AUTOMATED ELECTRICAL MOTOR QUALITY CONTROL VIA MACHINE LEARNING BASED VIBRATION ANALYSIS

by

Sibel Şentürk

B.S., Electrical and Electronics Engineering, Yeditepe University, 2015

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University
2021

ACKNOWLEDGEMENTS

First, I would like to start by sending my gratitude to my advisor, Professor Burak Acar, who encouraged me during the writing of this thesis and helped me carry out the whole process. Whenever I got lost, he illuminated my path by using his scientific background and being a good supervisor whenever I was confused. His guidance helped me a lot in completing this thesis. Secondly, I would like to send my thanks to Gürkan Aydemir, who has supported me with his ideas during this process, for his unparalleled support and friendship.

I would like to send my warmest thanks to Arçelik, which is one of the rare places where academia and industry work together today for providing me with a real case study subject and giving me space, time, and opportunity to focus and work in this field, and to Atölye 4.0 and Production Technologies Directorate. I would like to thank Turgut Köksal Yalçın, the manager of the Industrial Robotics team, with whom I have been working for about three years, and all my teammates for their support during this process. I am highly indebted to Haldun Dingeç, Vugar Kerimoğlu, and Ömer Faruk Özer for providing a research opportunity and supporting me. And also, I would like to thank WAT Motor R&D and Production teams, Serkan Yücesoy, Barış Tuğrul Ertuğrul, Kayhan Kızmaz, Mert Ali Çalduk, Cem İşbilir and Salim Mıstın.

I would like to thank Deniz Eroğlu and Abdülkadir Yazıcı for providing me significant infrastructure for data collection from the field and guiding me with their ideas and support every time I get stuck.

Unfortunately, during the preparation of this thesis, my dear father, Hasan Hüseyin Şentürk, who was one of the most important people in my life, passed away. I thank him very much for spending his last days with me, even by chance. I am sure he would be pleased to see this thesis completed. I would like to thank my dear mother, Havva Şentürk, for calming me down during this process and always being there for me. Life would be challenging to survive without her. I would like to thank my other family members Hatice Tezcan, Yılmaz Şentürk, Ziya Şentürk and Fatma Şentürk and my right-hand man Tuğra Şentürk.

I would like to thank my dear Hüseyin Büyükeşmeli, who is one of the gifts given to me in this life, for never hesitating to share my burden in life. I could not have completed this thesis without his support.

And I am indebted to my dear sisters, Merve Fidan, Yasemin Adıbelli, Burcu Seyitoğlu, Burçin Başyazıcı, Elif Uzunca, Ayşe Çapkulaç, Rabia Altıntaş and Betül Altıntaş Öner. They have always stood by me with their lovely friendship despite my negligence.

ABSTRACT

AUTOMATED ELECTRICAL MOTOR QUALITY CONTROL VIA MACHINE LEARNING BASED VIBRATION ANALYSIS

Industry 4.0 aims at the digital transformation to increase the reliability and capacity of production. With the integration of Sensor Analytics (SA) and Artificial Intelligence (AI) to manufacturing, the design of automated and optimized processes becomes more accessible. One of the areas where AI tools and SA are used is quality control tests of products. The main target of this thesis is to automate the quality control step based on vibration analysis by finding mechanical failures of Brushless Direct Current (BLDC) motors as an example of an AI-powered sensor analytics application. In addition, the feasibility and assessment of popular machine learning models are investigated. Two architectures are proposed to classify motors' quality. These methods are called sAIQC, Single-Stage AI-Powered Quality Control, and dAIQC, Double-Stage AI-Powered Quality Control. In the first method, motors are classified into healthy (pass) or faulty (fail), regardless of the data quality of the signal. The second proposed method is composed of two stages. The first stage makes a binary classification based on data quality, and then, the separated groups are classified at two independent classifiers in the second stage as pass or fail. Unweighted accuracy (UA), defined as the average accuracy of each class, is used as a performance metric of the classifiers. In experiments with the dataset containing 671 samples, the performance of sAIQC method was 84.9%; this performance with the dAIQC method was increased to 92.9%. Furthermore, in experiments using big data set consisting of 25580 vibration recordings and without a data quality label, the performance of the SAIQC method is 73.5% percent. In contrast, the performance of the dAIQC method is 89.5% percent.

