kallis ja erinevate vigade signaalide kogumine sellisel viisil võtab aega.

Lähtepunktiks võeti mikrokontrolleri kaartidel põhinev andmete kogumise süsteem. Kuna turul saadaolevad andmekogumiskaardid on kallid ja nende abil andmete kogumiseks on vaja eraldi süsteemi ning enamik kergesti kättesaadavaid seadistusi kasutas kas PLC või SCADA lahendust, kaaluti mikrokontrolleri kaarte nende paindlikkuse, skaleeritavuse ja töökindluse alusel. Erinevaid mikrokontrolleri kaarte võrreldi nende töötlemisvõimsuse ja võime alusel edastada andmeid pilve pikema aja jooksul asjade interneti protokollide abil, kuid neil oli andmete edastamise osas piiranguid, kuna võrgus esinevate katkestuste või viivituste tõttu täheldati andmekaotust. Andmete kogumise süsteemi kaudu kogutud andmeid võrreldakse ka Dewetroni andmehõivesüsteemi kaudu kogutud andmetega, et kinnitada andmete täpsus ja veenduda, et andmeid ei ole kaduma läinud.

Selle süsteemi puuduse kõrvaldamiseks kinnitati Raspberry Pi eraldi sõlmpunktina koos mikrokontrolleri kaardiga ja andmed edastatakse nüüd Raspberry Pi’le, kus need salvestatakse varukoopiana kohalikku andmebaasi ja sünkroonitakse reaalajas pilvega. Nii et isegi kui võrgus tekib katkestus, saab andmed ikkagi taastada kohalikust andmebaasist ja sünkroonida pilvega ilma probleemideta. Selline ülesehitus annab ka paindlikkuse ja töötlemisvõimsuse, et tulevikus rakendada masinõppe mudelit kohapeal, ning seda saab veelgi laiendada klastriks, kus üks Raspberry Pi tegutseb eri sõlmede juhtimisüksusena.

Lisaks sellele oli teine eesmärk koolitada masinõppe mudeleid elektrimasinate rikete tuvastamiseks. Sel eesmärgil käsitleti juhtumiuuringu puhul purunenud rootori vardaid ja laagrite rikkeid. Rikked tekitati füüsiliselt laboris asuvatele katseseadmetele, et koguda andmeid mootori rikkeseisundi kohta. Kogutud signaali eeltöödeldi, et eemaldada müra, ja kasutati kiiret Fourier' teisendust (FFT), et uurida sageduskomponentide kõikumisi tervete ja riketega olukordade vahel. Kõige silmapaistvamad sageduskomponendid eraldati ja seejärel kasutati neid masinõppe mudelite treenimiseks rikke tuvastamiseks. Masinõppe mudelite pime-valideerimise tulemusel saadakse erinevad suurema täpsusega mudelid, mis põhinevad rikete kombinatsioonil. Seega on lihtsamate kombinatsioonide puhul tuvastamisvõimalused piiratumad, kuna keerukate kombinatsioonide puhul annab närvivõrk parema täpsuse, kuid võtab rohkem aega, sarnaselt korduvate närvivõrkude puhul, võtab õppimine rohkem aega. See annab

võimaluse koostada lihtne kohandatud masinõppe algoritm, mis keskendub elektrimasinate riketele, et võimaldada võrrelda täpsust praeguste algoritmidega. See uuring on veel algusjärgus.

Probleemiks osutub piisavate andmete kogumine masinõppe algoritmide treenimiseks. Kuna rikete füüsiline tekitamine laboris võtab aega ja on kallis, siis ei ole otstarbekas iga elektrimasinaga seotud riket füüsilises seadeldises rakendada. Selle probleemi lahendamiseks pakutakse välja meetod sünteetiliste signaalide genereerimiseks statistiliste võrrandite abil. Elektrimasinate rikkeid käsitlevat statistilist võrrandit kasutatakse statistilise võrrandi improviseeritud versiooni tuletamiseks, et luua sünteetilisi signaale, sealhulgas käsitletavate rikete puhul. Käesolevas uuringus võetakse arvesse rootorivarraste rikkeid, improviseeritud statistilisi võrrandeid kasutatakse sünteetiliste signaalide genereerimiseks ühe, kahe ja kolme katkise rootorivarda puhul ning riketeta rootori olukorras. Nende sünteetiliste signaalide põhjal koolitatud mudelit kasutatakse seejärel reaalsete signaalide pimedaks valideerimiseks ning seda võrreldakse mudeliga, mis on koolitatud reaalsete signaalidega, kogutud varraste riketega füüsilisest katseseadmest.

Lisaks analüüsiti täiendavalt vooluspektrit, et töötada välja signaalitöötluse ja masinõppe abil algoritm, mis annaks tõenäolise prognoosi rikke esinemise kohta elektrimasinas. Väljapakutud meetod keskendub kõige tugevamatele sageduskomponentidele, mis näitavad rikke esinemise korral vooluspektri kõikumisi, ja ekstraheerib need omadused, et määratleda ülemineku seisund terve ja rikke seisundi vahel. See ülemineku seisund jaguneb omakorda kolmeks osaks, et määratleda rikke esinemise tõenäosus elektrimasinas. Algoritm on samm iseõppiva algoritmi väljatöötamise suunas, mida saab kasutada sõlmedes, et rakendada rikete ennustamist riistvaras kohapeal.

Appendix

Publication I

Raja, H. A.; Kudelina, K.; Asad, B.; Vaimann, T.; Kallaste, A.; Rassõlkin, A.; Khang, H. V. (2022). Signal Spectrum-Based Machine Learning Approach for Fault Prediction and Maintenance of Electrical Machines. Energies, 15 (24), #9507. DOI: 10.3390/en15249507.

energies

Article

Signal Spectrum-Based Machine Learning Approach for Fault Prediction and Maintenance of Electrical Machines

Hadi Ashraf Raja [1], Karolina Kudelina [1,*], Bilal Asad [1], Toomas Vaimann [1], Ants Kallaste [1], Anton Rassõlkin [1] and Huynh Van Khang [2]

[1] Department of Electrical Power Engineering & Mechatronics, Tallinn University of Technology, 19086 Tallinn, Estonia

[2] Department of Engineering Sciences, University of Agder, 4604 Kristiansand, Norway

* Correspondence: karolina.kudelina@taltech.ee

Abstract: Industrial revolution 4.0 has enabled the advent of new technological advancements, including the introduction of information technology with physical devices. The implementation of information technology in industrial applications has helped streamline industrial processes and make them more cost-efficient. This combination of information technology and physical devices gave birth to smart devices, which opened up a new research area known as the Internet of Things (IoT). This has enabled researchers to help reduce downtime and maintenance costs by applying condition monitoring on electrical machines utilizing machine learning algorithms. Although the industry is trying to move from scheduled maintenance towards predictive maintenance, there is a significant lack of algorithms related to fault prediction of electrical machines. There is quite a lot of research going on in this area, but it is still underdeveloped and needs a lot more work. This paper presents a signal spectrum-based machine learning approach toward the fault prediction of electrical machines. The proposed method is a new approach to the predictive maintenance of electrical machines. This paper presents the details regarding the algorithm and then validates the accuracy against data collected from working electrical machines for both cases. A comparison is also presented at the end of multiple machine learning algorithms used for training based on this approach.

Keywords: artificial intelligence; fault prediction; predictive maintenance; machine learning; neural network

Citation: Raja, H.A.; Kudelina, K.; Asad, B.; Vaimann, T.; Kallaste, A.; Rassõlkin, A.; Khang, H.V. Signal Spectrum-Based Machine Learning Approach for Fault Prediction and Maintenance of Electrical Machines. *Energies* **2022**, *15*, 9507. https://doi.org/10.3390/en15249507

Academic Editor: Luigi Piegari

Received: 30 November 2022
Accepted: 12 December 2022
Published: 15 December 2022

1. Introduction

The advancement in information technology and its integration with other fields have opened up new research and development areas. This has also resulted in the revolution of industrial standards with the massive turn of events along with the introduction of industrial standards 4.0 [1]. This has helped industries streamline processes and make them more efficient and cost-effective. The integration of information technology with physical devices developed a whole new domain, which has become the core of the modern industry and resulted in the advent of smart devices. This has made it easier for industries to reduce shutdown times and help tackle issues in a more systematic way. Smart devices can communicate with each other over the internet and make decisions or pinpoint the issues in the system, which has made the industrial process smoother. This field of development of smart devices and their communication over a network (or internet) is commonly referred to as the Internet of Things (IoT) [2,3]. These devices can not only communicate with each other, but they also can act as end nodes for data collection from electrical machines using sensors. This process is known as condition monitoring of electrical machines as the data are transmitted in real-time through smart devices and monitored in real-time. This helps monitor the health of the electrical machine and reduces shutdown times in case of fault occurrence. This collected data can further be used for training models for real-time fault detection, fault prediction, data analysis, and diagnostics of the connected machine [4–7]. IoT has cemented its position in industrial applications, become one of the fundamental pillars of modern industry [8], and further enhanced its importance due to predictive maintenance research [9–13].

This advancement has helped the industry to switch from scheduled maintenance towards predictive maintenance and real-time condition monitoring of electrical machines. Condition monitoring of electrical machines is now a standard implementation in any modern industry [14,15]. Data collection for further data analysis is a core part of this system that helps develop predictive maintenance systems [10,16]. In general, electrical machine maintenance can be divided into four phases: periodic, proactive, reactive, and predictive maintenance. Most of the industry still works with periodic and reactive maintenance, i.e., scheduled maintenance or when a machine is down, as shown in Figure 1. There is an uptrend to move towards proactive and predictive maintenance as it is more cost-efficient and productive. This move is mainly possible because of the recent development in the field of IoT and research on the development of more accurate machine learning-trained models for electrical machines with high-quality data collected with the help of condition monitoring systems.

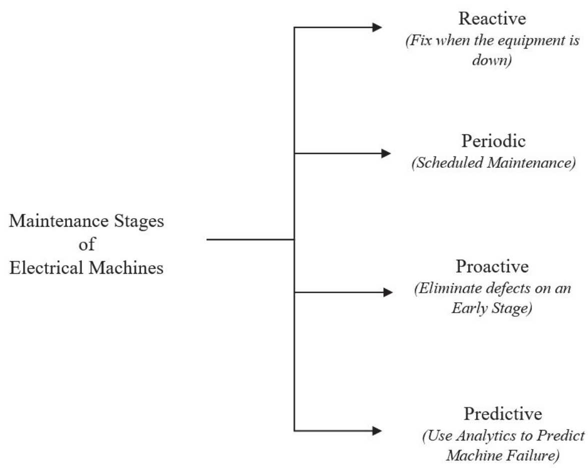


Figure 1. Maintenance phases of an electrical machine.

Condition monitoring systems are implemented concerning electrical machines and making their way into other industries. Currently, most of the work related to classification or detection using such systems is in the health domain [17–19]. Wearable devices are used to monitor patients for any abnormal health conditions [16,17] so that doctors can react quickly if a problem occurs. The development of such systems has also made its way into environmental control, where systems have been developed for monitoring air pollution [20], carbon dioxide [21], and solar power [22]. Systems are also being developed to monitor and control residential buildings, making them into smart homes [23] and monitoring grids, and controlling network congestion. With the development of condition monitoring systems, researchers are now looking at cost-effective systems, especially in the domain of electrical machines, with more work being done on microcontroller boards. This can help gather data, especially from off-shore electrical machines. Hence, condition monitoring systems are being developed, which are particularly targeted toward weather sensors [24] and wind turbines [25–27]. Researchers are also looking into a more portable approach towards condition monitoring systems with the advancement of electrical machines, which can give better mobility and less complexity [20,28–31].

The recent advancement in micro-controller boards has made way for the development of condition monitoring systems [7,32,33] utilizing these boards as it is not only cost-efficient but flexible and scalable too. This work is still in its early stages and is underdeveloped. Most of the condition monitoring systems designed using this approach lack the sampling frequency of data acquired through it, as generally, it is kept low. However, work is being done for higher sampling frequency and more stable data acquisition systems based on micro-controller boards [7,34]. These micro-controller boards are scaling at a rapid pace with their safety, security, and power taken into account. Soon, it will be enough to develop more stable and faster data acquisition systems utilizing these boards, with will be cost-efficient and productive compared to currently available data acquisition systems. Currently, most of the data acquisition systems in place are either developed with SCADA or PLC [35–38], which are complex to use and expensive. Data acquisition cards are available in the market but are expensive compared to micro-controller boards and need a separate system to set up.

One of the most crucial parts of predictive maintenance or condition monitoring [39] is the implementation of machine learning-based trained models. The accuracy of these models depends mainly on the quality of data used for training and their diversity. Hence, it is always instructed to utilize high-quality data samples with all possible scenarios and an optimal number of features to get better results. Artificial Neural Networks (ANNs) are the most commonly used models for fault detection classification. Although other offline models are available for the diagnostics of electrical machines like FEM [40,41], they are primarily for offline implementation and take much more time, which is unsuitable for real-time detection. ANNs models are usually trained with high processing power systems or cloud systems [42,43] to get minimum training time. However, there might be a chance that the model gets over-trained, so it is always good to check the optimal number of samples needed and optimize the machine learning algorithm.

Much work is being done on the condition monitoring of electrical machines [12] and fault detection [44] in electrical machines and bearings [45,46]. However, not much work is being done in the field of fault prediction in electrical machines. Some researchers are working in this domain, but some are utilizing the time domain while other systems are still being developed or are primarily for offline analysis and diagnostics [16,47]. Some of the other systems used in this domain only utilize fault detection while gathering data for training, or there are some commercialized products available. However, they are not only expensive, but their technology is still not disclosed.

This article presents a new signal spectrum-based approach for fault prediction in electrical machines based on signal processing and machine learning. Different machine learning models are trained using the data generated through the proposed method to check their accuracy and performance. A comparison between different neural networks is also given to help choose the best suited for this case.

2. Methodology

This section of the article can be divided into three parts: (i) Signal Processing, (ii) Data Preparation, and (iii) Machine Learning Training and comparison between trained models. The method proposed here is a new approach to fault prediction of electrical machines utilizing the frequency spectrum. This article includes the spectrum of an induction machine with broken rotor bar faults. In this article, one broken rotor bar is used as a reference for a faulty case. The method includes data gathered through Dewetron from multiple induction motors for healthy and faulty cases. The general overview of the data acquisition for the article, including the proposed method, is shown in Figure 2. The acquired data are preprocessed to get distinct features in healthy and faulty cases, which will be explained below. The processed data are then used for data generation to cover expected cases, and a model is trained to test it with real-time data from the induction machine.

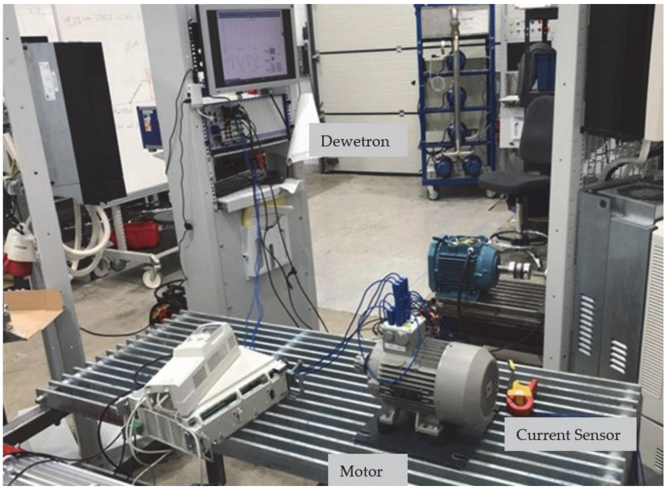


Figure 2. Data acquisition experimental setup.

As it is difficult to gather data for each scenario, primarily when the fault is generated or is about to be generated, it is of utmost importance to consider all the possible cases and the effect of fault frequency components. Researchers work on generating this data using Simulink models or statistical equations to cover more situations and gather more data samples. However, this approach usually does not consider the effect of external parameters on the electrical machine. Here, a more straightforward approach is taken to generate missing data in between or to cover all the expected outcomes. The general overview of the paper is shown in Figure 3.

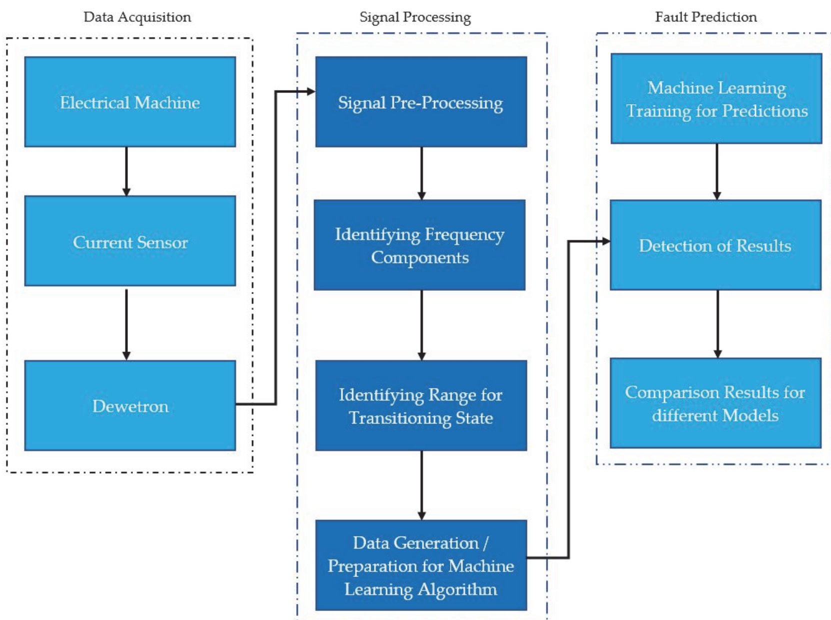
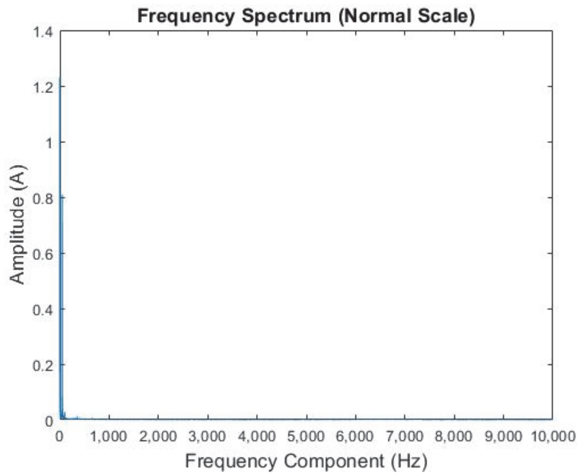


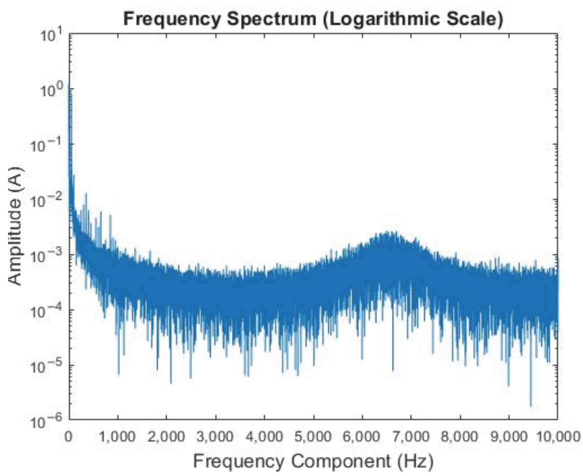
Figure 3. General overview of the proposed method in the article.

2.1. Signal Processing

The first step is to process the gathered data and make it suitable for training. The data gathered through Dewetron are in the time domain, which is converted into the frequency domain using Fast Fourier Transform (FFT). The approach taken here is considering the current signature of the electrical machines and the effect of faults on them. A comparison between the entire frequency spectrum of both healthy and faulty cases is shown in Figure 4a. The spectrum is also shown in the logarithmic scale in Figure 4b for better understanding.



(**a**)



(**b**)

Figure 4. Frequency spectrum of current sample (**a**) general scale (**b**) logarithmic scale.

As a wide range of frequency components are present in the spectrum, the first step is to identify the most prominent frequency components and filter out the negligible ones. This will give fewer frequency components and will make identifying distinct components easier. In this case, the frequency range was decided to be up to 500 Hz as, after this range, the frequency component amplitude is negligible and is not making any significant difference. The frequency spectrum after applying the cut-off is shown in Figure 5.

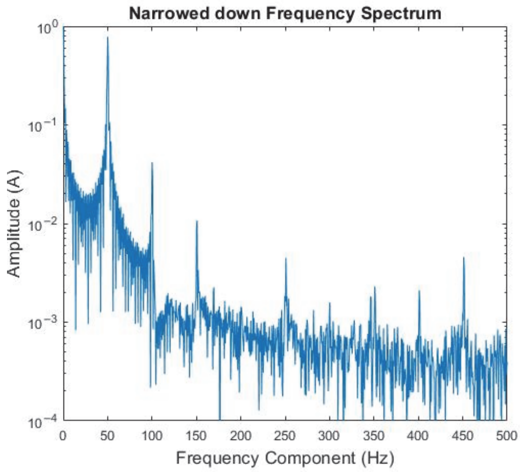
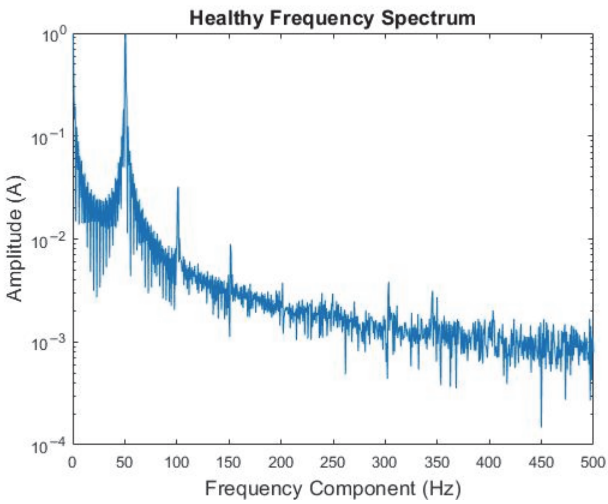


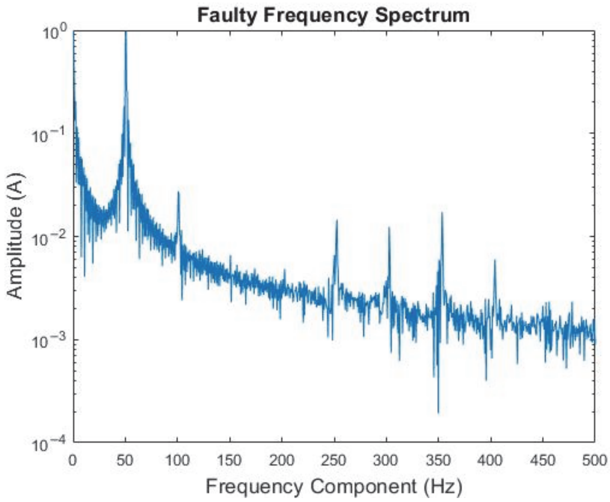
Figure 5. Narrowed down frequency spectrum example.

After narrowing down the frequency components used for training, a comparison is made between the healthy and faulty spectrums to narrow down the frequency components that make a difference. This comparison is carried out for multiple cases, and data are collected from different induction motors to help identify the correct frequency components. General spectrums and their difference for one of the samples are shown in Figure 6. Once it is narrowed down, the most prominent components are selected to help determine the specific fault. This help simplifies the training more, and these components will be used as one of the basics to complete combinations for the training of the predictive algorithm.

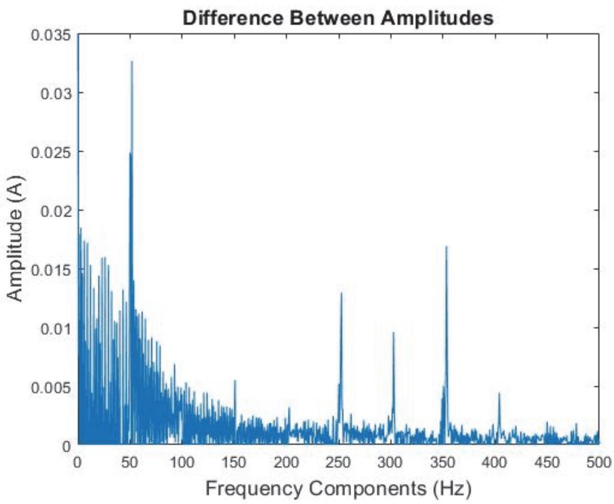


(**a**)

Figure 6. *Cont.*



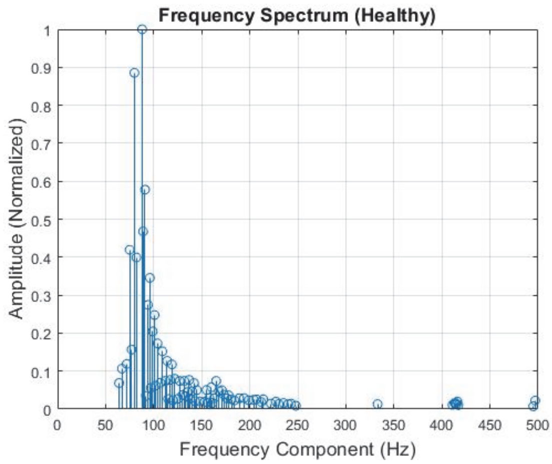
(**b**)



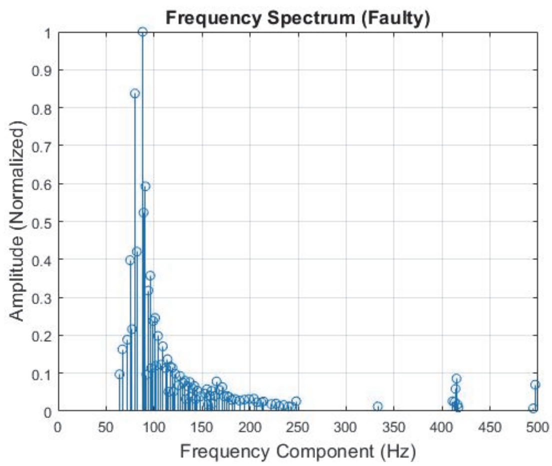
(**c**)

Figure 6. Frequency spectrum (**a**) healthy case, (**b**) faulty case, (**c**) difference to identify frequency components.

The data are then further processed to check the amplitude of these frequency components in healthy and faulty cases. This process is carried out for multiple samples from different induction machines so that the identified components are universal for this specific fault. All amplitudes are normalized to lie between the range of 0–1 to get consistent results. Figure 7 shows an example of a frequency spectrum for both healthy and faulty cases with frequency components and amplitudes.



(**a**)



(**b**)

Figure 7. Frequency spectrum with normalized amplitudes (0–1) (**a**) healthy case, (**b**) faulty case.

After going through multiple samples, the changes in amplitude of the frequency components for both healthy and faulty are singled out. After careful analysis, the amplitude range for the prominent frequency components for fault occurrence is determined. Some of the frequency components with their amplitude range for the healthy case are shown in Table 1, whereas the faulty case is shown in Table 2. This range will help specify the fault occurrence probability for the predictive algorithm.

Table 1. Frequency amplitude range for fault occurrence (healthy signal).

Frequency Component (Hz)	Minimum Amplitude (A)	Maximum Amplitude (A)
38.45 Hz	0.0031	0.612
43.33 Hz	0.0022	0.0797
100 Hz	1.3004×10^{-4}	0.0460
125.73 Hz	7.5295×10^{-5}	0.0139
250.21 Hz	4.92×10^{-5}	0.0092
380.86 Hz	2.96×10^{-6}	8.56×10^{-4}
404.66 Hz	9.1363×10^{-6}	0.0049

Table 2. Frequency amplitude range for fault occurrence (faulty signal).

Frequency Component (Hz)	Minimum Amplitude (A)	Maximum Amplitude (A)
38.45 Hz	4.72×10^{-4}	0.0633
43.33 Hz	6.8×10^{-4}	0.0841
100 Hz	8.6951×10^{-5}	0.0563
125.73 Hz	3.415×10^{-4}	0.0140
250.21 Hz	1.50×10^{-4}	0.0145
380.86 Hz	3.86×10^{-5}	0.0032
404.66 Hz	1.7432×10^{-5}	0.0083

After identifying the frequency components and their amplitude ranges for healthy and faulty cases, the trend of change in amplitude is noted. This will help to generate data and form combinations for the training of machine learning mode for fault prediction.

2.2. Data Preparation

After identifying the ranges and combinations, the next step is to prepare the data for training the machine learning model. In this case, the data gathered from the electrical machine are either for the healthy or faulty case. The combinations present are for either of the cases, and there are no such data samples at this point that can predict the movement before fault occurrence or chances of fault occurrence. To compensate for this lack of data points in between, we will be using average to get the range value of range between the healthy and faulty cases. Equation (1) depicts the calculation of frequency amplitudes for the case between healthy and faulty states.

$$y_t = \frac{\left(y_f + y_h\right)}{2} \tag{1}$$

Here, y_t is the higher average amplitude of the frequency component when it is transitioning from a healthy to a faulty state. Let us say this is the transition state of the motor, whereas y_h and y_f represent the maximum amplitude of the frequency component at healthy and faulty states, respectively. This will give the range of values for the frequency component amplitude between the transitioning state, which can be used further to determine which combinations can identify the faulty frequency components. A general overview of the amplitude range of different frequency components during the transitioning state is shown in Table 3.

Table 3. Frequency amplitude range for fault occurrence (transition state).

Frequency Component (Hz)	Minimum Amplitude (A)	Maximum Amplitude (A)
38.45 Hz	0.612	0.0633
43.33 Hz	0.0797	0.0841
100 Hz	0.0460	0.0563
125.73 Hz	0.0139	0.0140
250.21 Hz	0.0092	0.0145
380.86 Hz	8.56×10^{-4}	0.0032

The difference in ranges for a specific frequency component is graphically shown in Figure 8. This gives an idea about the specific ranges needed to generate combinations for training a machine learning model.

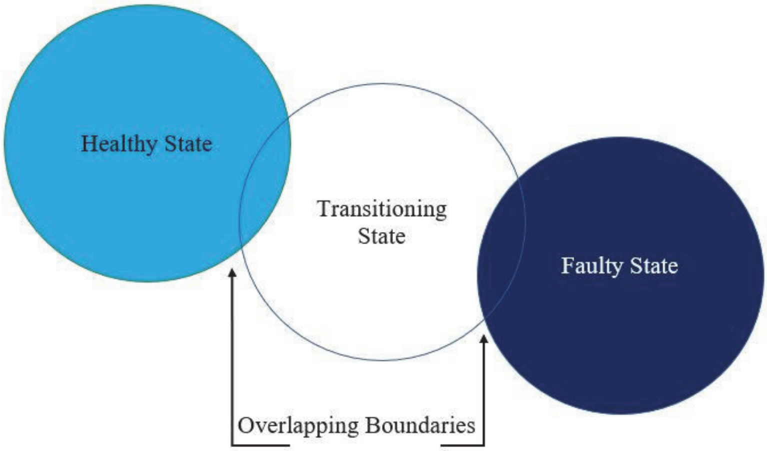


Figure 8. Overlapping frequency amplitude range for three states.

It can be seen that the ranges might overlap a bit, but it is either on the minimum side or maximum size of the transitioning state of the motor, which is to be expected. Once all the ranges of each considered frequency component are determined, the next step is to divide these ranges further into three parts, i.e., each part will be 30% of the range, excluding healthy and faulty areas. This will give us some idea about how much of a chance there is for the occurrence of a fault in the electrical machine. Table 4 shows the division of one of the frequency component amplitudes in the transition state.

Table 4. Division of transition state frequency component range for 250.21 Hz frequency component.

Minimum Amplitude (A)	Maximum Amplitude (A)	Fault Occurrence Probability
0	0.00828	0%
0.00828	0.01052	30%
0.01052	0.01185	60%
0.01186	0.01317	90%
0.01317	-	100%

Table 4 shows the division of the range for one of the frequency component's amplitudes. This will help define the probability of fault occurrence in the incoming signal and will further enable the determination of the fault occurrence level. Once these are established, data points are generated based on these ranges, which will then be used for training the machine learning algorithm. Multiple combinations of these ranges are created to avoid missing out on any possible scenarios. The generated data are then combined with the data for healthy and faulty states for the specific frequency ranges and used for training the machine learning algorithm. The probability of fault occurrence is taken as an average with a weight of the ranges used in the combination. A weight is assigned based on the critical value of the frequency component amplitude and in which range it lies.

For example, for the initial combinations, the range values are the same, so similar weightage is applied to each range value to determine the probability of the fault occurrence, and the average of those probabilities is taken. This will give us the same probability of the urgency of the fault that is occurring. The ranges are divided into five parts for simplicity, as shown in Table 4 above. The weightage assigned to each range is shown in Table 5. The final occurrence probability percentage is decided by taking the average for all frequency components, as shown in Equation (2).

$$f_p = \frac{(a_1 + a_2 + a_3 + \ldots + a_n)}{n} \quad (2)$$

Here, f_p is the probability of fault occurrence, whereas a_1 to a_n are the assigned probabilities to different frequency component amplitudes in the combination, and n is the total number of frequency components present in the combination. This will give an average probability for the fault occurrence, which is then simplified based on the division shown in Table 4. The general overview of the data points combination example used for training the machine learning mode is shown in Figure 9.

Table 5. The weightage assigned to the range of amplitude of frequency component.

Range %Age of Error	Weightage Assigned (0–1)
0%	0
30%	0.3
60%	0.6
90%	0.9
100%	1

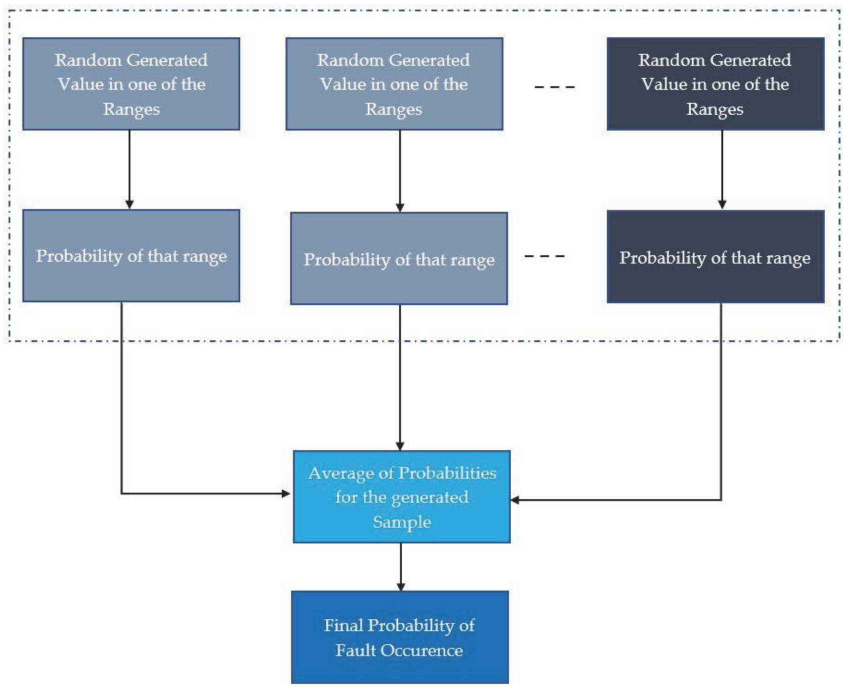


Figure 9. Training data points example.

This will determine the probability of fault occurrence in the electrical machine. However, for this research, the probability is rounded off to 0%, 30%, 60%, 90%, and 100% to classify the data for these five cases. However, they can be further divided into multiple options, and a machine learning model can be trained based on them. Once the data are prepared, different machine learning models are used for training the sample data. The blind validation method was used to determine the accuracy of the trained model.

2.3. Machine Learning Training and Comparison

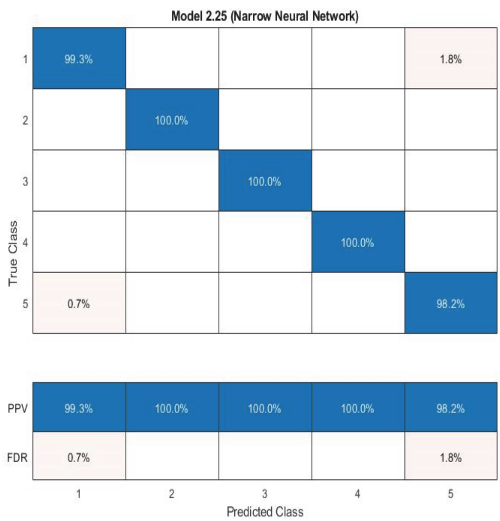
After completing the data points, the next step is to train machine learning models based on those data sets and validate the results to see if they predict the occurrence correctly or not. To ensure we cover all possible cases, healthy and faulty data points for validation were gathered from running electrical machines, whereas the data points for validation for transitioning state were randomly generated from the defined ranges. For comparison purposes, five different kinds of machine learning algorithms were selected, and models were trained using those models.

For machine learning, different neural network algorithms were trained to compare purposes. As this is the initial stage of the proposed algorithm, the data points used for training the machine learning based models were around 68,000 data samples with a validation sample count of 6800 data samples. These initial tests were carried out on smaller data sets and might need to be tested for bigger data sample sets. Each range was assigned a classification label, which is shown in Table 6.

Table 6. Classification assigned per range of error.

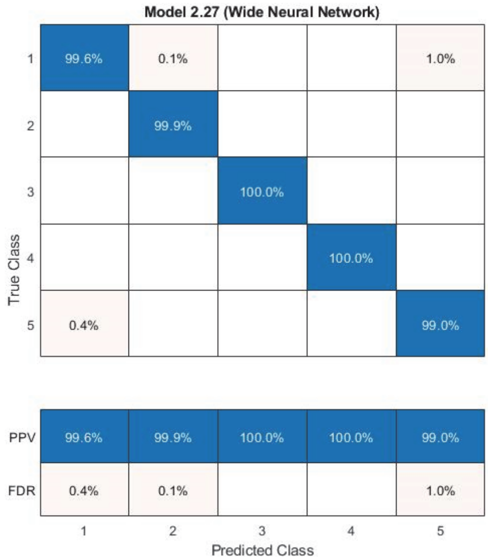
Range %Age of Error	Classification Label Assigned
0%	1
30%	2
60%	3
90%	4
100%	5

Machine learning models were trained based on blind validation, i.e., the samples used for validation were not used for the training of the models. A total of eight models were considered, with the majority from neural networks for classification. A comprehensive comparison between the accuracies of these models is given in Table 7. The confusion matrix for the validation results of the two models is shown in Figure 10.



(a)

Figure 10. *Cont.*



(**b**)

Figure 10. Machine learning results (**a**) narrow neural network, (**b**) wide neural network.

Table 7. Comparison results.

Machine Learning Algorithm	Accuracy (Validation)
Course Tree	93.9%
Gaussian Naïve Bayes	88.6%
Fine KNN	97.1%
Narrow Neural Network	99.3%
Medium Neural Network	99.3%
Wide Neural Network	99.6%
Bilayered Neural Network	99.3%
Trilayered Neural Network	99.1%

As can be seen from Table 7, all of the Neural Network techniques performed well, whereas the others were nearby. This might be because the data set is small and not too big. Further experiments need to be done with bigger data sets to confirm results.

3. Discussion and Conclusions

There has been much research in predictive maintenance, but it is still lacking a good predictive maintenance algorithm. Most of the algorithms being utilized at the moment are related to fault detection, and work is still under development in the area of the prediction of faults. Some commercial products are available in the market, but they are too expensive and company-dependent. Moreover, the technology included for these products is confidential but includes both hardware and software. The algorithm presented in this paper is a step towards a stable and general-purpose approach for fault prediction in electrical machines, which can be implied to different faults.

This paper proposes a novel single-spectrum-based approach for the predictive maintenance of electrical machines. This is a new concept and might need some more refinement and research. The method presented here is based on the current signature fluctuations because of faults in electric machines and utilizing those changes to predict faults. There is only one fault considered for this research, i.e., a broken rotor bar in the electrical machine for reference.

There are two different parts of the presented approach; one is signal processing, whereas the other is the preparation of data samples for training a machine learning model for fault prediction. The proposed method and experiment show a promising result. The data set used for training is small, which might be one reason for the higher accuracy of neural network models. Nevertheless, this shows that predictive maintenance can be achieved using the proposed method. There is still a need to test the approach on a bigger data set with more faulty scenarios and combinations. Although the method needs initial processing to be processed on the incoming signals before giving it to detection, it does not take much time, and the models can be implemented and tested in real-time scenarios.

However, there is still a need to improve the method and include different faults of the electrical machines and also work on multiple combinations between transition, healthy and faulty states. This will make the algorithm broader and will help with the predictive maintenance of machines. Moreover, the algorithm can be improved in determining the urgency of maintenance by adding more layers of transition state and combinations. It might be possible in the future to utilize the presented approach for predicting different faults of electrical machines in real time.

The presented algorithm/approach for fault prediction is still in its early phase. For test purposes, only one type of fault was considered to validate the algorithm. This work can be further extended by considering other faults and testing the algorithm to validate their accuracy. It would also be beneficial to test out its general approach and to evaluate its working with any fault of the electrical machine just by changing the faulty signal. The future work also includes the implementation of this algorithm with bigger data samples and more complex faults.

Author Contributions: Conceptualization, H.A.R. and K.K.; methodology, H.A.R.; software, H.A.R.; validation, H.A.R., K.K. and B.A.; formal analysis, H.V.K.; data curation, B.A.; writing—original draft preparation, H.A.R.; writing—review and editing, K.K.; visualization, A.R.; supervision, A.K.; project administration, T.V. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to these results received funding from the EEA/Norway Grants 2014–2021, Industrial Internet methods for electrical energy conversion systems monitoring and diagnostics. The “Industrial Internet methods for electrical energy conversion systems monitoring and diagnostics” benefits from a 993,000€ grant from Iceland, Liechtenstein and Norway through the EEA Grants. The project aims to provide research in the field of energy conversion systems and to develop artificial intelligence and virtual emulator-based prognostic and diagnostic methodologies for these systems. The project contract with the Research Council of Lithuania (LMTLT) No is S-BMT-21-5 (LT08-2-LMT-K-01-040).

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

References

1. Zhou, K.; Liu, T.; Zhou, L. Industry 4.0: Towards future industrial opportunities and challenges. In Proceedings of the 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, China, 15–17 August 2015; pp. 2147–2152. [CrossRef]
2. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
3. Internet of Things Architecture Project. Available online: http://www.iot-a.eu/public (accessed on 30 November 2022).
4. Pulavskyi, A.; Krivenko, S.S.; Kryvenko, L.S. Functional diagnostic using electrical impedance tomography reconstruction and the Internet of Things. In Proceedings of the 2017 6th Mediterranean Conference on Embedded Computing, MECO 2017—Including ECYPS 2017, Bar, Montenegro, 11–15 June 2017; pp. 1–4. [CrossRef]
5. Raja, H.A.; Raval, H.; Vaimann, T.; Kallaste, A.; Rassolkin, A.; Belahcen, A. Cost-efficient real-time condition monitoring and fault diagnostics system for BLDC motor using IoT and Machine learning. In Proceedings of the Diagnostika 2022–2022 International Conference on Diagnostics in Electrical Engineering, Proceedings, Pilsen, Czech Republic, 6–8 September 2022; pp. 1–4. [CrossRef]
6. Rajan, V.; Sobhana, N.; Jayakrishnan, R. Machine Fault Diagnostics and Condition Monitoring Using Augmented Reality and IoT. In Proceedings of the 2nd International Conference on Intelligent Computing and Control Systems, ICICCS 2018, Madurai, India, 14–15 June 2018; pp. 910–914. [CrossRef]
7. Raja, H.A.; Vaimann, T.; Rassolkin, A.; Kallaste, A.; Belahcen, A. IoT Based Tools for Data Acquisition in Electrical Machines and Robotics. In Proceedings of the 2021 IEEE 19th International Power Electronics and Motion Control Conference, PEMC 2021, Gliwice, Poland, 25–29 April 2021; pp. 737–742. [CrossRef]

8. Wang, J.; Lim, M.K.; Wang, C.; Tseng, M.-L. The evolution of the Internet of Things (IoT) over the past 20 years. *Comput. Ind. Eng.* **2021**, *155*, 107174. [CrossRef]
9. Frosini, L. Monitoring and Diagnostics of Electrical Machines and Drives: A State of the Art. In Proceedings of the 2019 IEEE Workshop on Electrical Machines Design, Control and Diagnosis, WEMDCD 2019, Athens, Greece, 22–23 April 2019; Volume 1, pp. 169–176. [CrossRef]
10. Parpala, R.C.; Iacob, R. Application of IoT concept on predictive maintenance of industrial equipment. *MATEC Web Conf.* **2017**, *121*, 02008. [CrossRef]
11. Zhang, W.; Yang, D.; Wang, H. Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey. *IEEE Syst. J.* **2019**, *13*, 2213–2227. [CrossRef]
12. Raja, H.A.; Kudelina, K.; Asad, B.; Vaimann, T. Fault Detection and Predictive Maintenance for Electrical Machines. In *New Trends in Electric Machines—Technology and Applications*; IntechOpen: London, UK, 2022.
13. Compare, M.; Baraldi, P.; Zio, E. Challenges to IoT-Enabled Predictive Maintenance for Industry 4.0. *IEEE Internet Things J.* **2019**, *7*, 4585–4597. [CrossRef]
14. Yoo, Y.; Baek, J.-G. A Novel Image Feature for the Remaining Useful Lifetime Prediction of Bearings Based on Continuous Wavelet Transform and Convolutional Neural Network. *Appl. Sci.* **2018**, *8*, 1102. [CrossRef]
15. Li, S.; Li, J. Condition monitoring and diagnosis of power equipment: Review and prospective. *High Volt.* **2017**, *2*, 82–91. [CrossRef]
16. Kudelina, K.; Asad, B.; Vaimann, T.; Rassõlkin, A.; Kallaste, A.; Van Khang, H. Methods of Condition Monitoring and Fault Detection for Electrical Machines. *Energies* **2021**, *14*, 7459. [CrossRef]
17. Sofi, A.; Regita, J.J.; Rane, B.; Lau, H.H. Structural health monitoring using wireless smart sensor network—An overview. *Mech. Syst. Signal Process* **2022**, *163*, 108113. [CrossRef]
18. Sheela, K.G.; Varghese, A.R. Machine Learning based Health Monitoring System. *Mater. Today Proc.* **2019**, *24*, 1788–1794. [CrossRef]
19. Rahman, N.A.A.; Jambek, A.B. Biomedical health monitoring system design and analysis. *Indones. J. Electr. Eng. Comput. Sci.* **2019**, *13*, 1056–1064. [CrossRef]
20. Potbhare, P.D.; Bhange, K.; Tembhare, G.; Agrawal, R.; Sorte, S.; Lokulwar, P. IoT based Smart Air Pollution Monitoring System. In Proceedings of the International Conference on Applied Artificial Intelligence and Computing, ICAAIC, Salem, India, 9–11 May 2022; pp. 1829–1834. [CrossRef]
21. Rodriguez-Huerta, R.; Martinez-Castillo, J.; Morales-Gonzalez, E.; Herrera-May, A.L. Development of a Monitoring System for CO/CO_2 with Android. In Proceedings of the IEEE International Conference on Engineering Veracruz, ICEV, Boca del Rio, Mexico, 14–17 October 2019. [CrossRef]
22. Prasanna, J.L.; Lavanya, D.; Kumar, T.A. Condition monitoring of a virtual solar system using IoT. In Proceedings of the 2nd International Conference on Communication and Electronics Systems, ICCES, Coimbatore, India, 19–20 October 2017; pp. 286–290. [CrossRef]
23. Yu, Y.; Hao, Z.; Li, G.; Liu, Y.; Yang, R.; Liu, H. Optimal search mapping among sensors in heterogeneous smart homes. *Math. Biosci. Eng.* **2022**, *20*, 1960–1980. [CrossRef]
24. Xie, P.; Kennedy, N.; French, G.; Popovici, E.; Edwards-Murphy, F.; O'Ríordáin, S.; Leahy, P. Sensing wind for environmental and energy applications. In Proceedings of the 25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies, Limerick, Ireland, 26–27 June 2014. [CrossRef]
25. Hsu, J.-Y.; Wang, Y.-F.; Lin, K.-C.; Chen, M.-Y.; Hsu, J.H.-Y. Wind Turbine Fault Diagnosis and Predictive Maintenance Through Statistical Process Control and Machine Learning. *IEEE Access* **2020**, *8*, 23427–23439. [CrossRef]
26. Zhao, L.; Zhou, Y.; Matsuo, I.B.M.; Korkua, S.K.; Lee, W.-J. The Design of a Remote Online Holistic Monitoring System for a Wind Turbine. *IEEE Trans. Ind. Appl.* **2020**, *56*, 14–21. [CrossRef]
27. Wymore, M.L.; Van Dam, J.E.; Ceylan, H.; Qiao, D. A survey of health monitoring systems for wind turbines. *Renew. Sustain. Energy Rev.* **2015**, *52*, 976–990. [CrossRef]
28. Srivastava, P.; Bajaj, M.; Rana, A.S. IOT based controlling of hybrid energy system using ESP8266. In Proceedings of the IEEMA Engineer Infinite Conference, eTechNxT 2018, New Delhi, India, 13–14 March 2018; pp. 1–5. [CrossRef]
29. Kalyanraj, D.; Prakash, S.L.; Sabareswar, S. Wind turbine monitoring and control systems using Internet of Things. In Proceedings of the International Conference on 21st Century Energy Needs—Materials, Systems and Applications, ICTFCEN, Kharagpur, India, 17–19 November 2016; pp. 1–4. [CrossRef]
30. Raju, L.; Gokulakrishnan, S.; Muthukumar, P.R.; Jagannathan, S.; Morais, A.A. Iot based autonomous demand side management of a micro-grid using arduino and multi agent system. In Proceedings of the International Conference on Power and Embedded Drive Control, ICPEDC 2017, Chennai, India, 16–18 March 2017; pp. 44–49. [CrossRef]
31. Mhatre, V.; Chavan, S.; Samuel, A.; Patil, A.; Chittimilla, A.; Kumar, N. Embedded Video Processing and Data Acquisition for Unmanned Aerial Vehicle. In Proceedings of the 2015 International Conference on Computers, Communications and Systems, ICCCS 2015, Kanyakumari, India, 2–3 November 2015.
32. Shyamala, D.; Swathi, D.; Prasanna, J.L.; Ajitha, A. IoT platform for condition monitoring of industrial motors. In Proceedings of the 2017 2nd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 19–20 October 2017; pp. 260–265.

33. Milenković, A.; Otto, C.; Jovanov, E. Wireless sensor networks for personal health monitoring: Issues and an implementation. *Comput. Commun.* **2006**, *29*, 2521–2533. [CrossRef]
34. Pjevalica, V.; Pjevalica, N.; Petrovic, N. Cloud Based Data Acquisition via IoT for Electric Power Quality Monitoring. In Proceedings of the Zooming Innovation in Consumer Technologies Conference (ZINC 2019), Novi Sad, Serbia, 29–30 May 2019; pp. 82–85. [CrossRef]
35. Gopalamma, A.S.L.K.; Naik, R.S. Smart Drive Control using IoT. In Proceedings of the TENCON 2019—2019 IEEE Region 10 Conference (TENCON), Kochi, India, 17–20 October 2019; pp. 1853–1856. [CrossRef]
36. Kuzlu, M.; Pipattanasomporn, M.; Rahman, S. Review of Communication Technologies for Smart Homes/Building Applica-tions. In Proceedings of the 2015 IEEE Innovative Smart Grid Technologies—Asia, ISGT ASIA 2015, Bangkok, Thailand, 3–6 November 2015.
37. Yakimov, P.I.; Lovev, A.N.; Tuliev, N.T. Implementation of Internet of Things based solution of universal power transducer. In Proceedings of the 26th International Scientific Conference Electronics, ET 2017—Proceedings 2017, Sozopol, Bulgaria, 13–15 September 2017; pp. 1–4. [CrossRef]
38. Debbag, Y.; Yilmaz, E.N. Internet based monitoring and control of a wind turbine via PLC. In Proceedings of the 2015 3rd International Istanbul Smart Grid Congress and Fair (ICSG 2015), Istanbul, Turkey, 29–30 April 2015; pp. 1–5. [CrossRef]
39. Wang, Q.; Wang, H.; Hou, L.; Yi, S. Overview of Tool Wear Monitoring Methods Based on Convolutional Neural Network. *Appl. Sci.* **2021**, *11*, 12041. [CrossRef]
40. Asad, B.; Vaimann, T.; Belahcen, A.; Kallaste, A.; Rassolkin, A. Rotor Fault Diagnostic of Inverter Fed Induction Motor Using Frequency Analysis. In Proceedings of the 2019 IEEE 12th International Symposium on Diagnostics for Electrical Machines, power Electronics and Drives (SDEMPED), Toulouse, France, 27–30 August 2019. [CrossRef]
41. Asad, B.; Vaimann, T.; Belahcen, A.; Kallaste, A.; Rassõlkin, A.; Iqbal, M. The Cluster Computation-Based Hybrid FEM–Analytical Model of Induction Motor for Fault Diagnostics. *Appl. Sci.* **2020**, *10*, 7572. [CrossRef]
42. Li, J.H.; Zhao, S.; Wang, Y.R. Using Cloud Computing and the Internet of Things to Construct Vehicle Condition Fault Di-agnosis Platform. *Proc. Appl. Mech. Mater.* **2014**, *574*, 752–755. [CrossRef]
43. Aazam, M.; Khan, I.; Alsaffar, A.A.; Huh, E.-N. Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved. In Proceedings of the 11th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 14–18 January 2014; pp. 414–419. [CrossRef]
44. Gong, W.; Chen, H.; Zhang, Z.; Zhang, M.; Wang, R.; Guan, C.; Wang, Q. A Novel Deep Learning Method for Intelligent Fault Diagnosis of Rotating Machinery Based on Improved CNN-SVM and Multichannel Data Fusion. *Sensors* **2019**, *19*, 1693. [CrossRef]
45. Kudelina, K.; Asad, B.; Vaimann, T.; Belahcen, A.; Rassõlkin, A.; Kallaste, A.; Lukichev, D.V. Bearing Fault Analysis of BLDC Motor for Electric Scooter Application. *Designs* **2020**, *4*, 42. [CrossRef]
46. Xu, G.; Liu, M.; Jiang, Z.; Söffker, D.; Shen, W. Bearing Fault Diagnosis Method Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning. *Sensors* **2019**, *19*, 1088. [CrossRef] [PubMed]
47. Kudelina, K.; Vaimann, T.; Rassolkin, A.; Kallaste, A.; Demidova, G.; Karpovich, D. Diagnostic Possibilities of Induction Motor Bearing Currents. In Proceedings of the International Scientific Technical Conference Alternating Current Electric Drives (ACED), Los Angeles, CA, USA, 23–25 January 2021; pp. 1–5. [CrossRef]

Publication II

Asad, B.; Raja, H.A.; Vaimann, T.; Kallaste, A.; Pomarnacki, R.; Hyunh, V.K. A Current Spectrum-Based Algorithm for Fault Detection of Electrical Machines Using Low-Power Data Acquisition Devices. Electronics 2023, 12, 1746. DOI: 10.3390/electronics12071746.

electronics

Article

A Current Spectrum-Based Algorithm for Fault Detection of Electrical Machines Using Low-Power Data Acquisition Devices

Bilal Asad [1,2,*], Hadi Ashraf Raja [2], Toomas Vaimann [2], Ants Kallaste [2], Raimondas Pomarnacki [3] and Van Khang Hyunh [4]

[1] Department of Electrical Power Engineering, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
[2] Department of Electrical Power Engineering and Mechatronics, Tallinn University of Technology, 12616 Tallinn, Estonia; hadi.raja@taltech.ee (H.A.R.); toomas.vaimann@taltech.ee (T.V.); ants.kallaste@taltech.ee (A.K.)
[3] Department of Electronic Systems, Vilnius Gediminas Technical University, 10223 Vilnius, Lithuania; raimondas.pomarnacki@vilniustech.lt
[4] Department of Engineering Sciences, University of Agder, 4879 Grimstad, Norway; huynh.khang@uia.no
* Correspondence: bilal.asad@taltech.ee

Abstract: An algorithm to improve the resolution of the frequency spectrum by detecting the number of complete cycles, removing any fractional components of the signal, signal discontinuities, and interpolating the signal for fault diagnostics of electrical machines using low-power data acquisition cards is proposed in this paper. Smart sensor-based low-power data acquisition and processing devices such as Arduino cards are becoming common due to the growing trend of the Internet of Things (IoT), cloud computation, and other Industry 4.0 standards. For predictive maintenance, the fault representing frequencies at the incipient stage are very difficult to detect due to their small amplitude and the leakage of powerful frequency components into other parts of the spectrum. For this purpose, offline advanced signal processing techniques are used that cannot be performed in small signal processing devices due to the required computational time, complexity, and memory. Hence, in this paper, an algorithm is proposed that can improve the spectrum resolution without complex advanced signal processing techniques and is suitable for low-power signal processing devices. The results both from the simulation and practical environment are presented.

Keywords: electrical machine; machine learning; data acquisition; FEM; signal processing; Arduino; artificial intelligence

check for updates

Citation: Asad, B.; Raja, H.A.; Vaimann, T.; Kallaste, A.; Pomarnacki, R.; Hyunh, V.K. A Current Spectrum-Based Algorithm for Fault Detection of Electrical Machines Using Low-Power Data Acquisition Devices. *Electronics* **2023**, *12*, 1746. https://doi.org/10.3390/electronics12071746

Academic Editors: Ryad Zemouri, Mélanie Lévesque and Arezki Merkhouf

Received: 10 March 2023
Revised: 4 April 2023
Accepted: 4 April 2023
Published: 6 April 2023

1. Introduction

The research in the predictive maintenance of electrical machines is touching new horizons. Cloud computation and distributed low-cost sensors are integral for Industry 4.0 standards. They can also be considered a paradigm shift in the predictive maintenance of electrical machines. Low-cost data acquisition sensors are becoming essential as electrical machines are becoming increasingly popular in small and medium-range electric vehicles. The research in the field of condition monitoring of electrical machines using stator currents [1–3], stator voltages [4–6], speed and torque ripples [7,8], stray flux [9–14], vibration analysis [15–19], thermal analysis [20–23], acoustic analysis [24–27], work in the steady-state interval [28], or transient regime [9,29–32] can be considered as mature enough after over a century of research. The research path started with conventional signal processing and harmonic estimation-based techniques. Here, the fundamental rule was to discover the fault-based new frequency components in the machine's global signal. The signal processing techniques were explored by researchers extensively to secure or protect the tiny, sensitive, fragile, and load-dependent fault-based information. For this purpose, the improvement in the spectrum resolution both in stationary and transient regimes was the common point of interest. To remove the spectral leakage, the best practice both in IEEE and industry standards is to obtain the coherent sampling to the maximum extent [33,34]. A variety of other methods have also been explored in the literature, such as filter banks [35], adaptive filters [36,37], 2D feature [38], optimization of truncating windows [39,40], singular value decomposition [41–43], orthogonal matching pursuit [44–46], interpolated DFT techniques [47], Taylor Fourier transforms [48], multiple signal classification (MUSIC) [49,50], fault estimation using weighted iterative learning [51], auxiliary classifier generative adversarial network [52], and estimation of signal parameters via rotational invariance technique (ESPIRIT) [53]. The complexity of the required memory and calculation time are, however, problems that can limit their application in low-power data processing devices. The next major research domain is the mathematical modelling of electrical machines, as those are essential for the design, control, analysis, and fault-based simulations of electrical machines. The main task on which researchers put a lot of focus is to reduce the approximations and the simulation time of the fault simulation-compatible mathematical models. A large amount of research can be found in literature, ranging from finite element method (FEM) [54] to analytical models such as modified winding function analysis (MWFA) [55–57], reluctance network-based [58], and hybrid models [59,60]. As these models should be detailed and able to simulate every kind of fault, the simulation time and complexity are a big issue. The extended simulation time for fault diagnostics is not acceptable, as in the most advanced diagnostic techniques the simulation should run in parallel with the actual hardware, such as digital twin and hardware in the loop. A considerable research effort regarding the minimization of the simulation time both in FEM and analytical techniques can be found in literature, where [61] used piece-wise polynomial function for model order reduction, [62] used Loewner matrix interpolation, [63,64] used proper orthogonal decomposition, [65] used Krylov subspace techniques, etc. The development of these models opened new research directions where they can be used in the hardware in the loop environment [66], parameters estimation [67,68], digital twin [69], and inverse problem theory [70]. The research in these domains is complicated though due to the complex mathematical models, coupling effects in the motor variables, multiple solution points of the same problem, etc. These problems then opened the field, such as optimization theory [71], probability and stochastic analysis [72], non-linear control theory [73], and statistical analysis [74] of the global signals for the predictive maintenance of electrical machines. The development of these models paved the way towards another more advanced field, artificial intelligence [75]. A significant number of AI-based research articles can be seen in the literature and the number is increasing by leaps and bounds. The accuracy and maturity of AI algorithms depends on the data size and its variety under different loading and faulty conditions. Thanks to the research in the field of mathematical modelling, data collection under different faulty and loading conditions for a variety of different machines is possible using simulations. Moreover, data storage on the cloud can increase the training data set every day. The common point in all conventional and advanced techniques is the input signal. Mostly, the global signals remain the same for all types of machines as the state variables of all machines are almost the same. Now there is a paradigm shift in the measurement of all those signals using low-cost data acquisition devices such as Arduino cards and sending the data in the database without loss or any additional infiltrations such as noise.

In this paper, an algorithm is proposed that can improve the spectral resolution with the help of the following contributions.

1. The integral number of cycles and the signal's length whose prime factors are appropriate are calculated first. The fractional parts of the signal in the start and end reduce the spectrum's resolution, and an inappropriate length of the signal with a large number or size of prime factors decreases FFT's efficiency by increasing the complexity, required memory, and calculation time.
2. The low sampling frequency is the main problem when the data acquisition devices are not very powerful and are intended to work online with systems such as Arduino.

In Industry 4.0, those low-cost devices can have significant importance because of Internet of Things (IoT), distributed smart sensors, and cloud computation. The low sampling frequency leads to poor frequency resolution and increased spectral leakage. The main reason for this is sharp changes in the acquired signal. Hence, those sharp changes are proposed to be removed using data interpolation. This step is also important when the diagnostic algorithms depend on the mathematical model of the system. The most accurate models are the finite element method (FEM), based which the computational complexity is always a challenge. By using data interpolation, only the minimum number of steps can be simulated, and the rest of the values can be approximated.

3. Detecting any data discontinuity and removing it. In low power smart sensors, the chances of data loss cannot be neglected. This data loss can happen during its transmission from card to cloud due to network issues, due to some clock issues in the data acquisition card itself, or due to limited memory to save the signal before its transmission. This data loss is fatal for FFT-based spectrum analysis. This is due to the resultant data discontinuities in the acquired signals. So, a method is devised to remove data discontinuity, if any.
4. Repeating the cycles for the improvement in the resolution with minimal discontinuity. The increased number of signal cycles lead to a better frequency resolution. As the current and voltage cycles of the electrical machines working under steady state regime are periodic, they can be repeated to increase the signal's length. This repetition of the signal should not be random, which can make the resolution worse. Hence, a technique is proposed to repeat the cycles before frequency analysis if necessary.

2. The Theoretical Background

Almost all kinds of faults modulate the machine's global variables with a particular set of frequencies. The number and the amplitude of those frequency components are a function of the fault type and severity. During the early stages of fault, these harmonics are tiny in amplitude and difficult to detect. They tend to hide themselves under the frequency lobe of the powerful neighboring frequency component. The strength of any diagnostic algorithm is determined from its ability to detect those harmonics at the early stage of the fault. For this purpose, the resolution of the frequency spectrum is of significant importance, which increases with the decrease in the spectral leakage of the powerful frequency components. To reduce the spectral leakage, a variety of advanced signal processing techniques are available in the literature, but at the cost of increased computational time and complexity. It makes those algorithms less suitable for low power signal processing and controller boards. For low power smart sensor-based data acquisition and processing devices, the following fundamental precautionary measures should be accounted for.

5. The signal frequency and sampling frequency must follow conditions of coherency. The perfect coherent data is very difficult to obtain because of measurement equipment limitations and noise. This non-coherency can be avoided by windowing techniques [76]. However, the clever selection of the window is very important to obtain a narrower main lobe with less leakage energy inside the lobes. So, specialized knowledge about the windowing function and its impact on the spectrum is needed to deal with the problems, which cannot be a very easy solution. The drawback of FFT is that any mismatch between the sampling frequency and signal frequency can cause spectral leakage.
6. The signal should have an integer number of cycles. The fractional parts of the signal in the start or end increase the spectral leakage and increase the requirement of windowing function. This approach will increase the efficiency of FFT, will reduce the dependency on windowing function, and will reduce spectral leakage, even if the signal is noisy or its frequency is near the Nyquist rate. The quality of the frequency spectrum can be checked by measuring the signal to noise ratio (SNR), total harmonic distortion (THD), spurious free dynamic range (SFDR), signal to noise and distortion

ratio (SNDR), effective number of bits (ENOF), etc. The number of cycles in a signal can be calculated as

$$J = M\frac{f_{in}}{f_s} = J_{int} + \Delta \tag{1}$$

In this equation, J is the total number of cycles, fin is the frequency of the fundamental component of the near sinusoid signal, f_s is the sampling frequency, M is the recorded signal's length, J_{int} are the integral number of signal cycles, and Δ is the fractional part. The non-zero Δ leads to the spectral leakage.

A signal from time domain to discrete domain can be represented as

$$x(t) = Asin(2\pi f_{in}t + \theta) + hh \tag{2}$$

$$x[k] = Asin\left(2\pi f_{in}\frac{k}{f_s} + \theta\right) + hh \tag{3}$$

$$x[k] = Asin\left(2\pi \frac{J}{M}k + \theta\right) + hh \tag{4}$$

$$x[k] = Asin\left(2\pi \frac{J_{int} + \Delta}{M}k + \theta\right) + hh \tag{5}$$

$$x[k] = x_1[k] + x_h[k] \tag{6}$$

where *hh* represents the higher order harmonics and can be defined as follows: a_n and b_n are the Fourier coefficients.

$$hh(t) = \sum\nolimits_{n\geq 2}(a_n cos2\pi n f_{in}t + b_n sin2\pi n f_{in}t) \tag{7}$$

In squirrel cage induction machines, the main causes of these higher order harmonics are the non-sinusoidal winding distributions, changing airgap reluctance due to rotor and stator slot openings, inherent eccentricity, material saturation, harmonics coming from the supply, and any fault if present in the machine. However, all these harmonics are tiny in comparison with the fundamental component and the overall current signal remains near sinusoidal. The initial purpose is to calculate J_{int} in the acquired signal and discard the fractional part Δ.

The integer number of cycles are calculated in the way that all values greater than the RMS value both on positive and negative half cycle are marked as +1 and -1. All elements are merged into one if the adjacent sign is the same to make a new signal say w[m]. We merge adjacent same values into one element and take the absolute value.

3. The Effect of Discontinuities in the Signal

Although FFT is a very powerful tool that is extensively used in the field of signal processing, for smooth, periodic, uniformly sampled points and long signals, FFT no doubt gives accurate results. However, the results become significantly erroneous if there are singularities or discontinuities in the signals. Thanks to the symmetrical and sinusoidal distributed design and performance parameters of electrical machines, almost all global signals such as current, voltage, and flux are periodic. The data discontinuities are however possible due to the limitations of the data acquisition devices, particularly if those are low power cards. This can be because of network limitations such as delay or loss of data transfer from the device to cloud. Because of the high sample rate, there is a high chance of data loss while data is being transferred from sensors to the low power cards. This is mostly because of the delay in the clearance of the buffers when data are being transmitted for a long time, i.e., a couple of days to weeks. An example of such a data acquisition system is shown in Figure 1.

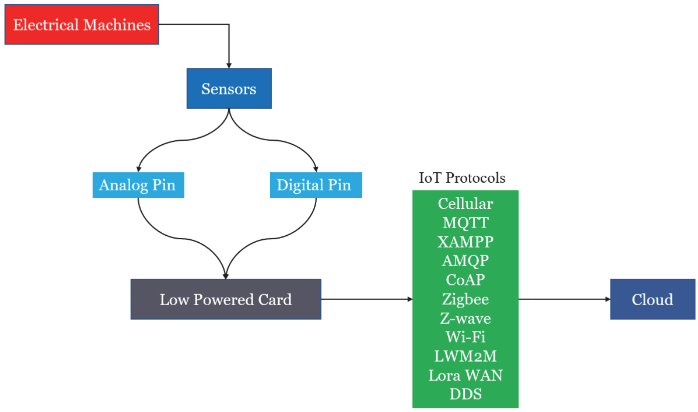


Figure 1. The schematic diagram of data acquisition and transmission to the cloud using IoT.

Data loss can occur in two scenarios for the above data acquisition setup, while the data are being transferred from sensors to the low powered cards and the other while the data are being transferred from the cards to cloud. The protocols used for data transmission have their own limitations too. The loss of data during transmission can be due to the limitation of network or delay/loss of network while transferring. Another reason might be due to the buffers being overloaded and not being properly cleared up before the next data come in, which can result in a loss of data while in transmission. These sharp changes in the signal are the potential cause of hiding the low power fault-based frequencies due to the increased spectral leakage of significant harmonics. It also decreases the computational time of FFT, decreases its efficiency, and increases the need for increased data length. The experimental setup used to recreate such scenario is shown in Figure 2.

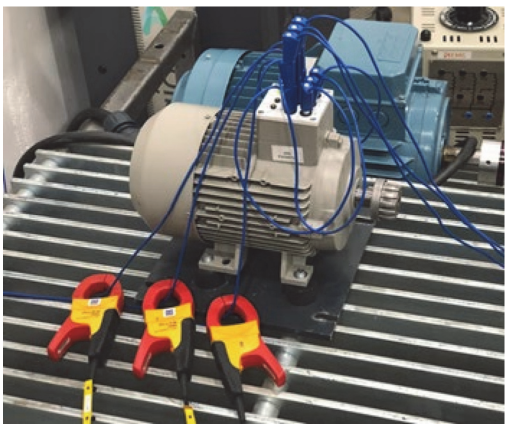


Figure 2. Experimental setup for data collection.

The induction motor is used to collect current signals for all three phases, and it is then transmitted to the cloud using Arduino (low powered card). This is the most common approach used for the data acquisition system when using a low powered card. There are alternate systems that have been proposed that further consider data losses with a local backup of collected data at a node [ref], but the following approach is still widely used. The flow chart of the setup used for data collection for this experiment is shown in Figure 3.

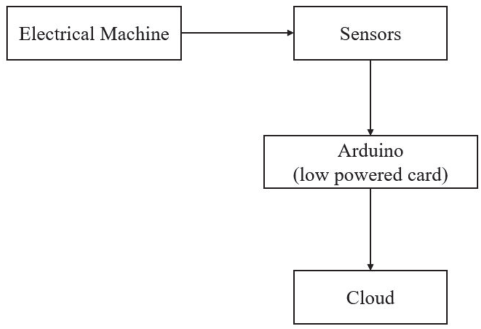


Figure 3. Flow chart for the data acquisition setup.

The setup was run continuously for multiple days with different sampling rates to generate data losses. At higher sampling rates, the data losses occurred more often as the buffer became overloaded. Because of the limitation of the processing power of Arduino (low powered cards), data loss became inevitable in these cases. This is why the sampling rate tended to be on the lower side in most cases, but this also resulted in the samples being too low and similar data loss issues could occur if it kept running for a more extended period. The other scenario was also created by interrupting the network connection. In this case, wi-fi was used to transmit data from Arduino to the cloud database. Upon interruption of the network, as no data were transmitted, this resulted in data being lost. For some protocols, it could result in a delay at the receiving end, but this will still have components lost for the received signal. The setup was used to obtain signals with data discontinuity to check the result of the proposed algorithm.

The data discontinuities were detected by making a moving subtraction filter. The amplitude difference of every two consecutive samples defined the magnitude of discontinuity in them. For example, in Figure 4, nine discontinuities along with their amplitude are discovered that need correction.

$$diff = |x[n]| - |x[n-1]| \tag{8}$$

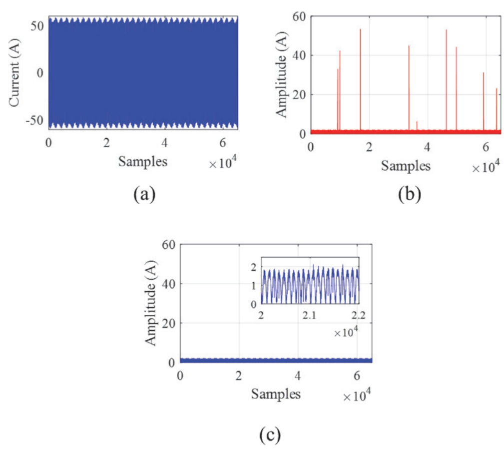


Figure 4. (**a**) The acquired stator current, (**b**) the result of moving subtraction filter for the detection of discontinuities, and (**c**) after the correction of discontinuous samples.

For correction, the discontinuous sample is replaced with the average value of the samples $x\,[n-1]$ and $x\,[n+1]$:

$$\hat{x}[n] = \frac{|x[n+1]| + |x[n-1]|}{2} \tag{9}$$

The integer number of cycles can be calculated using zero cross detection, but, in that case, wrong computation can occur if there is any data discontinuity in the signal. If there are more than one consecutive missing data samples then there are some possible methods of correction. Replace the missing samples with the samples from the same location of the subsequent cycle. The other way is that the samples will be replaced by random values, depending on the amplitude of the available samples at the start and end of the missing segment and the amplitude will be iteratively corrected. The third way is that if the cycles are affected in a worse manner, then it can be totally replaced with the healthy one from the signal. This paper at the moment deals with only one discontinuity between two healthy samples.

4. Counting the Integral Number of Cycles and Removing the Fractional Parts

The integral number of cycles are calculated in the following steps.

A. The samples of the acquired stator current are compared with the RMS value. The samples with a magnitude greater than the RMS value for both the positive and negative side are replaced with one, while all of the other samples are replaced with zero as shown in the equation below and Figure 5b.

$$y[k] = \begin{cases} 1, \frac{i_m}{\sqrt{2}} \leq i[k] \\ 0, \frac{i_m}{\sqrt{2}} \geq \left| i[k] \right| \\ -1, -\frac{i_m}{\sqrt{2}} \geq i[k] \end{cases} \tag{10}$$

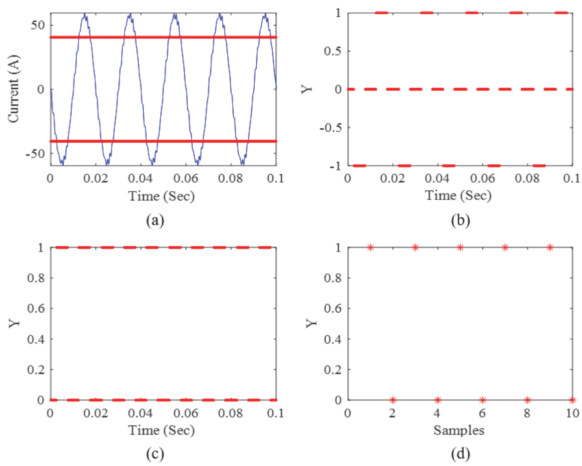


Figure 5. (**a**) The stator current with red line representing the RMS value, (**b**) the samples validating the conditions given in b, (**c**) the shifting of negative samples towards positive side by taking modulus, and (**d**) merging the consecutive samples of same value in one.

B. The modulus of the resultant vector is taken to shift the negative-sided samples to the positive side, as shown in Figure 5c.

C. The consecutive samples with same magnitude are merged into one and represented in Figure 5d. The final number of samples on the zero or unity axis are equal to the number of signal cycles.

After counting the number of cycles, the data are saved until the index of steric completing the integral number of cycles in Figure 5d. Now, two types of discontinuities may still persist in the signal: the minor discontinuity due to low sampling frequency, as shown in Figure 6, and the possible discontinuity at te starting and ending time.

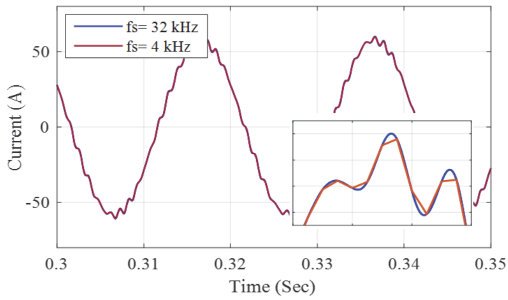


Figure 6. The estimation of intermediate solutions using data interpolation.

Both problems can be solved by signal interpolation. It will not only improve the smoothness of the signal, but also refine the zero crossing points, as shown in Figure 7.

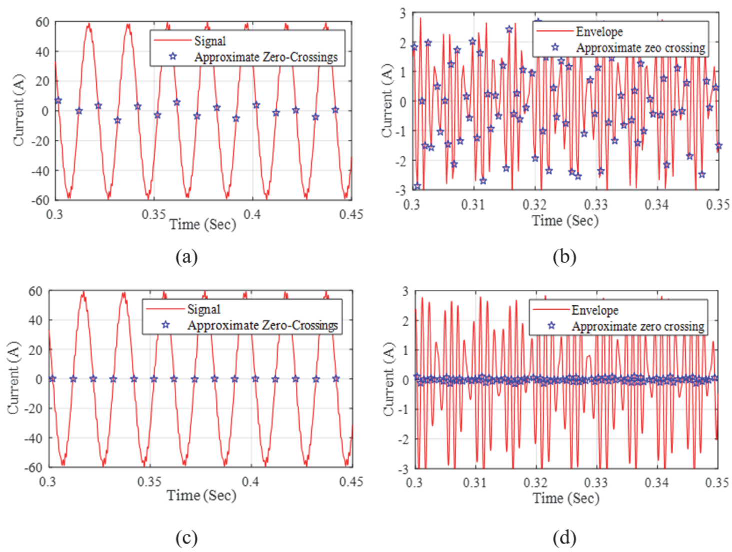


Figure 7. (**a**) The stator current and approximate zero crossings at a sampling frequency of 4 kHz, (**b**) the corresponding envelope shifted across zero line with approximate zero crossings, (**c**) the signal with improved sampling frequency and approximate zero crossings, and (**d**) the corresponding envelope shifted across zero line with approximate zero crossings.

5. Algorithm

The proposed algorithm is shown in Figure 8. Its main parts include the removal of DC offset which decreases the possibility of a frequency bin at 0Hz in the spectrum, detection and correction of data discontinuities which increase the spectral leakage, removal of starting and ending fractional parts and the repetition of the signal if necessary.

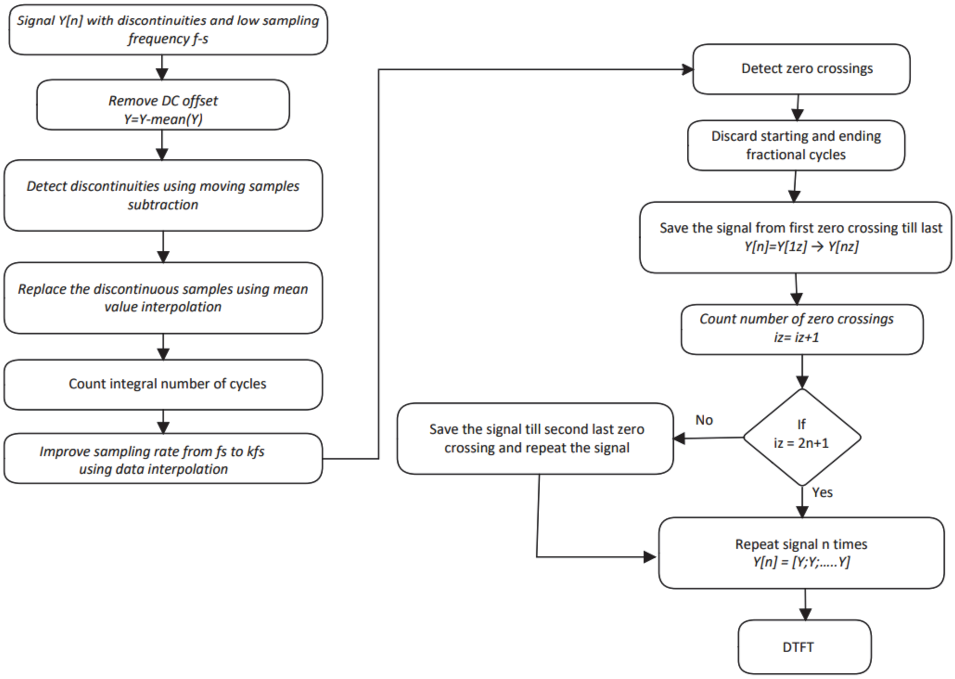


Figure 8. The algorithm for counting the integral number of cycles, removal of signal discontinuities and fractional parts of the signal, data interpolation, and repetition, if necessary.

6. Results

6.1. Simulation Results

The motor's stator current harmonics can be broadly classified into three major categories: the winding and supply-based odd multiples of the fundamental component, the slotting harmonics, and the fault generated harmonics. The mathematical description of these harmonics is given in Table 1. The fault and slotting harmonics are the function of slip and tend to move in the spectrum as the load varies, while the winding MMF and the supply harmonics retain their position in the spectrum. Electrical machine simulations are necessary for several reasons, such as design, control, analysis, and training of the fault diagnostic algorithms, creation of digital twin, inverse problem theory, hardware in the loop environment, and parameters estimation. However, the biggest drawback of finite element method (FEM) models of electrical machines is the computational complexity and the required simulation time. Moreover, the small step size and the simulation of complete geometry is required for better resolution of the spectrum because for predictive maintenance, the importance of wideband harmonics cannot be denied. For this purpose, the algorithm is first implemented on FEM-based simulation signals with a low sampling frequency. In Figure 9, it can be seen that even at a high step size with a sampling frequency of 4 kHz, the spectrum counting the integral number of cycles increases the resolution significantly without the need for any truncating window. Moreover, the effect of communication channel-based data discontinuities and their correction is shown in Figure 10.

Table 1. Fault definition frequencies.

Fault	Modulating Frequencies
Broken Rotor Bars	$f_{BR} = f_s \pm 2ksf_s, \;\; k = 1,2,3,\ldots$
Principal slotting harmonic (PSH) and Eccentricity	$f_{ecce} = \left[(kn_b \pm n_d)\left(\frac{1-s}{p}\right) \pm v\right] fs$ More precisely: $f_{ecce} = \left[1 \pm k\left(\frac{1-s}{p}\right)\right] fs$ $f_{ecce} = f_s \pm kf_r, \;\; k = 1,2,3,\ldots$

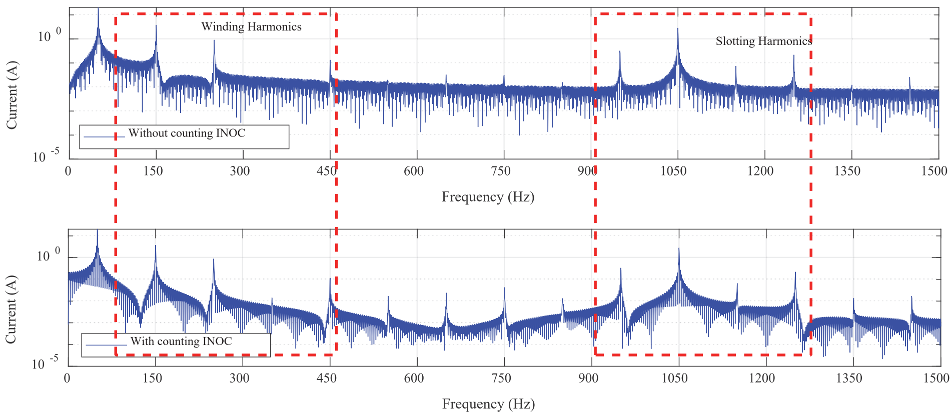


Figure 9. The simulated stator current spectrum showing stator winding and slotting harmonics before and after counting integral number of cycles (INOC).

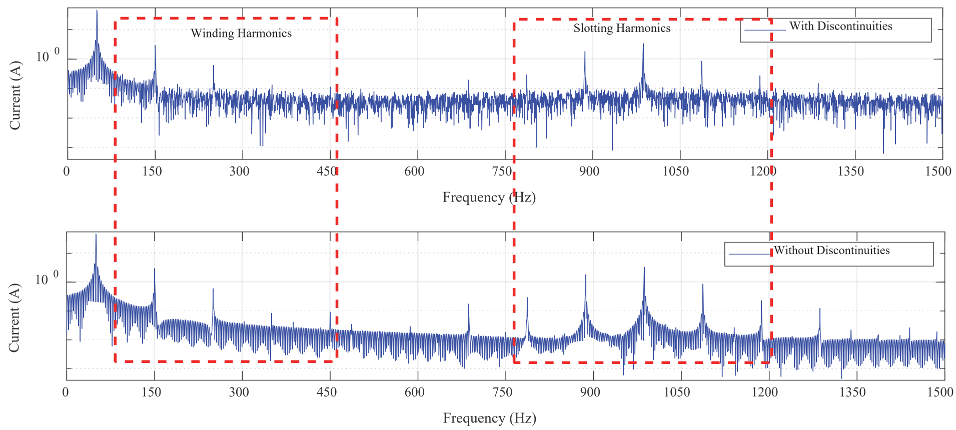


Figure 10. The effect of signal discontinuities on the spectrum resolution.

6.2. Practical Results

For practical investigations, two similar machines were connected back-to-back. One machine works as a loading machine, while the other was used as a testing motor where the healthy and broken rotor bar carrying rotor were tested. Table 2 shows the nominal parameters of the machine under investigation. Figures 11 and 12 show the improvement in the spectrum resolution by removing the fractional parts of the signal and data discontinuities without any truncating window. The tiny broken rotor bar harmonics near the strong supply and spatial harmonics became well legible.

Table 2. The machine specifications.

Parameter	Symbol	Value
Number of poles	P	4
Number of phases	φ	3
Connection	-	Star
Stator slots	Ns	48, non-skewed
Rotor bars	Nb	40, skewed
Rated voltage	V	333 V @50 Hz
Rated power	Pr	18 kW @50 Hz

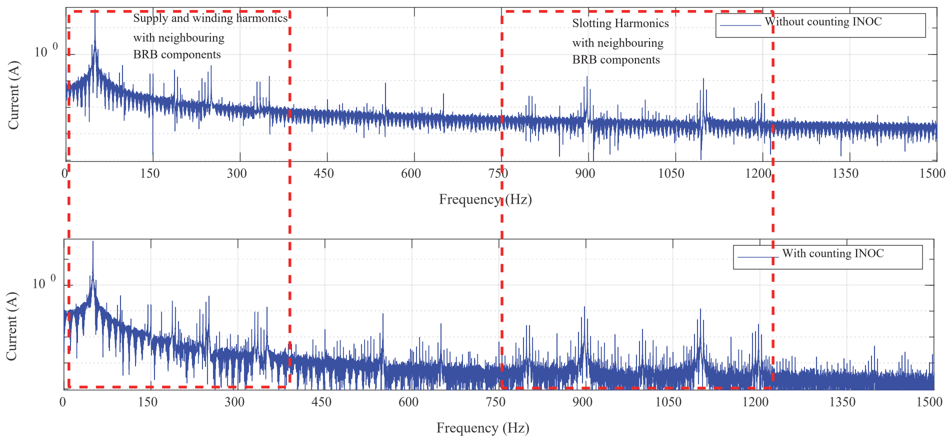


Figure 11. The practical stator current spectrum showing stator winding, slotting, and broken rotor bar-based harmonics before and after counting the integral number of cycles (INOC).

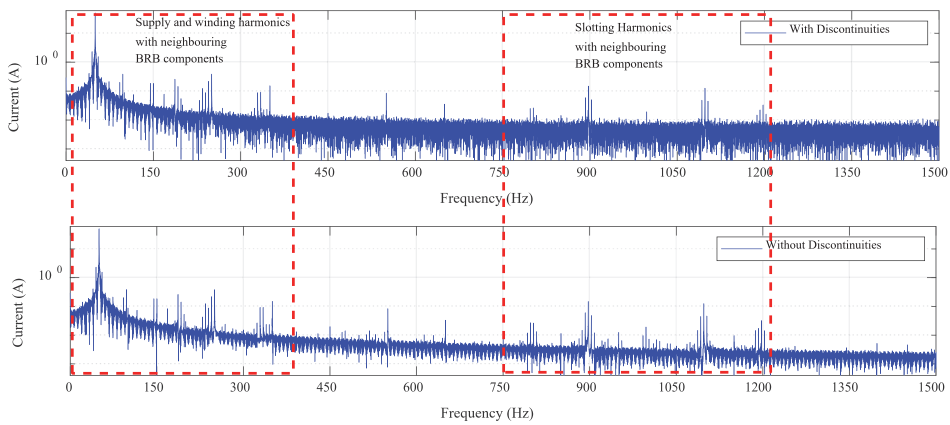


Figure 12. The practical stator current spectrum showing stator winding, slotting, and broken rotor bar-based harmonics with and without discontinuities.

The frequency of slotting harmonics in the current spectrum in comparison with their expected frequency according to the equations given in Table 1 as a function of slip is shown in Table 3. It is clear that the amplitude of those harmonics decreases with the decreasing slip, which makes their detection difficult when the machine is working under low or no-load conditions.

Table 3. The rotor slot harmonics (RSH).

Slip	Theoretical RSH1	Theoretical RSH2	RSH1 (Hz)	RSH2 (Hz)	RSH1 (A)	RSH2 (A)
0.0030	947	1047	946.7	1046.8	0.00042	0.0005
0.035	915	1015	914.76	1014.76	0.00185	0.0008
0.05	900	1000	899.2	999.2	0.0021	0.0007

7. Conclusions

Low sampling frequency, fractional parts of the signal at starting and ending, and data discontinuities in the time domain can lead to spectral leakage in the frequency domain when applying the FFT (Fast Fourier Transform) algorithm. Spectral leakage refers to the effect where energy from a signal at one frequency "leaks" into other nearby frequencies, creating artifacts in the spectrum that are not present in the original signal. There can also be interruptions between the transmitted signals due to limitations of the hardware used or because of a loss of network. This can also lead to data loss or the receiving signal missing some harmonics and having some junk values in between. This can further lead to an incorrect analysis of the collected signal, and, in some cases, it might even be more fatal, i.e., could lead to the machine being damaged if the issue occurs in the case of monitoring an electrical machine.

One way to mitigate these effects is by applying a window function to the data before performing the FFT. A window function can smooth out the signal at the edges of the analysis window, reducing the abrupt changes and thus the spectral leakage. However, even with a window function, some level of spectral leakage may still be present, depending on the characteristics of the signal and the choice of window function. Moreover, the application of advanced signal processing techniques makes it computationally complex for low power data acquisition and processing devices.

This paper shows how a simple algorithm can improve the spectrum resolution by removing the above-mentioned problems.

Author Contributions: Conceptualization, B.A., H.A.R. and T.V.; methodology, B.A. and H.A.R.; software, B.A. and H.A.R.; validation, T.V., A.K. and R.P.; formal analysis, V.K.H.; investigation, B.A.; resources, T.V.; data curation, B.A.; writing—original draft preparation, B.A. and H.A.R.; writing—review and editing, T.V.; visualization, A.K.; supervision, V.K.H.; project administration, V.K.H. and T.V.; funding acquisition, T.V. and V.K.H. All authors have read and agreed to the published version of the manuscript.

Funding: The "Industrial Internet methods for electrical energy conversion systems monitoring and diagnostics" benefits from a 993.000€ grant from Iceland, Liechtenstein, and Norway, through the EEA Grants. The aim of the project is to provide research in the field of energy conversion systems and to develop artificial intelligence and virtual emulator-based prognostic and diagnostic methodologies for these systems. Project contract with the Research Council of Lithuania (LMTLT) No is S-BMT-21-5 (LT08-2-LMT-K-01-040).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available within the article.

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

References

1. Skarmoutsos, G.A.; Gyftakis, K.N.; Mueller, A.M. Analytical Prediction of the MCSA Signatures Under Dynamic Eccentricity in PM Machines with Concentrated Non-Overlapping Windings. *IEEE Trans. Energy Convers.* **2021**, *37*, 1011–1019. [CrossRef]
2. Garcia-Bracamonte, J.E.; Ramirez-Cortes, J.M.; de Jesus Rangel-Magdaleno, J.; Gomez-Gil, P.; Peregrina-Barreto, H.; Alarcon-Aquino, V. An Approach on MCSA-Based Fault Detection Using Independent Component Analysis and Neural Networks. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 1353–1361. [CrossRef]

3. Asad, B.; Vaimann, T.; Belahcen, A.; Kallaste, A.; Rassõlkin, A.; Iqbal, M.N. Broken rotor bar fault detection of the grid and inverter-fed induction motor by effective attenuation of the fundamental component. *IET Electr. Power Appl.* **2019**, *13*, 2005–2014. [CrossRef]
4. Hang, J.; Hu, Q.; Sun, W.; Ren, X.; Ding, S.; Huang, Y.; Hua, W. A Voltage-Distortion-Based Method for Robust Detection and Location of Interturn Fault in Permanent Magnet Synchronous Machine. *IEEE Trans. Power Electron.* **2022**, *37*, 11174–11186. [CrossRef]
5. Hu, R.; Wang, J.; Mills, A.R.; Chong, E.; Sun, Z. High-Frequency Voltage Injection Based Stator Interturn Fault Detection in Permanent Magnet Machines. *IEEE Trans. Power Electron.* **2020**, *36*, 785–794. [CrossRef]
6. Irhoumah, M.; Pusca, R.; Lefevre, E.; Mercier, D.; Romary, R. Detection of the Stator Winding Inter-Turn Faults in Asynchronous and Synchronous Machines Through the Correlation Between Harmonics of the Voltage of Two Magnetic Flux Sensors. *IEEE Trans. Ind. Appl.* **2019**, *55*, 2682–2689. [CrossRef]
7. Yang, M.; Chai, N.; Liu, Z.; Ren, B.; Xu, D. Motor Speed Signature Analysis for Local Bearing Fault Detection with Noise Cancellation Based on Improved Drive Algorithm. *IEEE Trans. Ind. Electron.* **2019**, *67*, 4172–4182. [CrossRef]
8. Hu, K.; Liu, Z.; Tasiu, I.A.; Chen, T. Fault Diagnosis and Tolerance with Low Torque Ripple for Open-Switch Fault of IM Drives. *IEEE Trans. Transp. Electrif.* **2020**, *7*, 133–146. [CrossRef]
9. Tian, P.; Antonino-Daviu, J.A.; Platero, C.A.; Dunai, L.D. Detection of Field Winding Faults in Synchronous Motors via Analysis of Transient Stray Fluxes and Currents. *IEEE Trans. Energy Convers.* **2020**, *36*, 2330–2338. [CrossRef]
10. Gurusamy, V.; Baruti, K.H.; Zafarani, M.; Lee, W.; Akin, B. Effect of Magnets Asymmetry on Stray Magnetic Flux Based Bearing Damage Detection in PMSM. *IEEE Access* **2021**, *9*, 68849–68860. [CrossRef]
11. Filho, P.C.M.L.; Baccarini, L.M.R.; Batista, F.B.; Araujo, A.C. Orbit Analysis froma Stray Flux Full Spectrum for Induction Machine Fault Detection. *IEEE Sensors J.* **2021**, *21*, 16152–16161. [CrossRef]
12. Liu, X.; Miao, W.; Xu, Q.; Cao, L.; Liu, C.; Pong, P.W.T. Inter-Turn Short-Circuit Fault Detection Approach for Permanent Magnet Synchronous Machines Through Stray Magnetic Field Sensing. *IEEE Sensors J.* **2019**, *19*, 7884–7895. [CrossRef]
13. Gurusamy, V.; Bostanci, E.; Li, C.; Qi, Y.; Akin, B. A Stray Magnetic Flux-Based Robust Diagnosis Method for Detection and Location of Interturn Short Circuit Fault in PMSM. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 6045–6057. [CrossRef]
14. Park, Y.; Choi, H.; Bin Lee, S.; Gyftakis, K.N. Search Coil-Based Detection of Nonadjacent Rotor Bar Damage in Squirrel Cage Induction Motors. *IEEE Trans. Ind. Appl.* **2020**, *56*, 4748–4757. [CrossRef]
15. Reda, K.; Yan, Y. Vibration Measurement of an Unbalanced Metallic Shaft Using Electrostatic Sensors. *IEEE Trans. Instrum. Meas.* **2018**, *68*, 1467–1476. [CrossRef]
16. Song, L.; Wang, H.; Chen, P. Vibration-Based Intelligent Fault Diagnosis for Roller Bearings in Low-Speed Rotating Machinery. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 1887–1899. [CrossRef]
17. Samanta, A.K.; Routray, A.; Khare, S.R.; Naha, A. Minimum Distance-Based Detection of Incipient Induction Motor Faults Using Rayleigh Quotient Spectrum of Conditioned Vibration Signal. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–11. [CrossRef]
18. Rafaq, M.S.; Lee, H.; Park, Y.; Bin Lee, S.; Fernandez, D.; Diaz-Reigosa, D.; Briz, F. A Simple Method for Identifying Mass Unbalance Using Vibration Measurement in Permanent Magnet Synchronous Motors. *IEEE Trans. Ind. Electron.* **2021**, *69*, 6441–6444. [CrossRef]
19. Kudelina, K.; Asad, B.; Vaimann, T.; Belahcen, A.; Rassõlkin, A.; Kallaste, A.; Lukichev, D.V. Bearing Fault Analysis of BLDC Motor for Electric Scooter Application. *Designs* **2020**, *4*, 42. [CrossRef]
20. Choudhary, A.; Goyal, D.; Letha, S.S. Infrared Thermography-Based Fault Diagnosis of Induction Motor Bearings Using Machine Learning. *IEEE Sensors J.* **2020**, *21*, 1727–1734. [CrossRef]
21. Mohammed, A.; Melecio, J.I.; Djurovic, S. Open-Circuit Fault Detection in Stranded PMSM Windings Using Embedded FBG Thermal Sensors. *IEEE Sensors J.* **2019**, *19*, 3358–3367. [CrossRef]
22. Shi, Y.; Wang, J.; Hu, R.; Wang, B. Electromagnetic and Thermal Behavior of a Triple Redundant 9-Phase PMASynRM with Insulation Deterioration Fault. *IEEE Trans. Ind. Appl.* **2020**, *56*, 6374–6383. [CrossRef]
23. Mohammed, A.; Melecio, J.I.; Djurovic, S. Stator Winding Fault Thermal Signature Monitoring and Analysis by In Situ FBG Sensors. *IEEE Trans. Ind. Electron.* **2018**, *66*, 8082–8092. [CrossRef]
24. Lucas, G.B.; de Castro, B.A.; Rocha, M.A.; Andreoli, A.L. A New Acoustic Emission-Based Approach for Supply Disturbances Evaluation in Three-Phase Induction Motors. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 1–10. [CrossRef]
25. Parvathi Sangeetha, B.; Hemamalini, S. Rational-Dilation Wavelet Transform Based Torque Estimation from Acoustic Signals for Fault Diagnosis in a Three-Phase Induction Motor. *IEEE Trans. Ind. Inform.* **2018**, *15*, 3492–3501. [CrossRef]
26. Liu, F.; Wu, R.; Teng, F.; Liu, Y.; Lu, S.; Ju, B.; Cao, Z. A Two-Stage Learning Model for Track-Side Acoustic Bearing Fault Diagnosis. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–12. [CrossRef]
27. Liu, Z.; Yang, B.; Wang, X.; Zhang, L. Acoustic Emission Analysis for Wind Turbine Blade Bearing Fault Detection Under Time-Varying Low-Speed and Heavy Blade Load Conditions. *IEEE Trans. Ind. Appl.* **2021**, *57*, 2791–2800. [CrossRef]
28. Sun, C.; Liu, W.; Han, X.; Zhang, X.; Jiao, N.; Mao, S.; Wang, R.; Guan, Y. High-Frequency Voltage Injection-Based Fault Detection of a Rotating Rectifier for a Wound-Rotor Synchronous Starter/Generator in the Stationary State. *IEEE Trans. Power Electron.* **2021**, *36*, 13423–13433. [CrossRef]

29. Zamudio-Ramirez, I.; Ramirez-Nunez, J.A.; Antonino-Daviu, J.; Osornio-Rios, R.A.; Quijano-Lopez, A.; Razik, H.; Romero-Troncoso, R.J. Automatic diagnosis of electromechanical faults in induction motors based on the transient analysis of the stray flux via MUSIC methods. *IEEE Trans. Ind. Appl.* **2020**, *56*, 3604–3613. [CrossRef]
30. Gyftakis, K.N. A Comparative Investigation of Interturn Faults in Induction Motors Suggesting a Novel Transient Diagnostic Method Based on the Goerges Phenomenon. *IEEE Trans. Ind. Appl.* **2021**, *58*, 304–313. [CrossRef]
31. Park, Y.; Choi, H.; Shin, J.; Park, J.; Bin Lee, S.; Jo, H. Airgap Flux Based Detection and Classification of Induction Motor Rotor and Load Defects During the Starting Transient. *IEEE Trans. Ind. Electron.* **2020**, *67*, 10075–10084. [CrossRef]
32. Asad, B.; Vaimann, T.; Belahcen, A.; Kallaste, A.; Rassõlkin, A.; Ghafarokhi, P.; Kudelina, K. Transient Modeling and Recovery of Non-Stationary Fault Signature for Condition Monitoring of Induction Motors. *Appl. Sci.* **2021**, *11*, 2806. [CrossRef]
33. *IEEE Std 1658-2011*; IEEE Standard for Terminology and Test Methods of Digital-to-Analog Converter Devices. IEEE: Pisataway, NJ, USA, 2012. [CrossRef]
34. *P1057^{TM}/D8*; IEEE Standard for Digitizing Waveform Recorders. IEEE: New York, NY, USA, 2018.
35. Li, R.; Zhuang, L.; Li, Y.; Shen, C. Intelligent Bearing Fault Diagnosis Based on Scaled Ramanujan Filter Banks in Noisy Environments. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [CrossRef]
36. Gao, M.; Yu, G.; Wang, T. Impulsive Gear Fault Diagnosis Using Adaptive Morlet Wavelet Filter Based on Alpha-Stable Distribution and Kurtogram. *IEEE Access* **2019**, *7*, 72283–72296. [CrossRef]
37. Atta, M.E.E.-D.; Ibrahim, D.K.; Gilany, M.I. Broken Bar Faults Detection Under Induction Motor Starting Conditions Using the Optimized Stockwell Transform and Adaptive Time–Frequency Filter. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–10. [CrossRef]
38. Gong, W.; Chen, H.; Zhang, Z.; Zhang, M.; Gao, H. A Data-Driven-Based Fault Diagnosis Approach for Electrical Power DC-DC Inverter by Using Modified Convolutional Neural Network with Global Average Pooling and 2-D Feature Image. *IEEE Access* **2020**, *8*, 73677–73697. [CrossRef]
39. Hou, B.; Wang, D.; Chen, Y.; Wang, H.; Peng, Z.; Tsui, K.-L. Interpretable online updated weights: Optimized square envelope spectrum for machine condition monitoring and fault diagnosis. *Mech. Syst. Signal Process.* **2022**, *169*, 108779. [CrossRef]
40. Burriel-Valencia, J.; Puche-Panadero, R.; Martinez-Roman, J.; Sapena-Bano, A.; Pineda-Sanchez, M. Fault Diagnosis of Induction Machines in a Transient Regime Using Current Sensors with an Optimized Slepian Window. *Sensors* **2018**, *18*, 146. [CrossRef]
41. Cong, F.; Chen, J.; Dong, G.; Zhao, F. Short-time matrix series based singular value decomposition for rolling bearing fault diagnosis. *Mech. Syst. Signal Process.* **2013**, *34*, 218–230. [CrossRef]
42. Kang, M.; Kim, J.-M. Singular value decomposition based feature extraction approaches for classifying faults of induction motors. *Mech. Syst. Signal Process.* **2013**, *41*, 348–356. [CrossRef]
43. Li, H.; Liu, T.; Wu, X.; Chen, Q. A Bearing Fault Diagnosis Method Based on Enhanced Singular Value Decomposition. *IEEE Trans. Ind. Inform.* **2020**, *17*, 3220–3230. [CrossRef]
44. Yi, C.; Ran, L.; Tang, J.; Jin, H.; Zhuang, Z.; Zhou, Q.; Lin, J. An Improved Sparse Representation Based on Local Orthogonal Matching Pursuit for Bearing Compound Fault Diagnosis. *IEEE Sens. J.* **2022**, *22*, 21911–21923. [CrossRef]
45. Morales-Perez, C.; Rangel-Magdaleno, J.; Peregrina-Barreto, H.; Amezquita-Sanchez, J.P.; Valtierra-Rodriguez, M. Incipient Broken Rotor Bar Detection in Induction Motors Using Vibration Signals and the Orthogonal Matching Pursuit Algorithm. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 2058–2068. [CrossRef]
46. Wang, L.; Cai, G.; You, W.; Huang, W.; Zhu, Z. Transients Extraction Based on Averaged Random Orthogonal Matching Pursuit Algorithm for Machinery Fault Diagnosis. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 3237–3248. [CrossRef]
47. El Bouchikhi, E.H.; Choqueuse, V.; Benbouzid, M. Induction machine faults detection using stator current parametric spectral estimation. *Mech. Syst. Signal Process.* **2015**, *52–53*, 447–464. [CrossRef]
48. Avalos, G.; Aguayo, S.; Rangel-Magdaleno, J.; Paternina, M. Bearing fault detection in induction motors using digital Taylor-Fourier transform. In Proceedings of the 2022 International Conference on Electrical Machines (ICEM), Valencia, Spain, 5–8 September 2022; pp. 1830–1835. [CrossRef]
49. Boudinar, A.H.; Benouzza, N.; Bendiabdellah, A.; Khodja, M.-E. Induction Motor Bearing Fault Analysis Using a Root-MUSIC Method. *IEEE Trans. Ind. Appl.* **2016**, *52*, 3851–3860. [CrossRef]
50. Elbouchikhi, E.; Choqueuse, V.; Benbouzid, M. Induction machine bearing faults detection based on a multi-dimensional MUSIC algorithm and maximum likelihood estimation. *ISA Trans.* **2016**, *63*, 413–424. [CrossRef]
51. Xu, S.; Dai, H.; Feng, L.; Chen, H.; Chai, Y.; Zheng, W.X. Fault Estimation for Switched Interconnected Nonlinear Systems with External Disturbances via Variable Weighted Iterative Learning. *IEEE Trans. Circuits Syst. II Express Briefs* **2023**. [CrossRef]
52. Huang, N.; Chen, Q.; Cai, G.; Xu, D.; Zhang, L.; Zhao, W. Fault Diagnosis of Bearing in Wind Turbine Gearbox Under Actual Operating Conditions Driven by Limited Data with Noise Labels. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 1–10. [CrossRef]
53. Xu, B.; Sun, L.; Xu, L.; Xu, G. Improvement of the Hilbert Method via ESPRIT for Detecting Rotor Fault in Induction Motors at Low Slip. *IEEE Trans. Energy Convers.* **2013**, *28*, 225–233. [CrossRef]
54. Liang, X.; Ali, M.Z.; Zhang, H. Induction Motors Fault Diagnosis Using Finite Element Method: A Review. *IEEE Trans. Ind. Appl.* **2019**, *56*, 1205–1217. [CrossRef]
55. Marfoli, A.; Bolognesi, P.; Papini, L.; Gerada, C. Mid-Complexity Circuital Model of Induction Motor with Rotor Cage: A Numerical Resolution. In Proceedings of the 2018 XIII International Conference on Electrical Machines (ICEM), Alexandroupoli, Greece, 3–6 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 277–283. [CrossRef]

56. Faiz, J.; Ojaghi, M. Unified winding function approach for dynamic simulation of different kinds of eccentricity faults in cage induction machines. *IET Electr. Power Appl.* **2009**, *3*, 461–470. [CrossRef]
57. Asad, B.; Vaimann, T.; Belahcen, A.; Kallaste, A.; Rassõlkin, A.; Iqbal, M.N. Modified winding function-based model of squirrel cage induction motor for fault diagnostics. *IET Electr. Power Appl.* **2020**, *14*, 1722–1734. [CrossRef]
58. Li, C.; Wang, X.; Liu, F.; Ren, J.; Xing, Z.; Gu, X. Analysis of Permanent Magnet-assisted Synchronous Reluctance Motor Based on Equivalent Reluctance Network Model. *CES Trans. Electr. Mach. Syst.* **2022**, *6*, 135–144. [CrossRef]
59. Shen, M.; Pfister, P.-D.; Tang, C.; Fang, Y. A Hybrid Model of Permanent-Magnet Machines Combining Fourier Analytical Model with Finite Element Method, Taking Magnetic Saturation into Account. *IEEE Trans. Magn.* **2020**, *57*, 1–5. [CrossRef]
60. Asad, B.; Vaimann, T.; Belahcen, A.; Kallaste, A.; Rassõlkin, A.; Iqbal, M. The Cluster Computation-Based Hybrid FEM–Analytical Model of Induction Motor for Fault Diagnostics. *Appl. Sci.* **2020**, *10*, 7572. [CrossRef]
61. Dong, N.; Roychowdhury, J. General-Purpose Nonlinear Model-Order Reduction Using Piecewise-Polynomial Representations. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2008**, *27*, 249–264. [CrossRef]
62. Kassis, M.T.; Kabir, M.; Xiao, Y.Q.; Khazaka, R. Passive Reduced Order Macromodeling Based on Loewner Matrix Interpolation. *IEEE Trans. Microw. Theory Tech.* **2016**, *64*, 2423–2432. [CrossRef]
63. Zhai, Y.; Vu-Quoc, L. Analysis of Power Magnetic Components with Nonlinear Static Hysteresis: Proper Orthogonal Decomposition and Model Reduction. *IEEE Trans. Magn.* **2007**, *43*, 1888–1897. [CrossRef]
64. Far, M.F.; Martin, F.; Belahcen, A.; Montier, L.; Henneron, T. Orthogonal Interpolation Method for Order Reduction of a Synchronous Machine Model. *IEEE Trans. Magn.* **2017**, *54*, 1–6. [CrossRef]
65. Bai, Z. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Appl. Numer. Math.* **2002**, *43*, 9–44. [CrossRef]
66. Huang, S.; Tan, K.K. Fault Simulator Based on a Hardware-in-the-Loop Technique. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 1135–1139. [CrossRef]
67. Nadarajan, S.; Panda, S.K.; Bhangu, B.; Gupta, A.K. Online Model-Based Condition Monitoring for Brushless Wound-Field Synchronous Generator to Detect and Diagnose Stator Windings Turn-to-Turn Shorts Using Extended Kalman Filter. *IEEE Trans. Ind. Electron.* **2016**, *63*, 3228–3241. [CrossRef]
68. Bachir, S.; Tnani, S.; Trigeassou, J.-C.; Champenois, G. Diagnosis by parameter estimation of stator and rotor faults occurring in induction machines. *IEEE Trans. Ind. Electron.* **2006**, *53*, 963–973. [CrossRef]
69. Zhang, S.; Dong, H.; Maschek, U.; Song, H. A digital-twin-assisted fault diagnosis of railway point machine. In Proceedings of the 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI), IEEE, DTPI, Beijing, China, 15 July–15 August 2021; pp. 430–433. [CrossRef]
70. Bui, V.P.; Chadebec, O.; Rouve, L.-L.; Coulomb, J.-L. Noninvasive Fault Monitoring of Electrical Machines by Solving the Steady-State Magnetic Inverse Problem. *IEEE Trans. Magn.* **2008**, *44*, 1050–1053. [CrossRef]
71. Huang, Y.; Zhao, M.; Zhang, J.; Lu, M. The Hall Sensors Fault-Tolerant for PMSM Based on Switching Sensorless Control with PI Parameters Optimization. *IEEE Access* **2022**, *10*, 114048–114059. [CrossRef]
72. Zarch, M.G.; Alipouri, Y.; Poshtan, J. Fault Detection Based on Online Probability Density Function Estimation. *Asian J. Control.* **2016**, *18*, 2193–2202. [CrossRef]
73. Guo, J.; Queval, L.; Roucaries, B.; Vido, L.; Liu, L.; Trillaud, F.; Berriaud, C. Nonlinear Current Sheet Model of Electrical Machines. *IEEE Trans. Magn.* **2019**, *56*, 1–4. [CrossRef]
74. Frosini, L.; Harlisca, C.; Szabo, L. Induction Machine Bearing Fault Detection by Means of Statistical Processing of the Stray Flux Measurement. *IEEE Trans. Ind. Electron.* **2014**, *62*, 1846–1854. [CrossRef]
75. Lang, W.; Hu, Y.; Gong, C.; Zhang, X.; Xu, H.; Deng, J. Artificial Intelligence-Based Technique for Fault Detection and Diagnosis of EV Motors: A Review. *IEEE Trans. Transp. Electrif.* **2021**, *8*, 384–406. [CrossRef]
76. Wu, M.; Chen, D.; Chen, G. New Spectral Leakage-Removing Method for Spectral Testing of Approximate Sinusoidal Signals. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 1296–1306. [CrossRef]

Publication III

Raja, H. A.; Raval, H.; Vaimann, T.; Rassõlkin, A.; Kallaste, A.; Belahcen A. (2022). Cost-efficient real-time condition monitoring and fault diagnostics system for BLDC motor using IoT and Machine learning. International Conference on Diagnostics in Electrical Engineering (Diagnostika), 6. - 8. 9. 20226. - 8. 9. 2022, Pilsen, Czech Republic. IEEE.

Cost-efficient real-time condition monitoring and fault diagnostics system for BLDC motor using IoT and Machine learning

2022 International Conference on Diagnostics in Electrical Engineering (Diagnostika) | 978-1-6654-8082-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/Diagnostika55131.2022.9905102

Hadi Ashraf Raja
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
haraja@taltech.ee

Hardik Raval
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
harava@taltech.ee

Toomas Vaimann
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
toomas.vaimann@taltech.ee

Ants Kallaste
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
ants.kallaste@taltech.ee

Anton Rassõlkin
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
anton.rassolkin@taltech.ee

Anouar Belahcen
Department of Electrical Engineering and Automation
Aalto University
Espoo, Finland
anouar.belahcen@taltech.ee

***Abstract*—A cost-efficient condition monitoring and fault diagnostic system are presented in this paper using the Internet of Things and machine learning. Most condition monitoring systems nowadays are either costly or used to monitor current values without emphasizing the analysis part. On the other hand, predictive maintenance of different electrical machines, including BLDC motors, is becoming the need of the hour. It reduces the cost needed for maintenance and can also be used to evade more significant faults in the machine. The data is transmitted in real-time using a data acquisition system onto the cloud, which is further processed to determine if there is a chance of any fault occurring in the motor. A short comparison of the results of different machine learning algorithms is also discussed related to predictive maintenance.**

Keywords—condition monitoring, fault diagnostic, IoT, Internet of Things

I. INTRODUCTION

Brushless Direct Current motors (BLDC) have a vast implementation in industrial applications like electric vehicles, military, medical and others because of the easiness to control their speed and position. Therefore, maintenance of BLDC motos is of utmost importance as it is an essential part of these setups. Untimely failure of BLDC motor can halt operations and may cause irreversible losses. Plant engineers frequently inspect BLDC motors and the entire setup after a specific time, also termed scheduled maintenance, but these repeated checks are expensive. These scheduled maintenance checks can cost from 15% - 40% of the production cost and still might not prevent a more considerable expense in case of some component failure. Moreover, most of the time, there is no backup plan to check on the condition of the BLDC motor in between those checks, which might result in a bigger disaster if a fault occurs after a scheduled check as a person cannot predict the occurrence of a fault [1][2]. Therefore, automatic condition monitoring systems are being developed to stay ahead of the failures and predict or at least check on the machine in real-time [3][4].

With the advent of the industrial revolution in the form of Industry 4.0, more research is being done on the automation of the industry, including monitoring and predicting faults [5]. The continuous monitoring of electrical machines can help prevent the escalation of a fault in the machine, which can not only cut costs but also help keep the system loss to a minimum. [6][7][8]. These condition monitoring systems need stable data acquisition systems that can be easily used with different electrical machines. These systems are still evolving with the advancement in sensors and circuits [9]. Most of the earliest systems developed for data acquisition and monitoring electrical machines are in SCADA [10], which are heavy and expensive but not feasible with the new trend of portable devices. It is also difficult and expensive to add some sensors into the already deployed system and any changes can result in further costs [11]. Therefore, researchers are looking for better ways to replace these systems.

This paper presents an alternate approach to such systems with a cost-efficient data acquisition part and enables the end-user to view the transmitted data in real-time. It also presents a method to detect faults using a machine learning training algorithm, although that is still done offline. However, future prospects of this study include the implementation of the system in the cloud.

II. RELATED WORKS

The Industrial Revolution gave way to multiple studies related to fault detection and monitoring of electrical machines, including BLDC motors [7][12]. These studies are based on vibration signals to differentiate between healthy and faulty bearings using frequency components [13]. In this research, the authors provide a nonlinear model [14] for fault diagnosis of BLDC motor based on the vibration frequency spectrum. Whereas more proof has been provided that vibration signals can discover bearing issues related to variable speed [4]. Similarly, multiple other investigations into faults based on the stator of BLDC motors have been done [15][16][17]. However, most of this analysis has been done offline and not in real-time.

The recent advancement in information technology has also opened up ways for remote monitoring of electrical machines using the Internet of Things (IoT). Different systems are being developed to check the health of electrical machines [18] in real-time and predict faults to reduce costs [19]. This has helped move towards a new era away from

978-1-6654-8082-6/22/$31.00 ©2022 IEEE

heavy machines and expensive equipment. Researchers adapted the recent methods developed on IoT [20][21] more commonly due to ease of operation and access. Although this system is still in development and has a slow sample rate, further research is still ongoing with the increase in the sample rate and decrease in data loss [22].

Therefore, this paper presents the approach based on the data acquisition method using microcontroller boards which are then pushed to the cloud and can be monitored in real-time. The signal considered here is the current signal rather than the vibration signal and the diagnosis is made in the frequency domain. A model is also trained to detect faults using up-to-date machine learning algorithms.

III. Methodology And Results

This research can be divided into three parts, the data acquisition system at the start, detection of faults in the cloud using the similarity principle and a machine learning based trained model for fault detection. First, the data is transmitted from the esp32 board to the cloud, where it is checked for fault in real-time with a similarity function. After that, the approach towards the signal is slightly different as the current signal is considered for the training and detection of faults instead of vibration signals. The experimental setup is shown in Fig. 1.

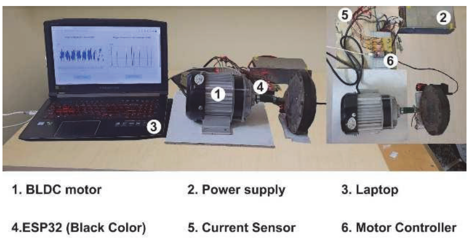


Fig. 1. Experimental Setup

The experimental setup includes a BLDC motor and a controller for speed, a current sensor, and an ESP32 microcontroller board. The data from the current sensor is read through the ESP32 board, calibrated and sent over to the cloud and the laptop for checking purposes. The transmitted data is then saved in a database on the cloud and shown at the front using a user interface. The general block diagram of the setup is shown in Fig. 2.

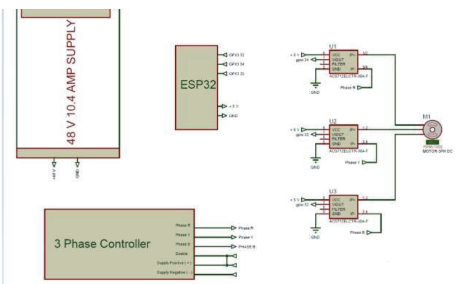


Fig. 2. Block Diagram

This research leading to these results has received funding from the PSG453, Digital twin for propulsion drive of autonomous electric vehicles" and ETAG21001, "Industrial internet methods for electrical energy conversion systems monitoring and diagnostics".

The data is then preprocessed online to convert the time domain into the frequency domain and compared with the already collected data in the frequency domain for healthy and faulty signals of BLDC motor. The similarity criteria threshold is taken to be around 70%. If the similarity is between 70% – 80%, then there is a mild chance of a fault occurring in the electrical machine over some time. If it is between 80% - 90%, there is a high possibility that the fault will occur shortly and a check should be done on the machine to be sure it is still in a good state. If the similarity is above 90%, then there is a high probability that the fault has occurred and the motor is already faulty. This data is presented in a user interface at the front for an end user to be relayed quickly and in real-time. An example of the user interface is shown in Fig. 3.

Fig. 3. Dashboard example

The faults considered here are the inner and outer faults of bearings. A machine learning based model was also trained to detect faults in real-time. However, at the time being, it is still offline and will be implemented online next. The data was collected through Dewetron in bulk to train the machine learning model for both healthy and faulty BLDC motors. The current signal was further preprocessed and its frequency spectrum was considered while training for a machine learning based algorithm. Fig. 4, Fig. 5 and Fig. 6 show the three-phase BLDC motor current Fast Fourier Transform for healthy, inner fault and outer fault signals, respectively.

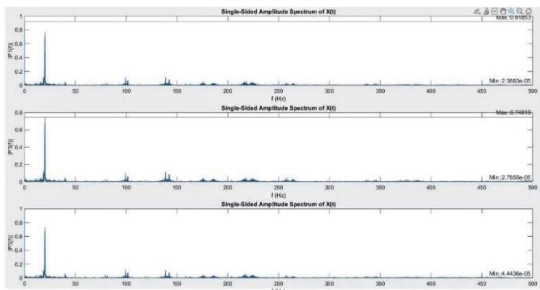


Fig. 4. FFT of BLDC Healthy Signal

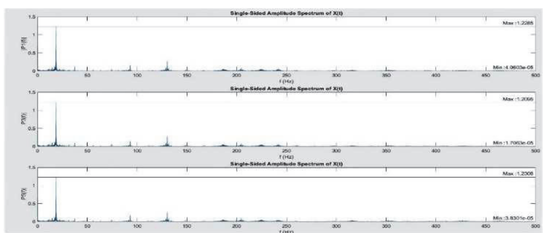


Fig. 5. FFT of BLDC Inner Fault

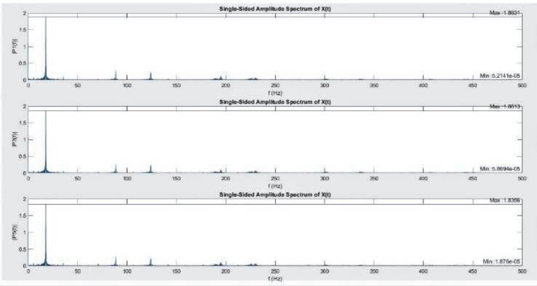


Fig. 6. FFT of BLDC Outer Fault

As it can be seen that in each case of healthy, inner fault and outer fault signals, the frequency components with prominent values differ. These components are taken as the fundamental difference between these three signals and are then used to train the machine learning model and are then validated on a data set. The frequency components are further processed to remove any low amplitude components and further enhance the difference between them shown in Fig. 7, Fig. 8 and Fig. 9.

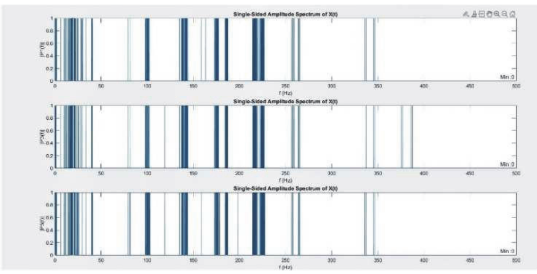


Fig. 7. Processed BLDC Healthy Signal

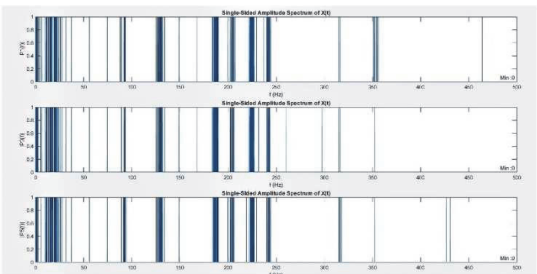


Fig. 8. Processed BLDC Inner Fault

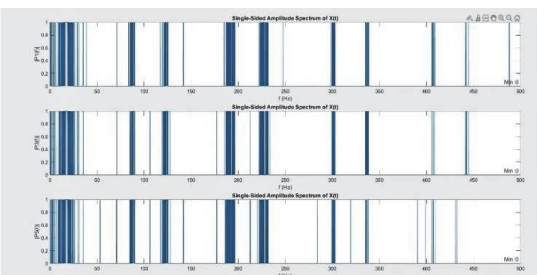


Fig. 9. Processed BLDC Outer Fault

After processing the components further, the data is then used to train different machine learning algorithms. A data set containing 100,000 samples were used for training purposes with a blind validation for 20,000 data sets. As a result, four different types of machine learning algorithms were selected for the training of the model, and the result of their accuracy is shown in Table 1.

TABLE I. COMPARISON OF MACHINE LEARNING ALGORITHMS

Type of Algorithm	Accuracy	Error
Decision Tree	87%	13%
Random Forest Tree	93%	7%
Super Vector Control	97%	3%
K nearest Neighbor	40%	60%

The above table shows that the Super Vector Control Algorithm performed the best among the four to classify the incoming signal to detect a fault. The random Forest Tree algorithm also gives a high accuracy, whereas K nearest neighbor is the worst.

IV. CONCLUSION

The advancement in information technology has pushed other fields to work towards integrating their fields with it. Similarly, IoT and cloud computation is becoming the new trend in industrial applications and the industry 4.0 revolution. As a result, more emphasis is being done on the automation of processes. This will also help detect faults in real-time and contain any unpreceded losses if present in machines. This can also help save up time during maintenance of the said machine as the phase or place the fault is occurring can also be identified and the maintenance team does not have to check everything to identify the source of the fault.

The proposed method here helps detect the fault in real-time using IoT and the cloud. Although the machine learning algorithm is still offline, its implementation in the cloud is not far off. The study further enhances maintenance speed and can help cut short the maintenance charges as it can help detect whether maintenance is needed. In the same way, it can also save up the cost of a significant loss if a fault occurs. The fault can be taken care of as soon as it is detected and can help prevent a bigger disaster. The setup is independent of any specific needs or equipment regarding data generation points or hardware requirements. It can be tailored to work anywhere and with any sensor as desired.

Future works for this study include implementing machine learning trained models on the cloud to detect faults in real-time. It also includes creating an algorithm to predict faults rather than only detecting such faults in an electrical machine. This way, the machine can be saved even before going into a faulty state and can help prevent further damage. It also includes the remote controlling of the electrical machine, including the BLDC motor and self-logic decision-making implementation. Hence, if the algorithm decides that a fault is expected to occur in the machine, the machine could be automatically shut down and caution is generated for the end user to send in a maintenance crew for a check-up.

REFERENCES

[1] J. K. Park, C. L. Jeong, S. T. Lee, and J. Hur, “Early detection technique for stator winding inter-turn fault in BLDC motor using input impedance,” IEEE Trans. Ind. Appl., vol. 51, no. 1, pp. 240–247, 2015. *(doi: 10.1109/TIA.2014.2330067.)*

[2] Y. Da, X. Shi, and M. Krishnamurthy, “A new approach to fault diagnostics for permanent magnet synchronous machines using electromagnetic signature analysis,” IEEE Trans. Power Electron., vol. 28, no. 8, pp. 4104–4112, 2013. *(doi: 10.1109/TPEL.2012.2227808.)*

[3] J. Podhajecki and S. Rawicki, “Numerical and experimental vibration analysis of BLDC motor,” 2019 19th Int. Symp. Electromagn. Fields Mechatronics, Electr. Electron. Eng. ISEF 2019, pp. 4–5, 2019. *(doi: 10.1109/ISEF45929.2019.9096907.)*

[4] R. M. Pindoriya, A. K. Mishra, B. S. Rajpurohit, and R. Kumar, “An Analysis of Vibration and Acoustic Noise of BLDC Motor Drive,” IEEE Power Energy Soc. Gen. Meet., vol. 2018-Augus, 2018. *(doi: 10.1109/PESGM.2018.8585750.)*

[5] R. Schmidt, M. Möhring, R. C. Härting, C. Reichstein, P. Neumaier, and P. Jozinović, “Industry 4.0 - Potentials for creating smart products: Empirical research results,” Lect. Notes Bus. Inf. Process., vol. 208, no. April, pp. 16–27, 2015. *(doi: 10.1007/978-3-319-19027-3_2.)*

[6] L. Cui, Y. Zhang, F. Zhang, J. Zhang, and S. Lee, “Vibration response mechanism of faulty outer race rolling element bearings for quantitative analysis,” J. Sound Vib., vol. 364, pp. 67–76, 2016. *(doi: 10.1016/j.jsv.2015.10.015.)*

[7] V. K. Sharma, A. Usman, and B. S. Rajpurohit, “Vibration Analysis of Brushless Permanent Magnet Motor using Numerical Methods: Part i,” 2020 IEEE Int. Power Renew. Energy Conf. IPRECON 2020, pp. 1–5, 2020. *(doi: 10.1109/IPRECON49514.2020.9315220.)*

[8] M. Cerrada, R. sanchez, C. Li, F. Pacheco, D. Cabrera, J. Valente de Oliverira, and R. E. Vasquez, “A review on data-driven fault severity assessment in rolling bearings,” Mech. Syst. Signal Process., vol. 99, pp. 169–196, 2018. *(doi: 10.1016/j.ymssp.2017.06.012.)*

[9] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus, and M. H. Ang, “Perception, planning, control, and coordination for autonomous vehicles,” Machines, 2017. *(doi: 10.3390/machines5010006.)*

[10] Y. Debbag, E. N. Yilmaz, “Internet based monitoring and control of a wind turbine via PLC,” 2015 3rd Int. Istanbul Smart Grid Congr. Fair, ICSG 2015, 2015. *(doi: 10.1109/SGCF.2015.7354935.)*

[11] I. Allafi and T. Iqbal, “Low-Cost SCADA System Using Arduino and Reliance SCADA for a Stand-Alone Photovoltaic System,” J. Sol. Energy, vol. 2018, pp. 1–8, 2018. *(doi: 10.1155/2018/3140309.)*

[12] Y. Da, X. Shi, and M. Krishnamurthy, “Health monitoring, fault diagnosis and failure prognosis techniques for brushless permanent magnet machines,” 2011 IEEE Veh. Power Propuls. Conf. VPPC 2011, pp. 1–7, 2011. *(doi: 10.1109/VPPC.2011.6043248.)*

[13] K. Kudelina, B. Asad, T. Vaimann, A. Belahcen, A. Rassolkin, and A. Kallaste, “Bearing fault analysis of bldc motor for electric scooter application,” Designs, vol. 4, no. 4, pp. 1–18, 2020. *(doi: 10.3390/designs4040042.)*

[14] S. Lu and X. Wang, “A New Methodology to Estimate the Rotating Phase of a BLDC Motor with Its Application in Variable-Speed Bearing Fault Diagnosis,” IEEE Trans. Power Electron., vol. 33, no. 4, pp. 3399–3410, 2018. *(doi: 10.1109/TPEL.2017.2703819.)*

[15] J. K. Park and J. Hur, “Detection of Inter-Turn and Dynamic Eccentricity Faults Using Stator Current Frequency Pattern in IPM-Type BLDC Motors,” IEEE Trans. Ind. Electron., vol. 63, no. 3, pp. 1771–1780, 2016. *(doi: 10.1109/TIE.2015.2499162.)*

[16] Y. Da, X. Shi, and M. Krishnamurthy, “Health monitoring, fault diagnosis and failure prognosis techniques for brushless permanent magnet machines,” 2011 IEEE Veh. Power Propuls. Conf. VPPC 2011, 2011. *(doi: 10.1109/VPPC.2011.6043248.)*

[17] S. T. Lee and J. Hur, “Detection technique for stator inter-turn faults in BLDC motors based on third-harmonic components of line currents,” IEEE Trans. Ind. Appl., vol. 53, no. 1, pp. 143–150, 2017. *(doi: 10.1109/TIA.2016.2614633.)*

[18] P. Sharma, and S. R. Kapoor, “Evaluation of arduino based das for condition monitoring of induction motor,” 2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC), pp. 1-4, 2017. *(doi: 10.1109/ICOMICON.2017.8279084.)*

[19] O. Andrei, S. Ungureanu, A. Miron, and A. C. Cziker, “IoT power monitoring device using Wi-Fi and Arduino,” 2021 9th International Conference on Modern Power Systems (MPS), pp. 1-6, 2021. *(doi: 10.1109/MPS52805.2021.9492651.)*

[20] M. Fran, S. Anitha, and R. R. Mohan, “IoT BASED WIND TURBINE MONITORING, FAULT DIAGNOSIS AND CONTROL USING UART,” Int. J. Adv. Res. Manag. Archit. Technol. Eng., vol. 3, no. 6, pp. 72–76, 2017.

[21] B. Srbinovski, G. Conte, A. P. Morrison, P. Leahy, and E. Popovici, “ECO: An IoT platform for wireless data collection, energy control and optimization of a miniaturized wind turbine cluster: Power analysis and battery life estimation of IoT platform,” Proc. IEEE Int. Conf. Ind. Technol., pp. 412–417, 2017. *(doi: 10.1109/ICIT.2017.7913266.)*

[22] H. A. Raja, T. Vaimann, A. Rassõlkin, A. Kallaste and A. Belahcen, "IoT Based Tools for Data Acquisition in Electrical Machines and Robotics," 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC), pp. 737-742, 2021. *(doi: 10.1109/PEMC48073.2021.9432553)*

Publication IV

Raja, H. A.; Asad, B.; Vaimann, T.; Rassõlkin, A.; Kallaste, A.; Belahcen A. (2022). Custom Simplified Machine Learning Algorithms for Fault Diagnosis in Electrical Machines. International Conference on Diagnostics in Electrical Engineering (Diagnostika), 6. - 8. 9. 2022, Pilsen, Czech Republic. IEEE.

Custom Simplified Machine Learning Algorithms for Fault Diagnosis in Electrical Machines

Hadi Ashraf Raja
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
haraja@taltech.ee

Bilal Asad
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
bilal.asad@taltech.ee

Toomas Vaimann
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
toomas.vaimann@taltech.ee

Ants Kallaste
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
ants.kallaste@taltech.ee

Anton Rassõlkin
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
anton.rassolkin@taltech.ee

Anouar Belahcen
Department of Electrical Engineering and Automation
Aalto University
Espoo, Finland
anouar.belahcen@taltech.ee

2022 International Conference on Diagnostics in Electrical Engineering (Diagnostika) | 978-1-6654-8082-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/Diagnostika55131.2022.9905174

***Abstract*—With advancements in science, machine learning and artificial intelligence integration with different fields have opened up new horizons. In this paper, some simplified custom machine learning algorithms are defined to train different faults for electrical machines. The industry has been moving towards predictive maintenance of machines rather than scheduled maintenance with the new industry 4.0 revolution. It has also paved the way for researchers to explore more in machine learning and have specific machine learning training algorithms catered to diagnose faults in electrical machines. Here, three different variations of a simplified machine learning algorithm are present for the training of faults of electrical machines. A comparison of the results is presented at the end, along with further studies carried out in this area.**

Keywords*—*machine learning, artificial intelligence, neural network

I. Introduction

With the advancement in information technology in recent years, machine learning has also grown and advanced rapidly within the domain of computing power and data analysis [1]. This growth has helped utilize software and computational hardware in a more efficient way to solve complex problems [2]. The core of this advancement lies with the methods and tools that process massive data sets allowing physical devices to learn and decide based on the previous data, resulting in the new industry revolution [3,4], commonly termed Industry 4.0. Industry 4.0 [5,6] is technically the automation of devices in the industry that enables them to communicate with each other and make a decision based on machine learning algorithms. This broad area is also known as the Internet of Things [7]. The devices can communicate with each other in a confined area and over a more extensive network and give suggestions and make decisions based on machine learning algorithms. The primary key to these decisions lies with machine learning algorithms and how well they are used according to the training data set. Machine learning algorithms are also becoming more scalable and agile with time and are a critical part of future information technology enhancement.

The machine learning algorithm can be divided mainly into four major areas: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning [8], with each further divided into different algorithms. These algorithms are becoming more common in our daily lives with the advent of large data sets for different fields being monitored and maintained online. There have also been examples of implementing machine learning algorithms in daily routine by big companies like google, facebook and others. As machine learning is a data-driven approach, it needs to keep learning with the recent data sets to get more accurate results. Hence, a minimum optimal data set and a suited algorithm are required to get the best outcome. Several algorithms, including classification analysis, data clustering, linear regression, linear discrement analysis, pattern recognition, reinforced learning technique, and others, can help build an efficient data-driven system [9,10]. Some of these algorithms are used for classification, whereas others can be used for prediction using regression or neural networks.

Although machine learning algorithms are divided into different categories, it is crucial to know how to choose the algorithm that best suits the scenario and data characteristics, as each algorithm is unique. Their results may vary with the other algorithms even of the same category [11]. As machine learning algorithms also use high processing power and storage [12,13], most processing has been moved to the cloud. Choosing the best among them is also necessary according to the need of the situation. Different research areas have integrated with information technology and used machine learning to automate or predict results in their respective areas to enhance their fields' capabilities further. This has made the application of machine learning broad, such as implementation in cyber security, IoT systems, e-healthcare, smart cities, sustainable energies, autonomous vehicles, etc. This has also resulted in a simplified version of machine learning algorithms that are best suited for the respective fields. The machine learning algorithms are still developing with the advancement in parallel to these fields and might enhance further.

This paper presents a simplified custom machine learning algorithm with two variations used to train a model for diagnosis purposes. This machine learning algorithm is still developing with further enhancements to multi-layered custom neural networks. The paper further explains the direction of research and its future prospects.

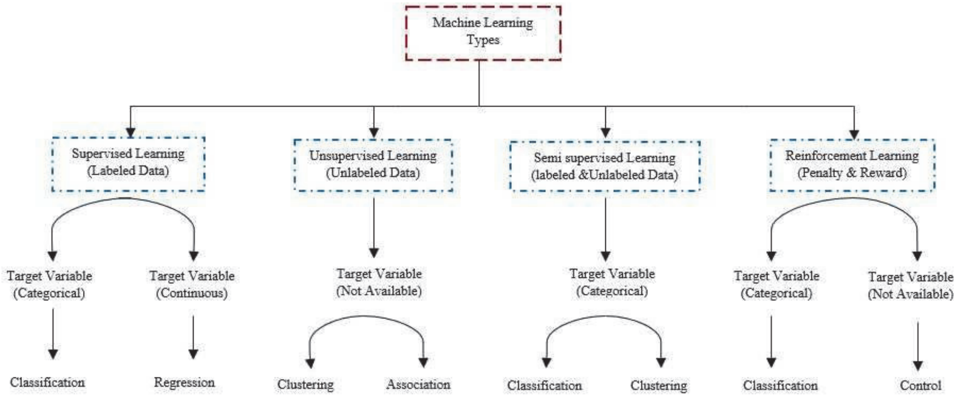


Fig. 1. Categories of Machine Learning Techniques

II. Types of Machine Learning Techniques

Machine learning algorithms are most commonly divided into three categories: supervised learning, unsupervised learning, and reinforcement learning. In addition, one other category is also taken into account, i.e., semi-supervised learning. A general overview of these four categories and a brief description are shown in Fig. 1.

A. Supervised Learning

This is the most commonly used machine learning category, as the machine learning algorithm devises a function by mapping the input to output data sample pairs [9]. The training data used here is labeled, i.e., the input and output are defined beforehand and the machine learning algorithm trains the model such that those input samples are paired with the output. The trained function is then used to predict the output for the new data set and is more of a task-driven approach [4]. The most common approach in supervised learning is "regression", which fits a function according to the data and "classification", which separates the data according to the training data set.

B. Unsupervised Learning

This is quite the opposite of supervised learning; we know the output for the input while training in supervised learning. In unsupervised learning, the data sets are unlabeled and there is no output. This approach is more suited to a data-driven approach [9] than a task-driven. In this approach, the algorithm extracts features from the data sets or trends and groups the results together based on similar patterns, trends, or features. The most common usage of unsupervised techniques is feature learning, pattern recognition, clustering, knn.

C. Semi-supervised Learning

This technique is a combination of both supervised and unsupervised learning and can be known as a hybrid technique [9]. The data used here for training is labeled and unlabeled and primarily used where labeled data is rare, especially in real-world implementation [8]. However, as unlabeled data is quite in number, a combination of both labeled and unlabeled data can generate better results in some scenarios. The most common usage of semi-supervised learning is text classification, fraud detection, machine translation and others.

D. Reinforcement Learning

Reinforcement learning is a self-learning technique in which the algorithm enables the machine to evaluate the result in a specific context and decide on the best possible outcome to improve its efficiency [14]. This approach can also be termed an environment-driven approach. This approach is based chiefly on penalty or reward, with its main purpose to minimize the risk by increasing the reward. This technique is mostly used to train AI models for automation setups like manufacturing, autonomous vehicles, robotics, and others.

There are different approaches to machine learning techniques for different problem scenarios, which mostly depend on the nature of data and where it will be implemented. Similarly, they can be used either to classify the incoming data or to predict values. Many classification techniques are present, like Linear Discriminant Analysis, Naïve Bayes, Decision Trees, Regression Techniques, Support Vector Machine. In this paper, a small custom machine learning algorithm is presented with variation to work in diagnosing faults in electrical machines. The presented technique is simple and to the point and will be further enhanced in the future to work with predictions of faults with high accuracy.

III. Methodology and Results

This research presents a simplified machine learning approach to neural networks with two variations to work on the fault diagnostics of the electrical machine. This is a more straightforward approach to get the result in a minimum number of layers and with higher accuracy in the shortest time possible. This is also to check if the results can be generated with higher accuracy using a small number of datasets. The first layer of the neural network consists of a dot product of weights and the incoming inputs, whereas the second layer is different for both variations. One variation includes a sigmoid function on the second layer, whereas the second variation has a hyperbolic tangent function present. Both equations are given in (1)(2), respectively.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2)$$

The variations are trained separately using 10,000 training data sets and then the results are validated with a different set of 500 validation data sets. The model trained is still under development to improve their results consistency as it may vary when the model is retrained. The models are trained using python with a custom neural network defined class and without the use of tensor flow or any other third-party library. The flowchart for the working and training of the neural network algorithm is shown in Fig. 2.

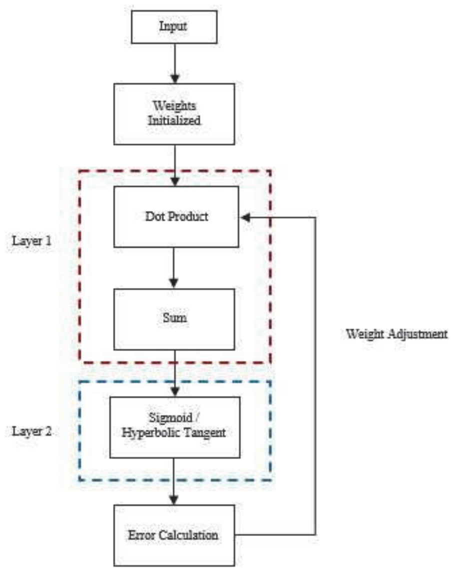


Fig. 2. Overview of Training of Machine Learning Model

The error is calculated after predicting the values after layer 2 and the weights are adjusted accordingly in layer 1 for the following input. The training undergoes 10,000 iterations for the test data set and a mean square error for the trained model is generated for both sigmoid and hyperbolic tangent variations. The results for the mean square error for both variations are shown in Fig. 3 and Fig. 4.

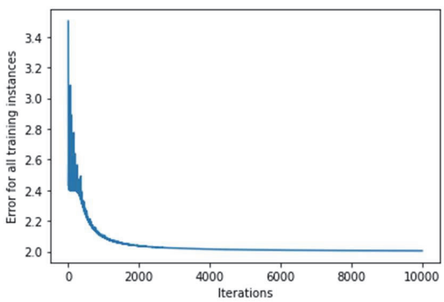


Fig. 3. Mean Square Error (MSE) of Model trained with Sigmoid Function

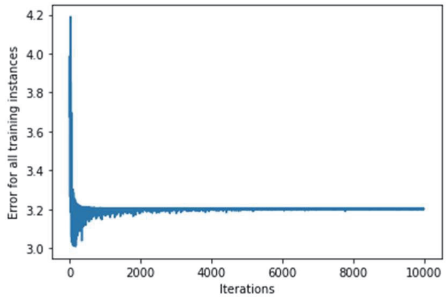


Fig. 4. Mean Square Error (MSE) of Model trained with Hyperbolic Tangent Function

From the above graphs, it can be seen that the model trained using the Sigmoid function has a low Mean Square Error (MSE) compared with the one trained using Hyperbolic Tangent Function. From the graphs of training algorithms, it is seen that the model with Sigmoid function is better off than the one with Hyperbolic Tangent Function. For validation of the results, a validation data set for 500 sets was used and the comparison results for both are summarized in Table 1.

TABLE I. COMPARISON FOR SIGMOID AND HYPERBOLIC TANGENT

	Sigmoid	**Hyperbolic Tangent**
Mean Square Error (MSE)	1.8	3.17
Accuracy	80.41%	65.48%

IV. CONCLUSION

Artificial Intelligence is taking its place in different research fields and is advancing at a rapid rate. Although there are a lot of machine learning algorithms present for classification, there is not a specific one that can be altered to one domain. This paper has presented two variations for a simplified machine learning algorithm with two layers that are a work in progress but target electrical machines' diagnostics. The proposed algorithms are simple and easy to train; they do not need a heavy system or wait for hours to get the trained model. Among the two variations of the presented approach, the one with the sigmoid function outperformed the hyperbolic tangent function in terms of performance.

The study is still a work in progress with future works, including the enhancement of neural networks to multi-layered algorithms with testing of different functions and having a consistent result in return. The future works also include further catering it towards fault diagnostics of electrical machines and adding some preprocessing, so it can detect and predict specific faults for electrical machines in the start. This will be further enhanced to make it a more generic approach.

REFERENCES

[1] I. H. Sarkar, M. H. Furhad, and R. Nowrozy, "AI-Driven Cybersecurity: An Overview, Security Intelligence Modeling and Research Directions," SN Comput Sci. 2021.

[2] K. Kannappan, L. Spector, M. Sipper, T. Helmuth, W. La Cava, J. Wisdom, and O. Bernstein, "Analyzing a decade of human-competitive ("humie") winners: What can we learn?," Genetic Programming Theory and Practice XII, pp. 149–166. Springer International Publishing, 2015.

[3] I. H. Sarker, M. M. Hoque, M. K. Uddin, and T. Alsanoosy, "Mobile data science and intelligent apps: concepts, ai-based modeling and research directions," Mob Netw Appl, pp. 1–19, 2020.

[4] I. H. Sarkar, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective," J big Data, pp. 1-29, 2020.

[5] B. Ślusarczyk, "Industry 4.0: Are we ready?," Polish Journal of Management Studies, 2018.

[6] V. Sima, I. G. Gheorghe, J. Subić, and D. Nancu, "Influences of the Industry 4.0 Revolution on the Human Capital Development and Consumer Behavior: A Systematic Review," Sustainability, vol. 12, no. 10, p. 4035, May 2020. *(doi: 10.3390/su12104035.)*

[7] C. Jeanneter, V. Callaghan, and S. B. Allouch, "The Internet-of-Things: Reflections on the Past, Present and Future from a User-centered and Smart Environment Perspective," Journal of Ambient Intelligence and Smart Environments, vol. 11, pp. 45-69, 2019.

[8] M. Mohammed, M. B. Khan, and B. E. M. Bashier, "Machine learning: algorithms and applications," CRC Press, 2016.

[9] J. han, and M. Kamber, "Data mining: concepts and techniques," Amsterdam: Elsevier, 2011.

[10] I. H. Witten, and E. Frank, "Data Mining: Practical machine learning tools and techniques," Morgan Kaufmann, Elsevier, 2005.

[11] I. H. sarkar, P. Watters, and A. S. M. Kayes, "Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. J Big Data, pp. 1-28, 2019.

[12] M. Aazam, I. Khan, A. A. Alsaffar, and E. N. Huh, "Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved," Proc. 2014 11th Int. Bhurban Conf. Appl. Sci. Technol. IBCAST, pp. 414-419 2014. *(doi: 10.1109/IBCAST.2014.6778179.)*

[13] J. H. Li, S. Zhao, and Y. R. Wang, "Using cloud computing and the internet of things to construct vehicle condition fault diagnosis platform" Applied Mechanics and Materials, 2014. *(doi: 10.4028/www.scientific.net/AMM.574.752.)*

[14] L. P. kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," Journal of Artificial Intelligence Research, 1996.

Publication V

Raja, H. A.; Vaimann, T.; Rassõlkin, A.; Kallaste, A. (2022). Condition Monitoring and Fault Detection for Electrical Machines using IOT. Proceedings of the Future Technologies Conference (FTC) 2022, Volume 2. Lecture Notes in Networks and Systems, vol 560. Springer, Cham. https://doi.org/10.1007/978-3-031-18458-1_12.

Condition Monitoring and Fault Detection for Electrical Machines using IOT

Hadi Ashraf Raja[1], Toomas Vaimann[1], Anton Rassõlkin[1] and Ants Kallaste[1]

[1] Tallinn University of Technology, Tallinn 19086, Estonia
haraja@taltech.ee

Abstract. Internet of Things (IoT) has become the need of the hour with the recent advancement in technology. The emergence of new technologies has helped to communicate between different machines and it has become easier to interact with them. This has helped with the reduction of maintenance costs and the time needed to fix the machine. Furthermore, it is necessary to monitor the impact of surrounding factors on electrical machines and detect any faults as soon as possible. With the integration of artificial intelligence and IoT, a system can be built to help detect faults as soon as it happens and help contain the impacts of such faults. In this research, a low-cost, efficient method to monitor any electrical machine and detect faults in real-time is being explored, which can help reduce maintenance charges. In addition, it can also help reduce the impact of the generated fault. For the research purpose, the setup consists of an induction motor and broken bars faults are considered for training models and detecting faults in real-time.

Keywords: Internet of Things, IoT, Electrical Machines, Condition Monitoring, Fault Detection.

1 Introduction

Recent advancements in technology have made day-to-day life much more convenient and efficient. This has also made our current devices capable of communicating with each other over the internet, thus forming a category of smart or intelligent devices. Communication of different intelligent devices through the internet is referred as the Internet of Things (IoT). The data collected from these devices can be used for many things, including diagnosing, detecting and predicting faults within the device [1][2]. This will help reduce costs and time for maintenance of the machine, hence resulting in a more cost-efficient maintenance method. Therefore, IoT plays a significant role in industrial development and can further enhance its effectiveness in the industry by utilizing predictive maintenance [3].

IoT applications in the industry have further increased from data monitoring towards predictive maintenance with development in industrial equipment. After condition monitoring [4][5] which has become quite common in the industry, predictive analytics for maintenance [6] using the collected data for machines is the next advancement. The maintenance of machines can be divided into reactive, periodic, proactive and predictive phases (see Fig. 1). Predictive maintenance is both cost and time efficient as it generates the most detailed report on the diagnostic of the fault.

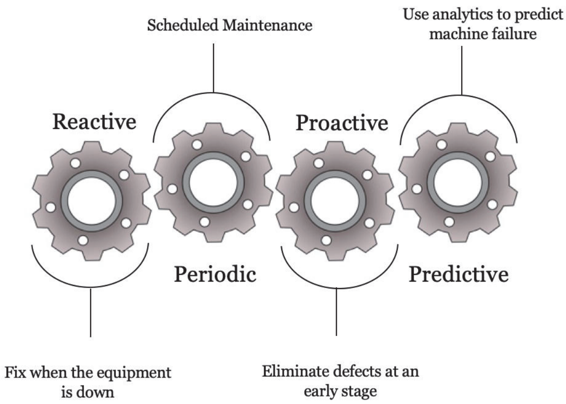


Fig. 1. Maintenance stages of electrical machines

Machine learning algorithms are mainly used to train models for predictive analysis and the accuracy of such models is highly dependent on the training data sets. Therefore, these models require a high number of data sets to get accurate or near accurate prediction and classification. Pattern recognition and deep learning algorithms, categorized under Artificial Neural Networks (ANNs) [7], are also part of machine learning algorithms. These models are mostly integrated on cloud storage or isolated servers as they need high storage and processing power [8][9].

This paper presents a cost-effective method for data acquisition from remote machines, including offshore plants and detection of faults in the recorded data set utilizing trained models using machine learning algorithms. The paper further explains the prospects and the direction of the research.

2 Related Work

With the advancement in technology and population, the need for more resources is also escalating. At the same time, there is a decrease in conventional resources, so there is an urgency to look for alternate resources and utilize the current resources with care. Condition monitoring, paired with failure prediction or fault detection, is crucial for the industry to cut costs and time. Detection of fault as it happens and monitoring the electrical machine in real-time can help identify the origin of the issue and solve it in minimal time. It can also help reduce the maintenance cost by shifting it from scheduled maintenance to predictive maintenance. Researchers have already started monitoring and controlling wind turbines to maximize their potential [10][11].

Researchers are looking into more cost-effective and stable condition monitoring methods with the advancement in technology. Much work has been done in the health sector using microcontroller boards. Most of the condition monitoring systems built are for patients with a wearable device to detect any abnormality in their conditions [12][13]. Systems have also been developed to monitor the temperature of infant incubators which can be controlled remotely using an android application [14]. Researchers are also looking into the application of these microcontroller boards in different fields of research like radiation monitoring systems [15], solar power remote monitoring systems [16], noise and air pollution monitoring systems [17], along with Carbon dioxide monitoring systems [18]. As microcontroller boards are easy to carry and can finish the job, researchers are also taking them into account when developing remote condition monitoring solutions for weather sensors [19] or wind turbines [20][21].

These cost-effective microcontroller boards are impressive when considering their scalability. These boards can also be used for condition monitoring of electrical machines, as done here for an induction machine [22][23]. With the advance in technology, more complex circuits are being introduced and used in different applications like autonomous vehicles [24] and robotics [25][26]. Hence, it is needed to monitor these machines and detect a fault in minimal time to reduce costs. The current data acquisition systems are still not final and developing with time as the safety circuit and sensors change [27]. The need is to have a stable data acquisition system that can be used with different electrical machines without much issue.

There are already some of the condition monitoring systems developed before in SCADA. However, it is not only hard to move them, but they are also expensive and require complex installations [28]. Current methods based on IoT [29][30] are also being introduced for condition monitoring. At the moment, they are only focused on data acquisition, but their sample frequency is also pretty low. Such systems should be able to gather information at a high sample frequency and should be able to help out with the detection of the fault when it occurs to help minimize the loss. Therefore, this paper presents a method to help cover the above points, which is still a work in progress for further enhancements.

3 Methodology & Results

This research explores a method to connect electrical machines to a cloud system and detect faults using cloud computing in a communicative IoT system. Cloud implementation will also be used to transfer the readings from sensors to a frontend in real-time. In comparison, trained neural network methods are used to detect faults in the cloud. One of the main aims of this research is to train such models that can help with the switch to predictive maintenance from scheduled maintenance which can help reduce costs in the case of offshore facilities and help contain the impact of faults in the case of local electrical machines. This will also help monitor the machines remotely and can be further enhanced to control the machines. The general overview of the experimental setup is presented below (see Fig 2).

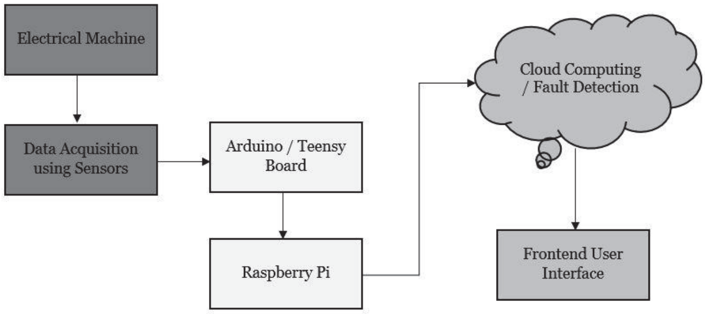


Fig. 2. General overview of the experimental setup

The research setup can be divided into three parts: data acquisition, which acts as the pre-requisite for further processing for fault detection. The second part is the condition monitoring of the electrical machines in real-time with a local backup, which ensures no loss of data while transmitting it to the cloud. Finally, faults detection can be considered the third part of the setup where models are trained and then used to detect faults in the incoming data signals. A general flowchart of the whole setup can be seen below (see Fig. 3).

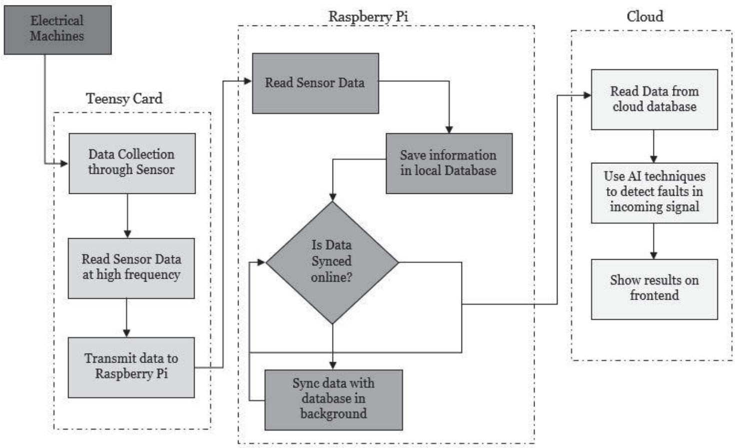


Fig. 3. Flow Chart for all three parts

3.1 Data Acquisition

The data is read through analog sensors from the electrical machine and transmitted through Arduino to Raspberry Pi. Arduino collects the data from the sensors and works as an analog to digital converter (ADC) in this case. A separate ADC can also be used instead of Arduino. For comparison purposes, the data was also recorded with Dewetron so that the recorded data signals could be compared and the setup could be corrected if there were any issues. Also, the data recorded with Dewetron was further used to train machine learning models for fault detection. The experimental setup for an induction machine is shown below (see Fig. 4).

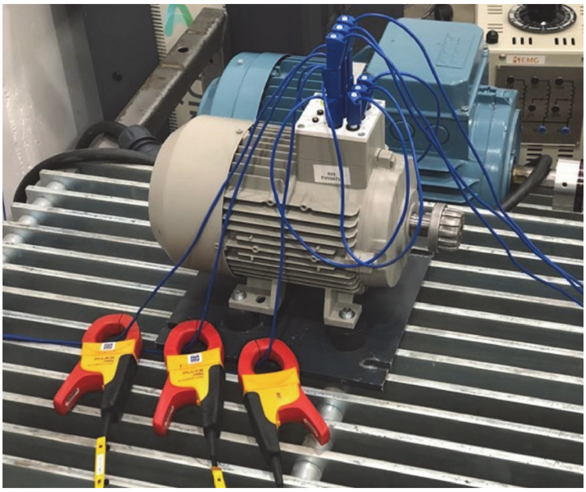


Fig. 4. Picture of Setup with induction machine

The Teensy board is used here instead of Arduino due to better speed and reliability. The converted data is transmitted to Raspberry Pi through Serial Peripheral Interface (SPI) for high data transfer rate. If the sampling rate is low, other methods such as Inter-integrated Circuits (I2C) or Universal Asynchronous Receiver-Transmitter (UART) can be used depending on the scenario. A general comparison of the data transfer rate for these three interfaces without any drop in data with breaks during the research is shown in Table 1.

Table 1. Comparison of Sample Rate for different Communication Methods

Communication Method	Sample Rate per second
UART	1800
I2C	2600
SPI	3600

These sample rates can be enhanced but drop data sets when run continuously for a more extended period of time. In this case, the tests were done for up to 7 days of continuous data transmission through the setup to get the optimal data transmission rate. Compared to Arduino, Teensy was preferred for the data acquisition part because of its advantage while collecting and processing data towards Raspberry Pi. If the data rate is low, it does not matter which card is being used for data acquisition, but it makes much difference for a high data rate. A general comparison between the transmission of data between Arduino and teensy board is given in Table 2.

Table 2. Time taken by IoT card for processing 10,000 samples

IoT Card	Processing Time for 10,000 samples
Arduino Mega	4.5 seconds
Teensy 4.0	0.75 seconds

According to the tests, different IoT cards take up different times for data transmission without any data loss for a longer period of time. Although the data rate can be increased for the specific cards, they end up losing data while continuous data transmission for a longer period of time. This data is then further processed inside Raspberry Pi and transmitted to the cloud.

3.2 Condition Monitoring

The acquired data is transmitted from Arduino to Raspberry Pi, where it is stored inside a local backup database on Pi before being transmitted to a database on the cloud for further computing. Both threads are running side by side here, Wi-Fi is being used to transmit data to the cloud, and a check is placed to make sure the data already saved up in the cloud database is not sent again. Other IoT communication methods for data transmission to the cloud are not used due to some limitations in transmission time and delay in the network due to congestion.

The database in the cloud is cloned in real-time with the one on Raspberry Pi. The values are projected on a frontend user interface in real-time. The raw values and processed values are also updated there so the data can be interpreted more easily. Each phase is represented differently, which can be further enhanced to show different sensors and devices. The data representation inside the user interface can be seen below (see Fig. 5).

Fig. 5. Example User Interface with dashboard

The user interface shows the spectrum in the frequency domain and whether a fault is present or not using knobs. It also shows a caution notification in red if an error is detected in the incoming data signal. The frequency spectrum shown here is further processed with a cut-off amplitude to omit low amplitude frequency components.

3.3 Fault Detection

The detection of faults in the incoming signal is also divided into two parts. The first part contains the training of models, whereas the second part consists of the detection of faults in the cloud. To detect faults in an electrical machine in the initial stages, it is essential to consider small frequency components. These components or harmonics can be seen in the frequency spectrum. Therefore, for the training of the model, large data samples are collected using Dewetron (see Fig. 6) for different model training. The samples were then recorded and exported for further processing.

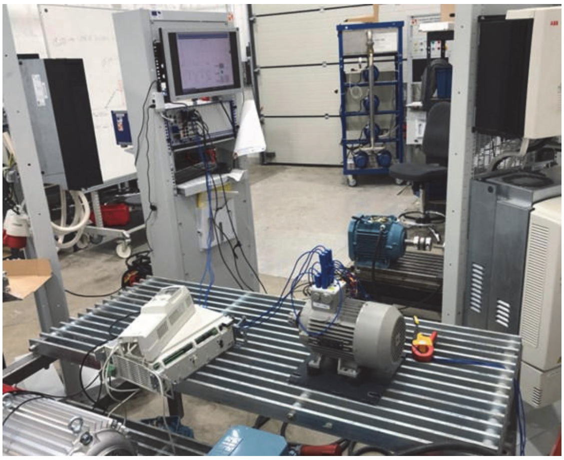


Fig. 6. Data Acquisition using Dewetron

The incoming signal is converted into the frequency domain by taking the Fast Fourier Transform (FFT) of the signal to identify the signal's frequency components. The data collected for training consists of data samples from a healthy motor to data samples consisting of faults due to broken bars. The data set consists of around 19.6 million dataset samples at a sampling frequency of 20k Hz. The collection of datasets is not done on a specific load but by varying loads so that the detection of faults is not limited to a particular load. The frequency spectrum is used to train the model, which helps differentiate between healthy and faulty motors based on different frequency components.

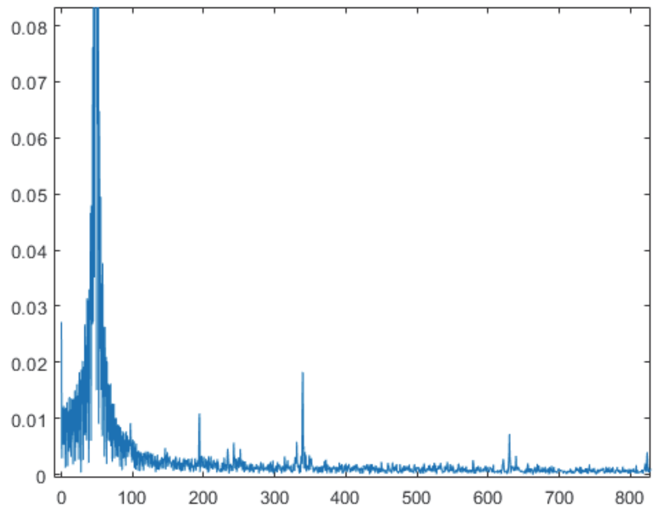


Fig. 7. FFT of one of the collected signals for training

Time-domain is not given a thought for model training as in the frequency domain; the frequency components are not dependent on the number of cycles. This helps out a great deal when faults are detected in real-time, as the incoming signals might not have a fixed number of cycles. For training of the model, the signals are classified either as healthy ‘0’ or faulty ‘1’ signals and different machine learning algorithms were used to get the optimal ones. Two trained models were selected for testing purposes. The training results for such models are shown below (see Fig. 8). The blind validation of the model was carried out on around 6 million data samples.

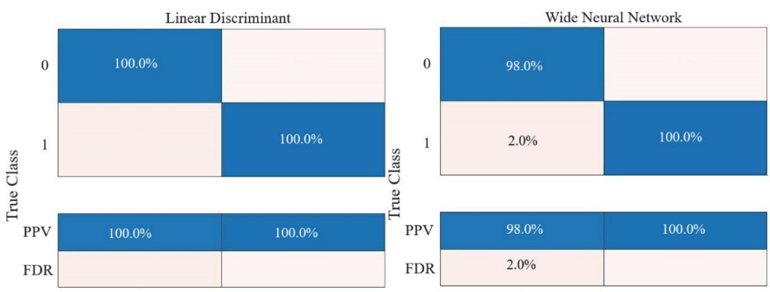


Fig. 8. Training Results for Machine learning Algorithm

The trained models were then used to detect faults in real-time on a running system with an induction motor. The results for detection for the faults are shown in Table 3. Although the accuracy of the models dropped a little from the blind validation of the trained models, that might be due to the initial samples giving false detection where the transition was done from healthy to faulty motor state. Both trained models were run with motor for a specific time ranging from minutes to hours to test the results.

Table 3. Trained Models' accuracy

Trained Machine Learning Model	Accuracy
Linear Discriminant	98%
Wide Neural Network	97.33%

Although the trained models give pretty good accuracy, there might be a need to train them using more scenarios and test cases to ensure that the accuracy remains valid. Also, to test them for a more extended period to ensure their accuracy holds out. However, the linear discriminant gives out the most accurate results, but with more data samples and complex scenarios, a neural network might take over the linear discriminant model.

4 Conclusion

IoT and cloud computing is becoming the new norm in the communication industry. This research aimed to propose a low-cost plug-and-play setup that can transmit data from remote locations or electrical machines. Further, it should detect different types of faults in the incoming data signals to reduce maintenance costs, especially for offshore locations. This will help reduce the cost and help contain the impact of fault on the machine. As scheduled maintenance costs are expensive, they are sometimes not needed for offshore locations. Also, in the case of remote electrical machines, if a fault occurs and is not addressed simultaneously can result in a bigger disaster.

The proposed setup helps reduce the maintenance cost by limiting it to maintenance when limited and helps limit the impact of the fault by detecting the fault at once and identifying the part of the electrical machine that is generating the fault. The study can be further enhanced to improve from maintenance when needed to predictive maintenance that can help limit the fault further, saving up cost and the time needed to fix and identify a more complex fault. This will also help to understand even a slight fluctuation in values that can help predict the occurrence of faults. The research is still being developed by enhancing different data generation points, including the availability of a 3 MW wind turbine, autonomous vehicle, and lab setup of a small wind generator.

Future works include the prediction of faults in real-time as they happen so the machine can be saved before it goes into a faulty state which can help reduce the maintenance costs. Also, controlling of the machine through Raspberry Pi so that as soon as it detects a fault, the machine can be stopped so that no further impact of fault occurs onto the machine. Also, remote control of the machine and self-learning models for fault detection with implementation onto Pi for edge computing is considered in future works.

The setup is independent of any specific data generation point or hardware requirement to be used as a plug-and-play device with any electrical machine. Although it does need calibration with different sensors, this is only needed once at the start. The technology can be enhanced to use with different types of sensors and in different research fields like robotics, electric transportation, vehicles, etc., and is open for further enhancements.

References

1. Asad, B., Vaimann, T., Belahcen, A., Kallaste, A.: Broken rotor bar fault diagnostic of inverter fed induction motor using FFT, Hilbert and Park's Vector approach. In: Proc. - 2018 23rd Int. Conf. Electr. Mach. ICEM 2018, pp. 2352–2358 (2018) doi: 10.1109/ICELMACH.2018.8506957.
2. Vaimann, T., Sobra, J., Belahcen, A., Rassõlkin, A., Rolak, A., Kallaste, A.: Induction machine fault detection using smartphone recorded audible noise. IET Sci. Meas. Technol. (2018) doi: 10.1049/iet-smt.2017.0104.
3. Parpla, R. C., Lacob, R.: Application of IoT concept on predictive maintenance of industrial equipment. In: MATEC Web Conf., vol. 121, pp. 1–8 (2017) doi: 10.1051/matecconf/201712102008.
4. Vaimann, T., Kallaste, A., Kilk, A.: Sensorless Detection of Induction Motor Rotor Faults Using the Clarke Vector Approach. Sci. J. Riga Tech. Univ. Power Electr. Eng., vol. 28, no. 1, pp. 16–21 (2011) doi: 10.2478/v10144-011-0007-9.
5. Sathyan, S., Belahcen, A., Kataja, J., Vaimann, T., Sobra, J.: Computation of stator vibration of an induction motor using nodal magnetic forces. In: Proc. - 2016 22nd Int. Conf. Electr. Mach. ICEM 2016, pp. 2198–2203 (2016) doi: 10.1109/ICELMACH.2016.7732827.
6. Barksdale, H., Smith, Q., Khan, M.: Condition Monitoring of Electrical Machines with Internet of Things. In: Conf. Proc. - IEEE SOUTHEASTCON, vol. 2018-April, pp. 1–4 (2018) doi: 10.1109/SECON.2018.8478989.
7. Foley, A. M., Leahy, P. G., Marvuglia, A., McKeogh, E. J.: Current methods and advances in forecasting of wind power generation. Renew. Energy, vol. 37, no. 1, pp. 1–8 (2012) doi: 10.1016/j.renene.2011.05.033.
8. Aazam, M., Khan, I., Alsaffar, A. A., Huh, E. N.: Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved. In: Proc. 2014 11th Int. Bhurban Conf. Appl. Sci. Technol. IBCAST 2014, pp. 414–419 (2014) doi: 10.1109/IBCAST.2014.6778179.
9. Li, J. H., Zhao, S., Wang, Y. R.: Using cloud computing and the internet of things to construct vehicle condition fault diagnosis platform. Applied Mechanics and Materials (2014) doi: 10.4028/www.scientific.net/AMM.574.752.
10. Mohanraj, M., Thottungal, R., Jaikumar, K.: A Can Bus Based Fault Monitoring and Diagnosis System in Wind Turbine. In: 2013 International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT), pp. 1-3 (2013) doi: 10.1109/ICEVENT.2013.6496547.
11. Wymore, M. L., Van Dam, J. E., Ceylan, H. Quai, D.: A survey of health monitoring systems for wind turbines. Renew. Sustain. Energy Rev., vol. 52, no. 1069283, pp. 976–990 (2015) doi: 10.1016/j.rser.2015.07.110.
12. Mukherjee, S., Ghosh, A., Sarkar, S. K.: Arduino based Wireless Heart-rate Monitoring system with Automatic SOS Message and/or Call facility using SIM900A GSM Module. In: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), pp. 1-5 (2019) doi: 10.1109/ViTECoN.2019.8899504.
13. Edwan, E., Abu-Musameh, M., Alsabah, A.: Blood Pressure Monitoring Using Arduino-Android Platform. In: 2020 International Conference on Assistive and Rehabilitation Technologies (iCareTech), pp. 87-91 (2020) doi: 10.1109/iCareTech49914.2020.00024.
14. Kohtimah, K., Sudrajat, M. I., Wahyu Hidayat, S.: Infant Incubator Temperature Controlling and Monitoring System by Mobile Phone Based on Arduino. In: 2019

International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 494-498 (2019) doi: 10.1109/ISRITI48646.2019.9034646.
15. Holovatyy, A. Teslyuk, V., Lobur, M., Sokolovskyy, Y., Pobereyko, S.: Development of Background Radiation Monitoring System Based on Arduino Platform. 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), pp. 121-124 (2018) doi: 10.1109/STC-CSIT.2018.8526696.
16. Haider-e-Karar, Khuwaja, A. A., Sattar, A.: Solar power remote monitoring and controlling using Arduino, LabVIEW and web browser. In: 2015 Power Generation System and Renewable Energy Technologies (PGSRET), pp. 1-4 (2015) doi: 10.1109/PGSRET.2015.7312240.
17. Janeera, D. A., Poovizhi, H., Sheik Haseena, S. S., Nivetha, S.: Smart Embedded Framework using Arduino and IoT for Real-Time Noise and Air Pollution Monitoring and Alert system. In: 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), pp. 1416-1420 (2021) doi: 10.1109/ICAIS50930.2021.9396041.
18. Lapshina, D., Kurilova, S. P., Belitsky, A. A.: Development of an Arduino-based CO2 Monitoring Device. In: 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), pp. 595-597 (2019) doi: 10.1109/EIConRus.2019.8656915.
19. Mabrouki, J., Azrour, M., Dhiba, D., Farhaoui, Y., Hajjaji, S. E.: IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts. Big Data Mining and Analytics, vol. 4, no. 1, pp. 25-32, March (2021) doi: 10.26599/BDMA.2020.9020018.
20. Xie, P., et al.: Sensing wind for environmental and energy applications. In: IET Conf. Publ., vol. 2014, no. CP639, pp. 64–69 (2014) doi: 10.1049/cp.2014.0660.
21. O'Donnell, D., Srbinovsky, B., Murphy, J., Popovici, E., Pakrashi, V.: Sensor measurement strategies for monitoring offshore wind and wave energy devices. In: J. Phys. Conf. Ser., vol. 628, no. 1, 2015, doi: 10.1088/1742-6596/628/1/012117.
22. Sharma, P., Kapoor, S. R.: Evaluation of arduino based das for condition monitoring of induction motor. In: 2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC), 2017, pp. 1-4, doi: 10.1109/ICOMICON.2017.8279084.
23. Andrei, O., Ungureanu, S., Miron, A., Cziker, A. C.: IoT power monitoring device using Wi-Fi and Arduino. In: 2021 9th International Conference on Modern Power Systems (MPS), 2021, pp. 1-6, doi: 10.1109/MPS52805.2021.9492651.
24. Ahmad, N. Bajwa, S. I. S., Ahmad, S., Kha, L., Malik, T. A., Tahir, Y.: Autonomous Unmanned Surface Vehicle a microcontroller based approach. In: 2011 International Conference on Electric Information and Control Engineering, ICEICE 2011 - Proceedings, 2011, doi: 10.1109/ICEICE.2011.5778076.
25. Kuts, V., Otto, T., Tähemaa, T., Bukhari, K., Pataraia, T.: Adaptive industrial robots using machine vision. In: ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE), 2018, doi: 10.1115/IMECE2018-86720.
26. Kuts, V., et al.: Synchronizing physical factory and its digital twin throughan iiot middleware: A case study. Proc. Est. Acad. Sci., 2019, doi: 10.3176/proc.2019.4.03.
27. Pendleton, S. D., et al.: Perception, planning, control, and coordination for autonomous vehicles. Machines, 2017, doi: 10.3390/machines5010006.
28. Debbag, Y., Yilmaz, E. N.: Internet based monitoring and control of a wind turbine via PLC. In: 2015 3rd Int. Istanbul Smart Grid Congr. Fair, ICSG 2015, 2015, doi: 10.1109/SGCF.2015.7354935.

29. Fran, M., Anitha, S., Mohan, R. R.: IoT BASED WIND TURBINE MONITORING, FAULT DIAGNOSIS AND CONTROL USING UART. Int. J. Adv. Res. Manag. Archit. Technol. Eng., vol. 3, no. 6, pp. 72–76, 2017.
30. Srbinovski, B., Conte, G., Morrison, A. P., Leahy, P., Popovici, E.: ECO: An IoT platform for wireless data collection, energy control and optimization of a miniaturized wind turbine cluster: Power analysis and battery life estimation of IoT platform. In: Proc. IEEE Int. Conf. Ind. Technol., pp. 412–417, 2017, doi: 10.1109/ICIT.2017.7913266.

Publication VI

Raja, H. A.; Vaimann, T.; Rassõlkin, A.; Kallaste, A.; Belahcen, A. (2021). IoT Based Tools for Data Acquisition in Electrical Machines and Robotics. 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC). Poland: IEEE, 737−742. DOI: 10.1109/PEMC48073.2021.9432553.

IoT Based Tools for Data Acquisition in Electrical Machines and Robotics

Hadi Ashraf Raja, Toomas Vaimann, Anton Rassõlkin, Ants Kallaste
Department of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
haraja@taltech.ee

Anouar Belahcen
Department of Electrical Engineering and Automation
Aalto University
Espoo, Finland
anouar.belahcen@aalto.fi

2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC) | 978-1-7281-5660-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/PEMC48073.2021.9432553

***Abstract*—Internet of Things (IoT) has seen rapid growth along with the recent advancement in information and communication area. With the introduction of new technologies, it has become easier to interact with machines and communicate with them. IoT cannot only be used to communicate and control these machines but it can be used further in diagnostics related to the detection or prediction of faults. As the infrastructure is advancing at a rapid speed, it has also become a need of the hour to update the way diagnostics and maintenance are carried out over them, to not only save cost but also time. This paper is a work in progress, where these opportunities will be explored in the context of IoT industrial applications.**

Keywords— Internet of Things, Data Acquisition, Condition Monitoring.

I. Introduction

Internet of Things (IoT) is the communication of different intelligent devices over the internet. The advancement in technology has not only made our life easier but has also paved new ways to be more efficient. The data from devices can be used to predict results but can also be used to diagnose the machine and predict faults [1], [2]. Hence, cutting time on the maintenance of a single machine but also making it cost-efficient to remove unnecessary maintenance checks. IoT has a lot of applications in the industrial area including predictive maintenance of industrial equipment [3].

As manufacturing has been advancing, IoT applications related to industrial development and monitoring have been increasing rapidly too. The inclusion of data collection from machines to monitor them [4], [5] and run predictive analytics for maintenance [6] is becoming a norm in the industrial field. Due to the advancement in this field, maintenance has been divided into several stages as described in Fig. 1. Usually, predictive maintenance is used, as it is not only cost-efficient but also saves time, as it gives a detailed report on the diagnostics of the fault.

Machine learning algorithms are used to train models for predictive analysis. The accuracy of any such model highly depends on the data set used for training. These models can be used for prediction of faults based on previous patterns, but they require a high number of data sets for accurate or near accurate predictions. Deep learning algorithms and pattern recognition algorithms also come in the subdomain of machine learning and are commonly known as Artificial Neural Networks (ANN), which is inspired by the working of a human brain [7]. As these trained models need high processing power and storage, they are mostly integrated with cloud computing [8], [9].

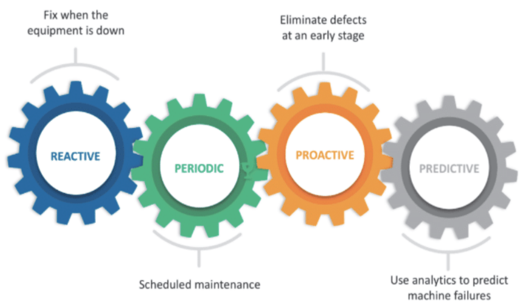


Fig. 1. Maintenance stages of electrical machines.

This paper presents an overview of the methodology used for collecting data at high frequency and syncing it with the cloud without data loss. The paper further explains the prospects of the research and in what direction it is heading.

II. Related Work

The main architecture that is considered in the power of IoT related to a smart grid is mainly applied to the overall transmission of power, perception of grid and fault diagnostics architecture. The hierarchical structure is similar to that of a traditional IoT system including the network, transmission, perception and application layer [10]. The general overview of the architecture is shown in Fig. 2.

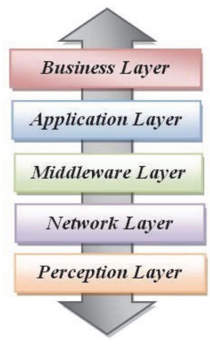


Fig. 2. Architecture for power Internet of Things

This research has been supported by the Estonian Research Council under grant PSG453 "Digital twin for propulsion drive of autonomous electric vehicle". The research leading to these results has received funding from the [EEA]/ [Norway] Grants 2014-2021, "Industrial Internet methods for electrical energy conversion systems monitoring and diagnostics".

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE
978-1-7281-5660-6/21/$31.00 ©2021 IEEE

Nowadays, as the requirements for human resources are increasing with the growth in population, the need for energy resources is also escalating, whereas the conventional resources are decreasing with time. Hence, there is an urgency to look for alternative energy sources and researches is being done in different fields to find feasible energy sources. Among these researches for alternative energy, wind energy is rising as one of the preferred possibilities. Wind energy is being widely used because of its advantages, like no greenhouse effect, nonpolluting, availability, no gas emission, etc. [11]. Wind turbines are being used to convert wind energy into a useful form of electrical energy but to get the maximum out of this system, this process should be monitored and controlled [12], [13].

Most of the time, wind turbines are in complex topography with limited access and a difficult way of communication. Therefore, much work has been done in controlling and monitoring the wind turbines remotely and sending the maintenance staff only when it is necessary. Monitoring the condition of the wind turbines at regular intervals is applied universally and is necessary to reduce the downtime of the turbines and get the maximum out of them [14]. Examples of work done, related to wind energy, utilizing wireless technologies are given in [15], [16]. Some researchers have also tried to deploy wind turbines on the roofs of buildings, but the turbines deployed cannot be bigger in size and their productivity is lacking with respect to cost [17], [18].

With time, more and more offshore wind turbines are being erected, hence, giving in space for an urgent need for predictive, proactive and commercial maintenance. Major problems faced due to these wind turbines are in maintenance and the downtime due to faults in bearings, gearboxes or other electromechanical components. These can be overcome by installing sensors [19] and continuous monitoring of the wind turbine. The fault rate of a wind turbine is high, which also results in a cost for maintenance of these wind turbines as offshore maintenance teams must be deployed after a specific number of days to check for faults, which also emphasizes the need for reliable fault detection and monitoring system for wind turbines [20].

With the current advancement in technology more complex systems are also being used in robotics [21], [22]K and autonomous vehicles[23]. The current systems used for data acquisition in robotics are still developing and changing over time due to the increase in the number of sensors and safety circuits[24]. The need to process these readings in real-time has also increased the need for higher sampling frequency. More research is being done in this regard taking into account different environmental effects too[25]–[28].

Some of these monitoring systems have already been developed in SCADA but these systems are very expensive and require more space and complex installation [29]. There are some methods based on IoT [30], [31] but their sample rate for collection of data is not only low but they are also solely focused on the purpose of collection rather than premature detection of faults and monitoring of turbines. Wind energy monitoring based on IoT should have strong reliability, accessibility, profitability and flexibility. It should include the main aspects of real-time monitoring of wind turbines, real-time collection of data from sensors, premature fault detection based on patterns generated from data collection, prediction of wind generation and other predictive analysis. Therefore, in this paper, we are presenting research that is a work in progress covering all the basics mentioned above.

III. Methodology

In this research topic, a method will be implemented to connect electrical machines over the cloud, thus forming a communicative IoT system. Apart from the implementation of the IoT system, cloud computing will also be utilized to run different machine learning and pattern recognition algorithms to predict and detect faults in electrical machines. One of the aims of this research area will be to switch electrical machines over to predictive maintenance from scheduled maintenance to make maintenance more time and cost-efficient for offshore electrical machines. One of the other tasks will be monitoring the incoming data and removing any noise if present. An industrial example for such type of IoT system is represented in Fig. 3.

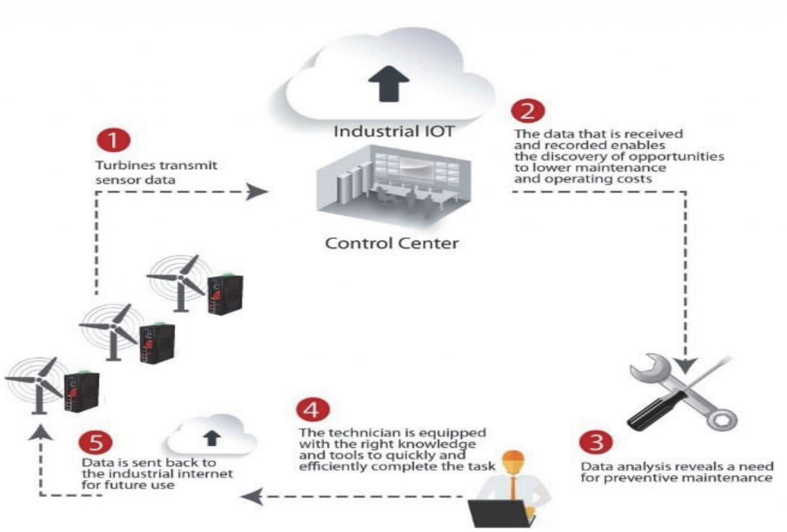


Fig. 3. Wind Turbine IoT Industrial implementation

Fig. 4. Experimental setup for data gathering from wind turbine

Our research focuses on not only collecting data in real-time for monitoring of the wind turbines but also a prediction of the faults beforehand so that they do not propagate into a major fault. This will also help to reduce the time taken by the maintenance team on offshore turbines and help them identify the fault beforehand. The research can be divided into two parts with first being considered as a prerequisite for machine learning and predictive analysis, and the second being predictive analysis and fault diagnostics. For our first part, we have set up an in-house test bench for the collection of data sets using wind turbines and wind tunnel. The running example of this setup is shown in Fig. 4. Also, we are developing our own setup for collecting data from sensors, which will be set up inside the wind turbines to detect different parameters like vibration, wind speed, temperature, voltage, etc. We are collecting data at high frequency to collect more information for the dataset, hence, we will be able to identify even a slight fluctuation and can predict the faults beforehand, which will not only decrease the overhead caused by major fault occurring later but also will reduce the downtime for the wind turbine.

The setup used for collecting data comprises of two-part with our sensors being attached to a Teensy card that can also act as an analog to digital converter and forwards the data in digital form to Raspberry Pi, which will not only act as a hub for different sensors but will also keep a local database as a backup to the cloud. The setup we will use for the collection of data from wind turbines is not only cost-effective as compared with other equipment but also does not take much space. Fig. 5 shows the setup of Raspberry Pi with an IoT card for the collection of data from a wind turbine. This setup can be used with any sensor for the collection of data and can be placed with any electrical machine for monitoring.

Raspberry Pi with the card and sensor is used for gathering data from the experimental setup and the values gathered at

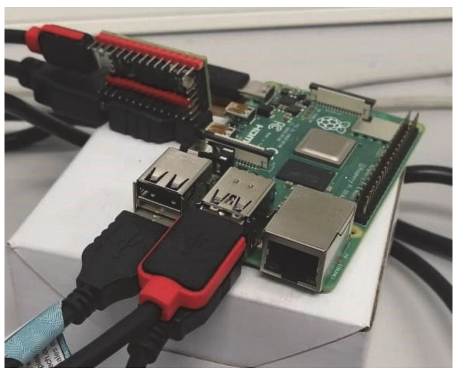


Fig. 5. Experimental setup of the card for gathering data

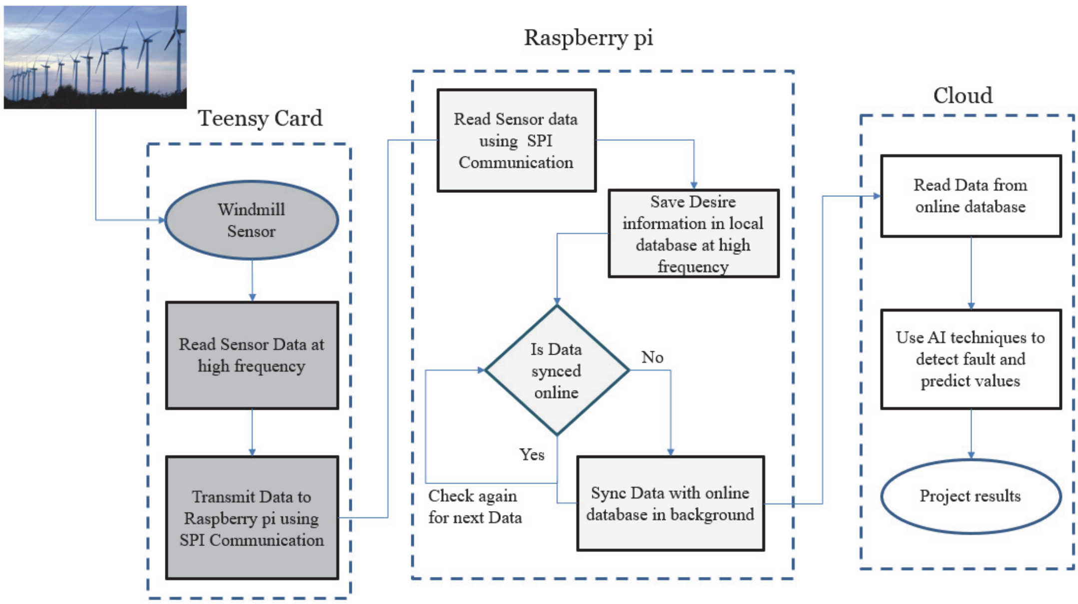


Fig. 6. Flow chart for data acquisition from a windmill to cloud

high frequency were accurate in accordance with the values gather through the Dewetron data acquisition device. Furthermore, the values recorded were also pushed into the database at high frequency without any loss of data. The sampling frequency attained until now by our setup is approximately ~3400 samples per second from recording to inserting it in the database in real-time, shown in Table I. The flow chart of the whole setup is shown in Fig. 6. The aim of our setup is also to explore different ways of communication between machines and to see, which one of the current methods is the most feasible and fastest way to transfer data without any loss. This is the first part of the research that we are currently working on, whereas for the second part we will be doing predictive analysis along with different machine learning and pattern recognition techniques to detect faults at a premature level. This step will be further worked on once we have enough data to train models and do predictions accurately.

TABLE I. SAMPLE RATE FOR DIFFERENT TYPES OF COMMUNICATION METHODS

Communication Method	Sample Rate / sec
UART	1800
SPI Interface	3600

TABLE II. PROCESSING TIME FOR DIFFERENT IOT CARDS PER 10,000 SAMPLES

Card	Processing time for 10,000 samples in seconds
Arduino Mega	4.5 sec
Teensy 4.0	0.75 seconds

Table II shows the time taken in the case for the processing of 10,000 samples for different cards. The processing time includes data acquisition from the sensors, processing of incoming data inside the card and then transmission of data to Raspberry Pi using either UART or SPI Interface communication.

IV. CONCLUSION

IoT devices and cloud computing are becoming more and more popular now a day as it gives the monitoring team more control over the electrical machine. This research aims to not only monitor offshore electrical machines but also to cut off extra expenses and save time from unnecessary scheduled maintenances. It will also help later to predict faults using incoming signals and diagnosis to pinpoint the necessary reasons for the generation of fault, which will further save time on maintenance and removal of that fault. We will be researching and implementing such an IoT system for offshore wind turbines. We already have different points for data generation, including the availability to monitor a real-life 3 MW wind turbine. Also, we have access to the TalTech satellite ground control tracking antenna in addition to the in-lab setup.

One of our future research aims is to extract data at 10000 samples per second in real-time on the cloud so that we can have enough data to not only train our models efficiently but also to determine even a slight fluctuation in values. The sampling frequency of 10,000 is taken as reference, for now, considering multiple sensors acquiring data at the same time, with further testing this value will be optimized. This will also help us to predict more accurate values in terms of power generation and maintenance due dates. This will further help our research aim to reduce the cost and time taken for offshore maintenance and will help detect the faults remotely.

This research can be further enhanced for monitoring the wind turbines remotely and predict whether a minor fault can give way to a major one, later on, this will also help to send out maintenance teams when needed and not periodically even where there is no need.

This research can also be applied in the field of robotics as there are several sensors implemented in its circuit and there is a high frequency for data acquisition in real-time. This setup can be used to acquire data and process it in real-time to predict movements and to keep remote maintenance of sensors and circuits. This can also be further enhanced to control the working of a robot remotely.

The technology developed is independent of any specific data generation point or hardware so it can be further adapted in other fields of research in electric transportation, vehicles, medical equipment, robotics, etc., and is open for future research.

REFERENCES

[1] B. Asad, T. Vaimann, A. Belahcen, and A. Kallaste, "Broken rotor bar fault diagnostic of inverter fed induction motor using FFT, Hilbert and Park's Vector approach," *Proc. - 2018 23rd Int. Conf. Electr. Mach. ICEM 2018*, pp. 2352–2358, 2018, doi: 10.1109/ICELMACH.2018.8506957.

[2] T. Vaimann, J. Sobra, A. Belahcen, A. Rassõlkin, M. Rolak, and A. Kallaste, "Induction machine fault detection using smartphone recorded audible noise," *IET Sci. Meas. Technol.*, 2018, doi: 10.1049/iet-smt.2017.0104.

[3] R. C. Parpala and R. Iacob, "Application of IoT concept on predictive maintenance of industrial equipment," *MATEC Web Conf.*, vol. 121, pp. 1–8, 2017, doi: 10.1051/matecconf/201712102008.

[4] T. Vaimann, A. Kallaste, and A. Kilk, "Sensorless Detection of Induction Motor Rotor Faults Using the Clarke Vector Approach," *Sci. J. Riga Tech. Univ. Power Electr. Eng.*, vol. 28, no. 1, pp. 16–21, 2011, doi: 10.2478/v10144-011-0007-9.

[5] S. Sathyan, A. Belahcen, J. Kataja, T. Vaimann, and J. Sobra, "Computation of stator vibration of an induction motor using nodal magnetic forces," *Proc. - 2016 22nd Int. Conf. Electr. Mach. ICEM 2016*, pp. 2198–2203, 2016, doi: 10.1109/ICELMACH.2016.7732827.

[6] H. Barksdale, Q. Smith, and M. Khan, "Condition Monitoring of Electrical Machines with Internet of Things," *Conf. Proc. - IEEE SOUTHEASTCON*, vol. 2018-April, pp. 1–4, 2018, doi: 10.1109/SECON.2018.8478989.

[7] A. M. Foley, P. G. Leahy, A. Marvuglia, and E. J. McKeogh, "Current methods and advances in forecasting of wind power generation," *Renew. Energy*, vol. 37, no. 1, pp. 1–8, 2012, doi: 10.1016/j.renene.2011.05.033.

[8] M. Aazam, I. Khan, A. A. Alsaffar, and E. N. Huh, "Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved," *Proc. 2014 11th Int. Bhurban Conf. Appl. Sci. Technol. IBCAST 2014*, pp. 414–419, 2014, doi: 10.1109/IBCAST.2014.6778179.

[9] J. H. Li, S. Zhao, and Y. R. Wang, "Using cloud computing and the internet of things to construct vehicle condition fault diagnosis platform," in *Applied Mechanics and Materials*, 2014, doi: 10.4028/www.scientific.net/AMM.574.752.

[10] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern classification," *New York John Wiley, Sect.*, 2001.

[11] M. Mohanraj., R. Thottungal, and K. Jaikumar, "A CAN bus based system for monitoring and fault diagnosis in wind turbine," *2013 Int. Conf. Emerg. Trends VLSI, Embed. Syst. Nano Electron. Telecommun. Syst. ICEVENT 2013*, pp. 4–6, 2013, doi: 10.1109/ICEVENT.2013.6496547.

[12] M. A. Sayyed, N. Ansari, and S. A. Mobeen, "a Can Bus Based Fault Monitoring and Diagnosis System in Wind Turbine," no. 3, pp. 56–58, 2014.

[13] M. L. Wymore, J. E. Van Dam, H. Ceylan, and D. Qiao, "A survey of health monitoring systems for wind turbines," *Renew. Sustain. Energy Rev.*, vol. 52, no. 1069283, pp. 976–990, 2015, doi: 10.1016/j.rser.2015.07.110.

[14] F. P. García Márquez, A. M. Tobias, J. M. Pinar Pérez, and M. Papaelias, "Condition monitoring of wind turbines: Techniques and methods," *Renew. Energy*, vol. 46, pp. 169–178, 2012, doi: 10.1016/j.renene.2012.03.003.

[15] P. Xie *et al.*, "Sensing wind for environmental and energy applications," *IET Conf. Publ.*, vol. 2014, no. CP639, pp. 64–69, 2014, doi: 10.1049/cp.2014.0660.

[16] D. O'Donnell, B. Srbinovsky, J. Murphy, E. Popovici, and V. Pakrashi, "Sensor measurement strategies for monitoring offshore wind and wave energy devices," *J. Phys. Conf. Ser.*, vol. 628, no. 1, 2015, doi: 10.1088/1742-6596/628/1/012117.

[17] M. Luleva, "The Green Optimistic Micro-Wind Turbine Array Power's Intel's Headquarters," 2015. [Online]. Available: https://www.greenoptimistic.com/intel-micro-wind-turbine/. [Accessed: 16-Feb-2020].

[18] T. Levesque, "Inhabitat Micro Wind Turbine: Small Size, Big Impact," 2007. [Online]. Available: https://inhabitat.com/micro-wind-turbines-small-size-big-impact/. [Accessed: 16-Feb-2020].

[19] S. J. Watson, B. J. Xiang, W. Yang, P. J. Tavner, and C. J. Crabtree, "Condition monitoring of the power output of wind turbine generators using wavelets," *IEEE Trans. Energy Convers.*, vol. 25, no. 3, pp. 715–721, 2010, doi: 10.1109/TEC.2010.2040083.

[20] Z. Fu, M. Zhao, Y. Luo, and Y. Yuan, "Self-healing strategy for wind turbine condition monitoring system based on wireless sensor networks," *ICCSE 2016 - 11th Int. Conf. Comput. Sci. Educ.*, no. Iccse, pp. 544–549, 2016, doi: 10.1109/ICCSE.2016.7581639.

[21] V. Kuts, T. Otto, T. Tähemaa, K. Bukhari, and T. Pataraia, "Adaptive industrial robots using machine vision," in *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, 2018, doi: 10.1115/IMECE2018-86720.

[22] V. Kuts *et al.*, "Synchronizing physical factory and its digital twin throughan iiot middleware: A case study," *Proc. Est. Acad. Sci.*, 2019, doi: 10.3176/proc.2019.4.03.

[23] N. Ahmad, S. I. S. Bajwa, S. Ahmad, S. L. Khan, T. A. Malik, and Y. Tahir, "Autonomous Unmanned Surface Vehicle a microcontroller based approach," in *2011 International Conference on Electric Information and Control Engineering, ICEICE 2011 - Proceedings*, 2011, doi: 10.1109/ICEICE.2011.5778076.

[24] S. D. Pendleton *et al.*, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, 2017, doi: 10.3390/machines5010006.

[25] A. S. Jaiswal and V. Baporikar, "Embedded Wireless Data Acquisition system for unmanned vehicle in underwater environment," in *2015 IEEE Underwater Technology, UT 2015*, 2015, doi: 10.1109/UT.2015.7108223.

[26] B. Lawal, S. S. A. Ali, and A. Bin Awang, "Massive MIMO systems for underwater acoustic communication," in *USYS 2016 - 2016 IEEE 6th International Conference on Underwater System Technology: Theory and Applications*, 2017, doi: 10.1109/USYS.2016.7893912.

[27] V. Mhatre, S. Chavan, A. Samuel, A. Patil, A. Chittimilla, and N. Kumar, "Embedded video processing and data acquisition for unmanned aerial vehicle," in *Proceedings - 2015 International Conference on Computers, Communications and Systems, ICCCS 2015*, 2016, doi: 10.1109/CCOMS.2015.7562889.

[28] V. Djapic, G. Galdorisi, A. Jones, and M. Ouimet, "Heterogeneous Autonomous Mobile Maritime Expeditionary Robots," in *OCEANS 2016 MTS/IEEE Monterey, OCE 2016*, 2016, doi: 10.1109/OCEANS.2016.7761092.

[29] Y. Debbag and E. N. Yilmaz, "Internet based monitoring and control of a wind turbine via PLC," *2015 3rd Int. Istanbul Smart Grid Congr. Fair, ICSG 2015*, 2015, doi: 10.1109/SGCF.2015.7354935.

[30] M. Fran, S. Anitha, and R. R. Mohan, "IoT BASED WIND TURBINE MONITORING , FAULT DIAGNOSIS AND CONTROL USING UART," *Int. J. Adv. Res. Manag. Archit. Technol. Eng.*, vol. 3, no. 6, pp. 72–76, 2017.

[31] B. Srbinovski, G. Conte, A. P. Morrison, P. Leahy, and E. Popovici, "ECO: An IoT platform for wireless data collection, energy control and optimization of a miniaturized wind turbine cluster: Power analysis and battery life estimation of IoT platform," *Proc. IEEE Int. Conf. Ind. Technol.*, pp. 412–417, 2017, doi: 10.1109/ICIT.2017.7913266.

Publication VII

Raja, H. A.; Kudelina, K.; Asad, B.; Vaimann, T. (2022). Fault Detection and Predictive Maintenance of Electrical Machines. In: New Trends in Electric Machines - Technology and Applications IntechOpen. DOI: 10.5772/intechopen.107167.

Fault Detection and Predictive Maintenance for Electrical Machines

Hadi, Ashraf Raja, Tallinn University of Technology, Tallinn, Estonia, haraja@taltech.ee
Karolina, Kudelina, Tallinn University of Technology, Tallinn, Estonia, karolina.kudelina@taltech.ee
Bilal, Asad, Tallinn University of Technology, Tallinn, Estonia, bilal.asad@taltech.ee
Toomas, Vaimann, Tallinn University of Technology, Tallinn, Estonia, toomas.vaimann@taltech.ee

Abstract

Nowadays, most domestic and industrial fields are moving towards Industry 4.0 standards and integration with information technology. To decrease shutdown costs and minimize downtime, manufacturers switch their production to predictive maintenance. Algorithms based on machine learning can be used to make predictions and detect timely potential faults in modern energy systems. For this, trained models with the usage of data analysis, cloud, and edge computing are implemented. The main challenge is the amount and quality of the data used for model training. This chapter discusses a specific version of a condition monitoring system, including maintenance approaches and machine learning algorithms and their general application issues.

Keywords: electrical machines, fault diagnostics, predictive maintenance, artificial intelligence, condition monitoring, neural networks.

1. Introduction

The recent advancement in information technology, especially the integration of technology with different fields of research, has made day-to-day life convenient and opened up new research areas. One of these fields is the Internet of Things (IoT), which enables physical devices to communicate through the internet. The advent of these smart or intelligent devices and their implementation in industrial applications resulted in the industrial revolution, commonly known as industrial standard 4.0. These devices are not only able to communicate with each other but also able to make decisions based on defined logic or controlled remotely also referred as Cyber-physical systems. This has further paved the way for condition monitoring of electrical devices, where these devices act as data acquisition points. The collected data can then be used to monitor specific electrical machines. Further data analysis can be done on the collected data to include fault diagnostics on these devices, including the prediction of faults [1,2].

Industrial standard 4.0 have given way to the implementation of condition monitoring [3.4] at a mass scale in the industry, leading toward predictive maintenance [5, 6] of electrical machines in the near future. Many companies are working on different predictive maintenance algorithms to reduce their scheduled maintenance costs. This research will further improve the effectiveness of electrical machines in the industry [7] and help reduce unforeseen errors and faults. Most companies are also researching finding the lifespan of the equipment based on previous patterns and external environmental variables to get the best results out of their setup. Researchers have already implemented different condition monitoring setups to maximize the potential of different electrical machines, including offshore wind turbines [8, 9], but most of this equipment is expensive and heavy.

At the moment, the industry is trying to move towards predictive and proactive maintenance to help reduce costs due to unexpected errors and faults that could have been handled before they become a more significant issue. The maintenance of electrical machines is usually divided into four phases: reactive, periodic, proactive, and periodic, as shown in Figure 1. Among the four phases, most of the industry is still on scheduled maintenance but is trying to move towards predictive maintenance as it is not only cost efficient but also generates a more detailed report on fault diagnostics.

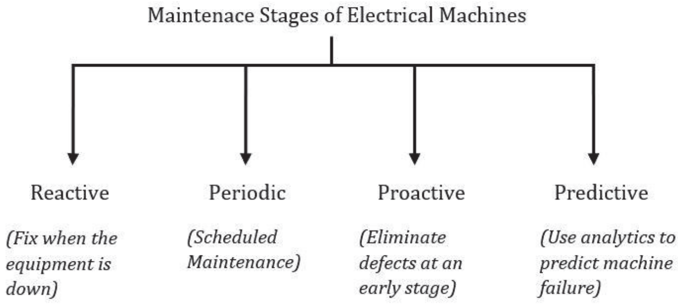


Figure 1. Maintenance types of electrical machines**.**

With the move towards predictive maintenance, researchers are also looking for ways to utilize newer technology to get better results. The research is not only going on in this area but also in other areas like wearable devices for condition monitoring of patients to check on any abnormality [10, 11], solar-powered condition monitoring systems [12], air and noise pollution monitoring systems [13], and much more. This is because of the advancement in the technology of microcontroller boards that have given researchers more options to explore. More researchers are including these boards in their research because of their scalability. There have already been researches going on like the development of a condition monitoring system for wind turbines [14], weather sensors [15], electrical machines [16-18], autonomous vehicles [19], and robotics [20, 21]. Most of these condition monitoring systems are still in development and might need much more improvement before they can become stable and be used on a large scale. One of the most common issues is the sample rate at which data is gathered using these devices and its transmission without any data loss.

Most of the systems already in place use SCADA / PLC which are not only complex and expensive but also harder to transport [22]. One of the other issues with these systems is that although they are data acquisition points, there is no data analysis of the collected data. Hence, it is just lying there and not being utilized anywhere for fault diagnostics or being used to deduce any results. For the analysis of the collected data, cloud computation is used along with edge computing, which helps analyze the data and deduce results from it. For the analysis part, machine learning algorithms are mainly used to train models based on collected data from these machines. These trained models are implemented on the cloud to get near accurate classification and prediction related to incoming data from the electrical machines. These models mainly were implemented on cloud storage or isolated servers as they need high processing power and storage space. However, now things are moving towards edge computation from the cloud. This will result in these models being implemented at the edge node where the data is being collected rather than on the cloud, which will help identify errors on the edge and further reduce the time needed to make a decision. This will also result in reduced bandwidth needed to transfer the data over the network.

This chapter discusses a concise overview of a condition monitoring system using microcontroller cards, following a small data pre-processing and analysis. Further, some light is shown on the machine learning algorithms and the training of data sets for different faults, and a short detail related to predictive maintenance is given, how it can help, and at what stage it is currently at, followed by a short conclusion.

2. Condition Monitoring System

This section will discuss a particular approach on condition monitoring system based on micro-controller boards and cloud resources. The condition monitoring system technically consists of three parts: the data acquisition system, the edge node and the cloud. Usually, the researchers do not consider an edge node system. However, it is always better to have a local backup, computation power, and space to run some analysis if needed. The data acquisition part will consist of the microcontroller board, with the edge node being the one that helps in case of any data loss over the network.

The data acquisition part will gather data from the electrical machine using sensors. The incoming data is calibrated before transferring it through the micro-controller board to the edge node. In most cases, as the industry uses analog sensors, this part also acts as an analog to digital converter (ADC). The acquired data is then transmitted towards the second part i.e., the edge node. The edge node acts as a local backup where the incoming data is stored in a mysql database. The database is synced in real-time with the database present in the cloud. Some pre-processing can also be done on edge, including digital filtering. The third part of the system, which is in the cloud, runs the frontend UI for the end user. It also runs diagnostics in the background on the latest synced data to look for faults. As the time difference between data acquisition from the electrical machine to showing the diagnostic results on the frontend is not much, this system can also be referred to as a real-time condition monitoring system. Figure 2 shows a rough flow chart of the implementation of a condition monitoring part.

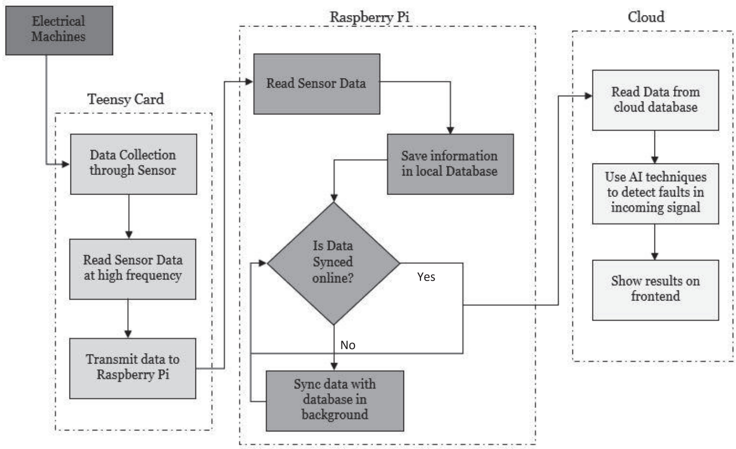


Figure 2. Flow chart of a condition monitoring system.

The data is collected from the electrical machines using a microcontroller, an Arduino or a teensy. One of the microcontroller cards (i.e., Arduino) is shown in Figure 3. The collected data is read through one of the analog or digital pins on the micro-controller card, depending on the sensor used for data collection. If the sensor is IoT compatible, the data can be read over on the digital pin. In contrast, the general analog sensors used in the industry need to be calibrated and their output adjusted before they can be passed onto the micro-controller board. As the pins on the microcontroller boards do not allow a negative voltage or more than a specific voltage, before providing the data to the pins of the board, it is necessary to make sure that the sensor output is calibrated correctly. If by any chance, there is a negative voltage or higher voltage than the one pin can handle, there is a high chance that the board will short circuit. So, it is essential to make sure that this is handled correctly; otherwise, might end up in a short circuit of the board and with the data collected being junk without any real meaning.

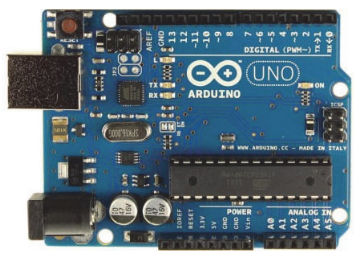


Figure 3. Arduino board.

Once the data is received on the board, it is then forwarded to the edge node, which is made up of Raspberry Pi. The data read here through the analog pin is at high speed. To ensure it is transmitted at the same speed without any loss of data Serial Peripheral Interface (SPI) connection is used between the microcontroller board and Raspberry Pi. Also, to be sure, the voltages for both the microcontroller board and Pi are different as some microcontroller boards give an output of 5 Volts at high. In contrast, Pi works with a voltage of 3.3 Volts when high, so it is also needed to ensure the transmitted values do not go over it. If a high sample rate is not needed, then UART communication should be preferred. A short description of different communication methods and their sample rate for a longer period of time is shown in Table 1.

Table 1. Comparison of Sample Rate for different Communication Methods

Communication Method	Sample Rate per second
UART	1800
I2C	2600
SPI	3600

The above sample rate per second is just a comparison between the speed for different communication methods for a specific micro-controller board. In this case, the micro-controller board considered is Arduino Mega. The communication devices and other specifications, including the buffer capacity of the logger device, are the same in all three cases, i.e. Arduino Mega and Raspberry Pi. The results shown in Table 1 are approximately maximum sampling rates of an Arduino Mega that can be achieved when run for over a couple of days with the specific communication method without any data loss during transmission from Arduino Mega to Raspberry Pi. These specific results are hardware dependant and changing the micro-controller board will change the speed range, e.g. teensy has a far better range. An experimental setup with an induction motor and analog current sensors for data acquisition is shown in Figure 4.

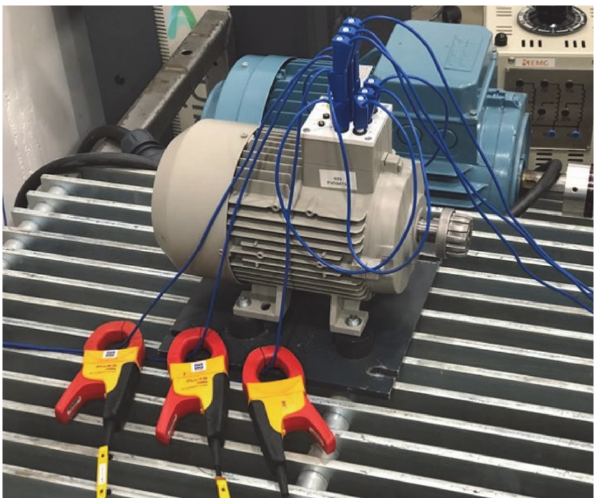


Figure 4. Experimental setup of induction motor with analog sensors.

The communication method for data transmission between the microcontroller board and Pi can be decided based on the sample rate needed for the transmission. These sample rates are based on continuous data transmission from a couple of hours to days without any data loss between the transmission. Similarly, the choice of microcontroller board might also impact the sample rate for transmission, as the newer board having better computation power gives better results. Once the data is transmitted to Pi, it is saved up in a local database and synced online simultaneously to ensure that every bit of data is synced online with the cloud without any loss. Pi also acts as a node that is capable of running analysis (like digital filtering) if needed going forward. The transmitted data is then analyzed on the cloud and based on different trained models; results are deduced whether any fault is present or not. As it is harder to understand incoming data in numerical form, the deduced results are then shown at a front end hosted on the cloud. The graphical user interface (GUI) is user-friendly and helps the end user understand the result without much information related to the system. An example of such a GUI is shown in Figure 5.

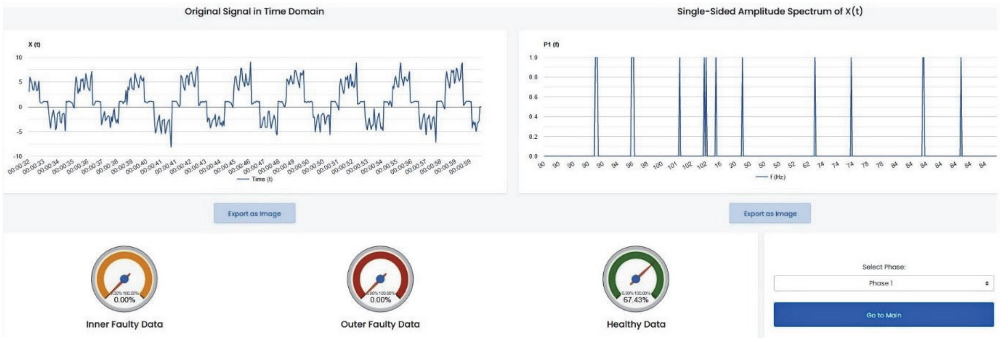


Figure 5. Example of GUI.

The GUI shown in Figure 5 runs on the cloud with scalable resources. It mainly consists of two parts, the GUI and the diagnostic analysis running in the background. The GUI is built using PHP, whereas the diagnostic analysis primarily uses Python as the primary language, with the results saved in a Mysql database. The saved results are then projected on the GUI as soon as they are updated in the database. The cloud resources used here are scalable. With low data processing, i.e., only one or a couple of diagnostic analysis running in the background, resources with 4 vCPU cores, 16 GB RAM and 128 GB Disk are good enough. This can be further scaled up depending on the number of diagnostic analysis and edge devices connected with the cloud, i.e., increased incoming data flow.

Further analysis results can also be shown on the GUI including the chance of a fault occurring in each phase and the option to control the electrical machine remotely if power to the machine is routed through the micro-controller board. Hence, there are multiple ways this system can be extended further. This can help the end user understand the situation of the electrical machine in more detail. This can also help identify which phase of the electrical machine or which part of the machine is generating issues which can further reduce the time taken to identify the root cause of the fault. This helps maintenance teams in reducing the time needed to fix it and decide whether the fault needs to be fixed urgently or can be done later.

3. Data Pre-processing and Analysis

The incoming data needs to be pre-processed before it can be used for analysis. In this chapter, the analyses are focused on steady state operation. As the data is coming in the time domain and is raw, it is needed to make sure whether it can be utilized for the need or not. To detect faults in the early stage, it is reasonable to consider small frequency components by taking Fourier transforms of the incoming signal. For effective fault detection, different operating conditions must be considered, such as control environment, load, ambient environment, etc. Figure 6 presents the current frequency spectra of a motor with broken rotor bars in several control modes – grid fed, scalar, and direct-torque control. As seen, a significant shifting in frequency components occurs between the signals in different control modes. This is important to be considered during the model training.

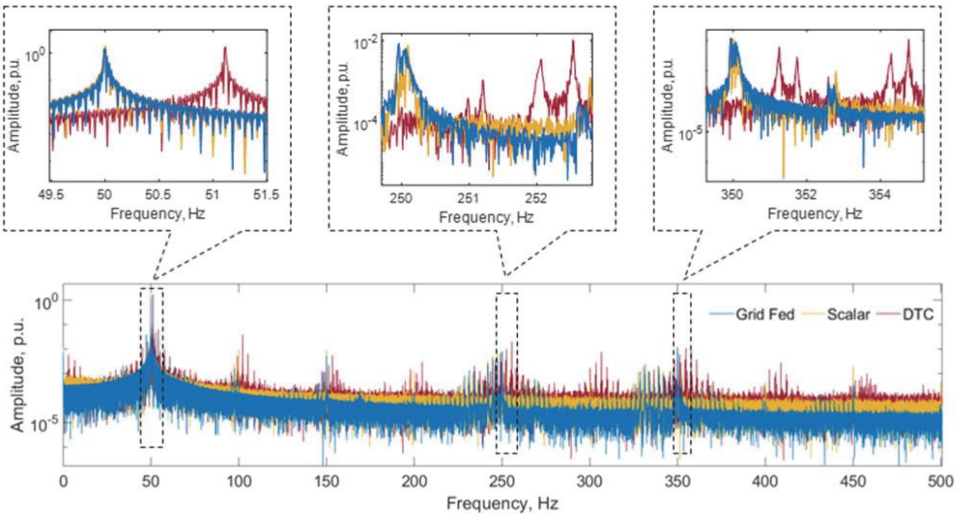


Figure 6. Current frequency spectra of a faulty motor under different control modes.

At the same time, load also should be considered. Figure 7 presents the current frequency spectra of a motor with broken rotor bars under different loads. It is seen that the behavior of the signal changes as the load increases.

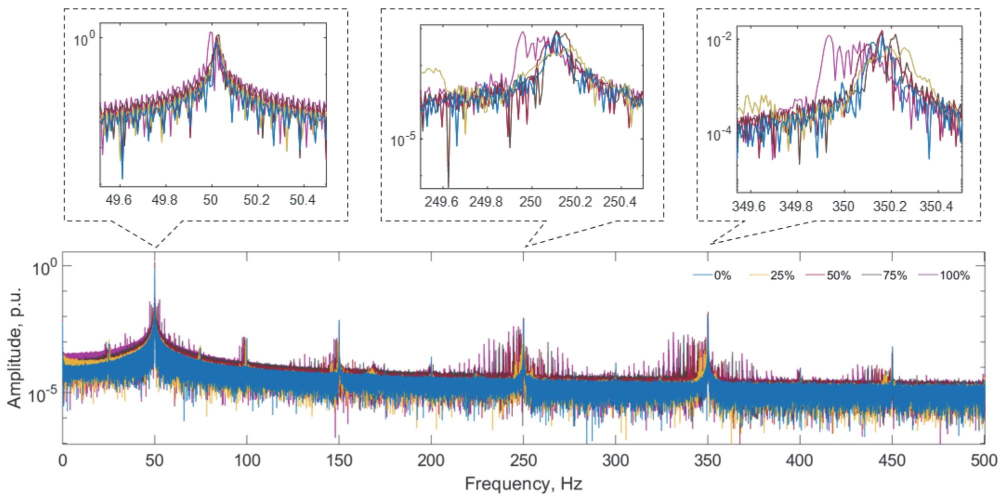


Figure 7. Current frequency spectra of a faulty motor under different loads.

In both cases, there are two regions to be studied to make predictions. Firstly, the frequency range of 0-500 Hz, where the impact of the fault is the highest on even harmonics. Specifically, the most prominent are harmonics on 50, 250, and 350 Hz. Besides, harmonics at 750 Hz can be important to be studied and considered for fault prediction.

The data is first converted into the time domain and sampled according to the sampling frequency to make sure we have enough cycles. Figure 8 shows an example of sample data set in the frequency domain. As the time domain does not have significant components based on which healthy and faulty data can be distinguished; hence, the data is converted into the frequency domain first and the frequency spectrum is analyzed to find the specific difference between the healthy and the faulty electrical machines.

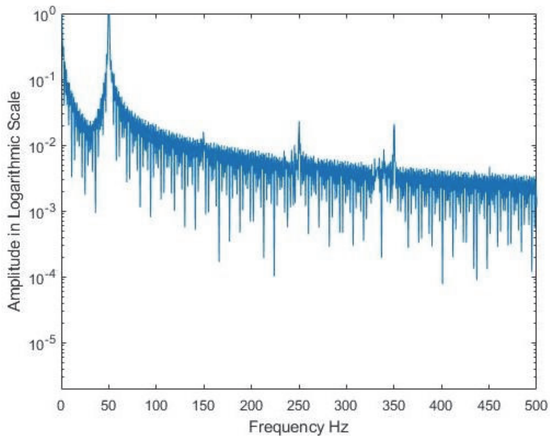


Figure 8. The frequency spectrum of the signal to be trained.

The frequency spectrum of a faulty electrical machine includes different frequency components usually not present in a healthy electrical machine frequency spectrum. Identifying those components and utilizing different analyzing techniques to identify them in the incoming data is part of fault detection. Including those frequency components to extract as features for training different machine learning models can help identify electrical machines' faults. Fault detection can be divided into two parts: signal processing and machine learning trained algorithms. Different analyses based on fast Fourier transforms can be used for the signal processing part.

4. Machine Learning Algorithms

The most common technique used for the detection of faults at the moment is utilizing machine learning trained algorithms. With the advent of artificial intelligence, making self-learning or systems with the aptitude for the decision has helped streamline multiple processes. Machine learning algorithms help create a complex weighted combination based on training data that can be used later to deduce results for the incoming data. Figure 9 presents the mostly spread machine learning algorithms [23].

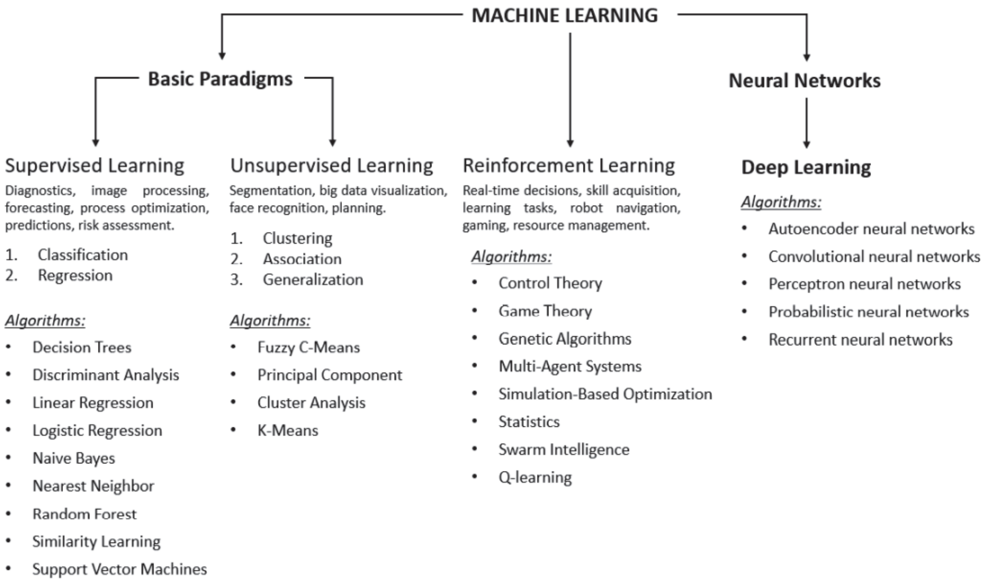


Figure 9. Machine learning algorithms [23].

One of the primary drawbacks of machine learning algorithms is that they need a lot of data to train a high-accuracy model. However, it usually depends on the complexity of the model. Suppose the model will be used for classification, with classification being divided into two labels. In that case, the accuracy will be pretty high even with a low training data set. But suppose that is to be changed by classifying the classification into four different labels. In that case, the system's complexity will increase, resulting in the algorithm needing more data to make an accurate model. Figure 10 shows the general working of a machine learning or neural network model, to be precise.

There are different types of machine learning algorithms based on specific logic. The training data set results in a statistical complex function based on the selected algorithm that gives a trained model. Among the machine learning algorithms, the most used are neural networks. Neural networks are further divided into three main types:

- Artificial Neural Network (ANN),
- Recurrent Neural Network (RNN),
- Convolution Neural Network (CNN),

ANN and RNN are primarily used for training for models related to detection or prediction. Most ANN models are regression-based or feed-forward models, whereas RNN is feed backward neural network models. Neural network model training is divided into three layers: the input layer, the hidden layer, and the output layer. The hidden layer is where the weighted nodes are set up, as the weight of these nodes is adjusted with each training data set. Once the model is trained using the training data set, a blind validation can be carried out to test the accuracy of the model before implementing it in a real-time scenario.

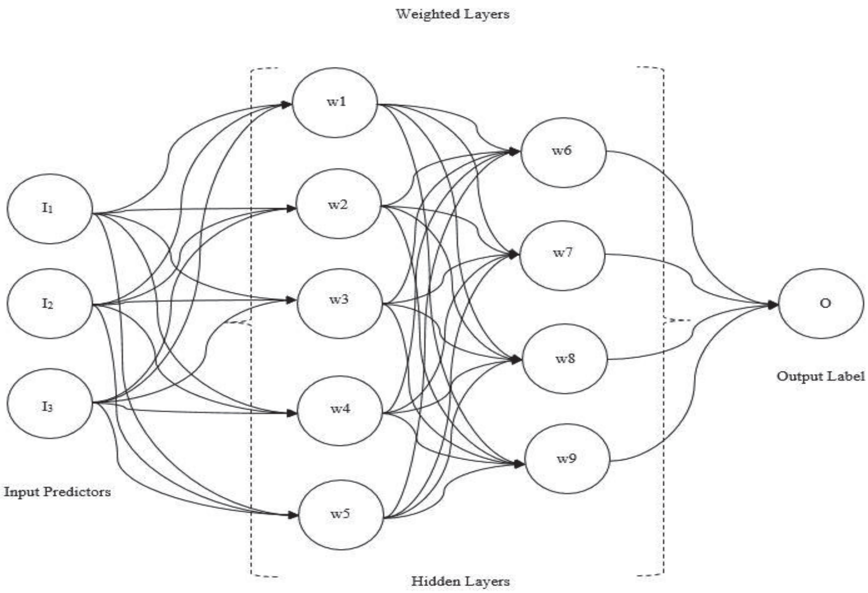


Figure 10. Neural network schematics.

These models, after training, are usually implemented in the cloud and are used to detect faults in the incoming signals. Although they can be trained to be precise, the data needed for it is usually great. That is why researchers are looking into generating such data programmatically based on the real-time collected data and frequency harmonics. If this is reached, it will be possible to mass produce faulty data according to the need of the electrical machine for training a machine learning model according to the need. In the future, it might also be possible to implement these models onto the edge or nodes to move the computation from the cloud toward edge computing.

Training of machine learning models also has other issues with accuracy based on the complexity of the system. Table 2 shows an example results from a set of experiments that confirms that changes in complexity or size of a database do impact the accuracy of different machine learning models. In this specific example, the algorithms were run with specific conditions to compare them under similar training and validation processes. However, the results can still be optimized as the training process (i.e., epochs, ...) and the test approach (i.e., v-fold cross-validation, holdout validation, …) can also result in different results. Hence, changing approaches can result in better or, even in some cases, worst performance, for example, a trilayered neural network with two categories and a smaller training set can result in an overfitted model.

Table 2. Accuracy comparison of different neural network models.

Neural Network Algorithm	Smaller Training Set		Bigger Training Set	
	Two Categories	Four Categories	Two Categories	Four Categories
Narrow Neural Network	88.30%	65.00%	70.00%	38.50%
Medium Neural Network	82.50%	63.30%	73.50%	46.50%
Wide Neural Network	88.30%	73.30%	76.20%	51.50%
Bilayered Neural Network	82.10%	62.10%	75.20%	43.10%
Trilayered Neural Network	95.40%	64.20%	73.3%	53.40%

As the system becomes more complex, a larger number of data is needed, but this also shows that there is a chance that another machine learning algorithm can perform better for the same scenario. Hence, these trained models are still flexible and there is a need to either get the optimal number of data sets for the training of the machine learning-based models or implement a custom-made machine learning model that can help identify faults related explicitly to electrical machines with high accuracy.

5. Predictive Maintenance

As the industry is moving towards predictive maintenance from scheduled maintenance, there is still much research to be done in this area. Most of the research going on is related to fault detection rather than fault prediction, but there are companies working in this specific area. The most important thing in this field is to identify the faulty frequency components in the early stage of the fault and the behavior of the signal and its frequency components when the signal is shifting from a healthy state to a faulty state. Once these things are identified, the next step is to train such a model that will be able to predict whether the fault is going to occur and in how much time. This will depend based on pre-processing of data and classification of the components. This is not a small task and needs dedication and time.

Researchers are looking for better ways to get a prediction model for faults to help identify them even before they occur. This leads us towards predictive maintenance, there might be some companies that are already running some kind of predictive maintenance algorithms with their systems, but at the moment, the hardware setup they have to use alongside it is quite expensive. So, another main issue in this area is to make it such that it is not only cheap but no specific hardware setup is needed in this regard. There are also multiple directions in which predictive maintenance trained algorithms can be utilized. There can be a combination of different algorithms to get higher accuracy or more accurate results. Similarly, fuzzy logic systems can also be used in accordance with machine learning algorithms and signal processing to get a more accurate system for predictive maintenance.

Another issue that the researchers commonly face in this aspect is the lack of data. As the data collected in an industrial environment is limited, especially in the case of faults, training a model with quality data and testing it out is quite difficult. Also, the data required to train models properly should be good in quality and quantity. Some researchers are working on observing the pattern in different faults to generate a statistical equation for the faults so that synthetic signals can be generated, which can help cover up this issue. The main issue in this aspect is to correctly identify the range of amplitude of frequency components that are generated when a fault is present in the electrical machine. This is not easy as it requires much data analysis and robust testing, but immense research is taking place in this direction.

6. Conclusion

The industrial revolution and information technology advancements have opened up new research areas to make things more convenient for industrial applications. IoT, with its usage in condition monitoring, fault detection, and remote controlling, is already becoming the norm for the industry. It will be more important in the near future to implement predictive maintenance for the industry to move away from scheduled maintenance to cut short on losses. Hence, fault diagnostics and predictive maintenance are the need of the hour. Here, a concise overview of a condition monitoring system is given along with the issues in the machine learning algorithm and the possibilities of predictive maintenance are discussed.

Although there are still many limitations, such as micro-controller boards are still in development, fault prognostics, limitation of available data, lack of statistical and predictive models. However, much research is being done in these areas, with the micro-controller boards being advanced rapidly, making them more reliable and stable. An increase in their computational power will also result in a more stable and quicker transmission of data. The bigger problem is still related to lack of data, resulting in trained models not being up to the mark. However, researchers are currently developing statistical models by reengineering. By observing the signals for different faults from an electrical machine, researchers are trying to develop statistical models that can generate signals similar to the fault. Although the process takes much time and concentration, researchers are getting near and it might be possible in the near future to generate faulty signals based on statistical models.

This chapter discusses a specific version of a condition monitoring system with discussion related to maintenance approaches, machine learning algorithms and some of the issues faced in this aspect.

Acknowledgments

This research was funded by [EEA]/[Norway] Grants 2014–2021, “Industrial Internet methods for electrical energy conversion systems monitoring and diagnostics”.

References

[1] Asad B, Vaimann T, Belahcen A, Kallaste A. Broken rotor bar fault diagnostic of inverter fed induction motor using FFT, Hilbert and Park’s Vector approach. In: Proc. - 2018 23rd Int. Conf. Electr. Mach. ICEM 2018, pp. 2352–2358 (2018) doi: 10.1109/ICELMACH.2018.8506957.

[2] Vaimann T, Sobra J, Belahcen A, Rassõlkin A, Rolak A, Kallaste A. Induction machine fault detection using smartphone recorded audible noise. IET Sci. Meas. Technol. (2018) doi: 10.1049/iet-smt.2017.0104.

[3] Vaimann T, Kallaste A, Kilk A: Sensorless Detection of Induction Motor Rotor Faults Using the Clarke Vector Approach. Sci. J. Riga Tech. Univ. Power Electr. Eng., vol. 28, no. 1, pp. 16–21 (2011) doi: 10.2478/v10144-011-0007-9.

[4] Sathyan S, Belahcen A, Kataja J, Vaimann T, Sobra J. Computation of stator vibration of an induction motor using nodal magnetic forces. In: Proc. - 2016 22nd Int. Conf. Electr. Mach. ICEM 2016, pp. 2198–2203 (2016) doi: 10.1109/ICELMACH.2016.7732827.

[5] Kudelina K, Asad B, Vaimann T, Rassõlkin A, Kallaste A, Khang HV. Methods of Condition Monitoring and Fault Detection for Electrical Machines. Energies. 2021; 14(22):7459. https://doi.org/10.3390/en14227459

[6] Barksdale H, Smith Q, Khan M. Condition Monitoring of Electrical Machines with Internet of Things. In: Conf. Proc. - IEEE SOUTHEASTCON, vol. 2018-April, pp. 1–4 (2018) doi: 10.1109/SECON.2018.8478989.

[7] Parpla R C, Lacob R. Application of IoT concept on predictive maintenance of industrial equipment. In: MATEC Web Conf., vol. 121, pp. 1–8 (2017) doi: 10.1051/matecconf/201712102008.

[8] Mohanraj M, Thottungal R, Jaikumar K. A Can Bus Based Fault Monitoring and Diagnosis System in Wind Turbine. In: 2013 International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT), pp. 1-3 (2013) doi: 10.1109/ICEVENT.2013.6496547.

[9] Wymore M L, Van Dam J E, Ceylan H, Quai D: A survey of health monitoring systems for wind turbines. Renew. Sustain. Energy Rev., vol. 52, no. 1069283, pp. 976–990 (2015) doi: 10.1016/j.rser.2015.07.110.

[10] Mukherjee S, Ghosh A, Sarkar S K. Arduino based Wireless Heart-rate Monitoring system with Automatic SOS Message and/or Call facility using SIM900A GSM Module. In: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), pp. 1-5 (2019) doi: 10.1109/ViTECoN.2019.8899504.

[11] Edwan E, Abu-Musameh M, Alsabah A. Blood Pressure Monitoring Using Arduino-Android Platform. In: 2020 International Conference on Assistive and Rehabilitation Technologies (iCareTech), pp. 87-91 (2020) doi: 10.1109/iCareTech49914.2020.00024.

[12] Haider-e-Karar, Khuwaja A A, Sattar A. Solar power remote monitoring and controlling using Arduino, LabVIEW and web browser. In: 2015 Power Generation System and Renewable Energy Technologies (PGSRET), pp. 1-4 (2015) doi: 10.1109/PGSRET.2015.7312240.

[13] Janeera D A, Poovizhi H, Sheik Haseena S S, Nivetha S. Smart Embedded Framework using Arduino and IoT for Real-Time Noise and Air Pollution Monitoring and Alert system. In: 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), pp. 1416-1420 (2021) doi: 10.1109/ICAIS50930.2021.9396041.

[14] Xie P et al. Sensing wind for environmental and energy applications. In: IET Conf. Publ., vol. 2014, no. CP639, pp. 64–69 (2014) doi: 10.1049/cp.2014.0660.

[15] Mabrouki J, Azrour M, Dhiba D, Farhaoui Y, Hajjaji S E: IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts. Big Data Mining and Analytics, vol. 4, no. 1, pp. 25-32, March (2021) doi: 10.26599/BDMA.2020.9020018.

[16] Sharma P, Kapoor S R. Evaluation of arduino based das for condition monitoring of induction motor. In: 2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC), 2017, pp. 1-4, doi: 10.1109/ICOMICON.2017.8279084.

[17] Andrei O, Ungureanu S, Miron A, Cziker A C. IoT power monitoring device using Wi-Fi and Arduino. In: 2021 9th International Conference on Modern Power Systems (MPS), 2021, pp. 1-6, doi: 10.1109/MPS52805.2021.9492651.

[18] Raja H, Vaimann T, Rassõlkin A, Kallaste A, Belahcen A. IoT Based Tools for Data Acquisition in Electrical Machines and Robotics. In: 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC), 2021, pp. 737-742, doi: 10.1109/PEMC48073.2021.9432553.

[19] Ahmad N, Bajwa S I S, Ahmad S, Kha L, Malik T A, Tahir Y. Autonomous Unmanned Surface Vehicle a microcontroller based approach. In: 2011 International Conference on Electric Information and Control Engineering, ICEICE 2011 - Proceedings, 2011, doi: 10.1109/ICEICE.2011.5778076.

[20] Kuts V, Otto T, Tähemaa T, Bukhari K, Pataraia T. Adaptive industrial robots using machine vision. In: ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE), 2018, doi: 10.1115/IMECE2018-86720.

[21] Kuts V et al.: Synchronizing physical factory and its digital twin throughan iiot middleware: A case study. Proc. Est. Acad. Sci., 2019, doi: 10.3176/proc.2019.4.03.

[22] Debbag Y, Yilmaz E N. Internet based monitoring and control of a wind turbine via PLC. In: 2015 3rd Int. Istanbul Smart Grid Congr. Fair, ICSG 2015, 2015, doi: 10.1109/SGCF.2015.7354935.

[23] Kudelina K, Vaimann T, Asad B, Rassõlkin A, Kallaste A, Demidova G. Trends and Challenges in Intelligent Condition Monitoring of Electrical Machines Using Machine Learning. *Applied Sciences*. 2021; 11(6):2761. https://doi.org/10.3390/app11062761.

Curriculum vitae

Personal data

Name:	Hadi Ashraf Raja
Date of birth:	08.12.1988
Place of birth:	Pakistan
Citizenship:	Pakistan
E-mail:	hadi.raja@taltech.ee

Education

2019–2023	Tallinn University of Technology, Estonia—PhD Electrical Power Engineering & Mechatronics
2012–2014	Myongji University, South Korea — MS Information and Communication Engineering
2006–2010	GC University (Lahore campus), Pakistan—BS Electrical Engineering

Language competence

English	Fluent
Urdu	Fluent (Native)
Punjabi	Fluent

Professional employment

2019–till date	Doctoral Researcher, Tallinn University of Technology, Estonia
2014–2016	Researcher, Nain Information, South Korea
2012–2014	Researcher, Myongji University, South Korea

Field of Research

- Electrical and electronics engineering
- Internet of Things
- Fault diagnosis and prognosis of electrical machines
- Machine learning and deep learning techniques
- Condition monitoring of electrical machines

Projects

- PSG453 "Digital twin for propulsion drive of autonomous electric vehicle" (1.01.2020−31.12.2024); Principal Investigator: Anton Rassõlkin; Tallinn University of Technology, School of Engineering, Department of Electrical Power Engineering and Mechatronics; Financier: Estonian Research Council; Financing: 430 500 EUR.
- PRG675 "New Generation of High-Performance Power Electronic Converters Simultaneously Applicable for DC and AC Grids with Extended Functionalities" (1.01.2020−31.12.2024); Principal Investigator: Oleksandr Husev; Tallinn University of Technology, School of Engineering, Department of Electrical Power Engineering and Mechatronics; Financier: Estonian Research Council; Financing: 742 150 EUR.
- ETAG21001 "Industrial internet methods for electrical energy conversion systems monitoring and diagnostics" (1.01.2021−31.12.2023); Principal Investigator: Toomas Vaimann; Tallinn University of Technology, School of Engineering, Department of Electrical Power Engineering and Mechatronics (partner); Financier: Research Council of Lithuania; Financing: 251 250 EUR.

Elulookirjeldus

Isikuandmed

Nimi:	Hadi Ashraf Raja
Sünniaeg:	08.12.1988
Sünnikoht:	Pakistan
Kodakondsus:	Pakistani
E-post:	hadi.raja@taltech.ee

Hariduskäik

2019–2023	Tallinna Tehnikaülikool, Eesti – PhD Elektroenergeetika ja Mehhatroonika
2007–2009	Myongji University, Lõuna-Korea – MS Information and Communication Engineering
2003–2007	GC Ülikool (Lahore'i linnak), Pakistan – BS Electrical Engineering

Keelteoskus

Inglise keel	kõrgtase
Urdu keel	emakeel
Pandžabi keel	kõrgtase

Teenistuskäik

2019–…	Doktorant, Tallinna Tehnikaülikool, Eesti
2014–2016	Teadlane, Nain Information, Lõuna-Korea
2012–2014	Teadlane, Myongji University, Lõuna-Korea

Teadustöö põhisuunad

- Elektrotehnika ja elektroonika
- Asjade internet
- Elektrimasinate veadiagnostika ja prognoosimine
- Masinõpe ja süvaõppemeetodid
- Elektrimasinate seisundi jälgimine

Jooksvad projektid

- PSG453 "Isejuhtiva elektrisõiduki veoajami digitaalne kaksik" (1.01.2020−31.12.2024); Vastutav täitja: Anton Rassõlkin; Tallinna Tehnikaülikool, Inseneriteaduskond, Elektroenergeetika ja mehhatroonika instituut; Finantseerija: Sihtasutus Eesti Teadusagentuur; Eraldatud summa: 430 500 EUR.
- PRG675 "Laiendatud funktsionaalsusega alalisvoolu ja vahelduvvooluvõrkude jaoks üheaegselt rakendatavad suure jõudlusega elektrilised elektroonilised muundurid" (1.01.2020−31.12.2024); Vastutav täitja: Oleksandr Husev; Tallinna Tehnikaülikool, Inseneriteaduskond, Elektroenergeetika ja mehhatroonika instituut; Finantseerija: Sihtasutus Eesti Teadusagentuur; Eraldatud summa: 742 150 EUR.
- ETAG21001 "Tööstuslikul internetil baseeruvad energiamuundussüsteemide seire- ja diagnostikameetodid" (1.01.2021−31.12.2023); Vastutav täitja: Toomas Vaimann; Tallinna Tehnikaülikool, Inseneriteaduskond, Elektroenergeetika ja mehhatroonika instituut (partner); Finantseerija: Research Council of Lithuania; Eraldatud summa: 251 250 EUR.

ISSN 2585-6901 (PDF)
ISBN 978-9949-83-982-7 (PDF)