ÖZET

MAKİNE ÖĞRENİMİNE DAYALI TİTREŞİM ANALİZİ İLE ELEKTRİK MOTORU OTOMATİK KALİTE KONTROLÜ

Endüstri 4.0 üretimin güvenilirliğini ve kapasitesini artırmak için dijital dönüşümü hedeflemektedir. Sensör Analitiği (SA) ve Yapay Zeka'nın (AI) üretime entegrasyonu ile otomatikleştirilmiş ve optimize edilmiş süreçlerin tasarımı daha erişilebilir hale gelmektedir. AI araçlarının ve SA'nin kullanıldığı alanlardan biri ürünlerin kalite kontrol testleridir. Bu tezin ana hedefi, AI destekli bir sensör analitiği uygulaması örneği olarak Fırçasız DC motorların (BLDC) mekanik arızalarını bularak titreşim analizine dayalı kalite kontrol adımını otomatikleştirmektir. Ayrıca popüler makine öğrenimi modellerinin uygulanabilirliği ve değerlendirilmesi araştırılmıştır. Motorların kalitesini sınıflandırmak için iki mimari önerilmiştir. Bu yöntemler sAIQC, Tek Aşamalı Yapay Zeka Destekli Kalite Kontrol ve dAIQC, Çift Aşamalı Yapay Zeka Destekli Kalite Kontrol olarak adlandırılır. İlk yöntemde motorlar, sinyalin veri kalitesinden bağımsız olarak sağlıklı(geçti) veya hatalı(kaldı) olarak sınıflandırılır. Önerilen ikinci yöntem iki aşamadan oluşur. İlk aşamada veri kalitesine göre ikili bir sınıflandırma yapılır ve daha sonra ayrılan gruplar, ikinci aşamada geçti veya kaldı olarak iki bağımsız sınıflandırıcı kullanılarak sınıflandırılır. Her sınıfın ortalama doğruluğu olarak tanımlanan ağırlıksız doğruluk, sınıflandırıcıların performans metriği olarak kullanılır. 671 örnek içeren veri seti ile yapılan deneylerde, sAIQC methodunun performansı %84,9 iken; dAIQC methodu ile bu performans %92,9'a yükseltilmiştir. Ayrıca 25580 titreşim kaydından oluşan ve veri kalite etiketi olmayan büyük veri seti kullanılarak yapılan deneylerde SAIQC methodunun performansı 73,5%'dir. Buna karşılık, dAIQC methodunun performansı 89,5%'dir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$A_k(n)$	Fast Fourier Transform of segmented signal
B_d	Rolling element diameter
C	The punishment variable of SVM
C_e	The cross-entopy function
c_i	i^{th} class
$criterion$	The criterion for splitting attributes
f_n	FFT frequency range
fs	Sampling Rate
f_{shaft}	Bearing shaft frequency
hop	The number of samples between each successive FFT window
i_f	The beginning point of steady state of current signal
i_l	The last point of steady state of current signal
K	The number of segments.
L	The length of segments of periodogram.
Le	The number of frames of STFT
l_j	j^{th} low dimensional equivalent of s_j
max_depth	The maximum depth of the tree
$MaxIndex$	Frequency points corresponding to the maximum amplitude
n	The size of data
$n\_estimators$	The number of trees in the forest
$nfft$	The number of FFT points
N_{up}	The number of uniq FFT points
N_r	Number of rotating elements in a bearing
P	Joint probability distribution in high-dimensional space
$p(c_i)$	Probability of the class c_i in a node
P_d	Bearing pitch diameter
p_i	Probability of i^{th} sample
P_i	The output of the softmax function for i^{th} sample

p_{ij}	Joint probability
$P(f)$	Power spectrum density
$p_{j\|i}$	Probability of s_j chosen by s_i as its neighbor sample
p_m	the mid-point of steady state of current signal
Pp_i	Previous probability
Q	Joint probability distribution in low-dimensional space
q_{ij}	Joint probability in high dimensional space
$q_{j\|i}$	Low dimensional counterpart of $p_{j\|i}$
$Rdiff$	The succesive differences of each element in $MaxIndex$ array
R_i	The residuals
s_i	i^{th} n dimensional observation in s
$status$	The flag indicating that the data was collected correctly
$subsample$	Subsample percentages
V_{RMS}	The RMS value of periodogram
W	The weights
$\mathbf{W_s}$	Normal vector to hyperplane
w_{0s}	The offset of hyperplane
w_{len}	Determination of window length
w_0	The bias value
Win	Window vector
$\bar{x}$	Mean of observation samples
x_{clen}	The length of current signal
x_{input}	The input data of the perceptron
x_i	Observation sample in sample data x
xw	Windowing signal
y_i	The actual label

γ	The influence degree of SVM for a single training example
δ	The gradient
η	Learning rate
λ	The regularization parameter og gradient boosting

ϕ	Rolling element contact angle
σ_i	Gaussian kernel function
σ^2	Variance of sample data x
χ	The signal
χ_c	Current signal
$\chi_k(j)$	Segmented signal

LIST OF ACRONYMS/ABBREVIATIONS

AE	Auto-encoder
AI	Artificial Intelligence
AIQC	AI-Powered Quality Control
ANN	Artificial Neural Network
BDA	Big Data Analytics
BLDC	Brushless Direct Current
BPF	Ball pass frequency
BPFI	Ball pass frequency inner race
BPFO	Ball pass frequency outer race
BPNN	Backpropagation Neural Network
CC	Cloud Computing
CNN	Convolutional Neural Network
CPS	Cyber-Physical Systems
CV	Cross-validation
CWT	Continuous Wavelet Transform
dAIQC	Double-Stage AI-Powered Quality Control
DAQ	Data-Acquisition
DC	Direct Current
EEMD	Ensemble Empirical Mode Decomposition
FFT	Fast Fourier Transform
FN	False Negative
FP	False Positive
FTF	Fundamental train frequency
H	Harmonics
HHT	Hilbert–Huang Transform
IMF	Intrinsic Mode Function
IoT	Internet of Things
KL	Kullback-Leibler

KNN	k-Nearest Neighbor
LDA	Linear Discriminant Analysis
LSTM	Long Short-Term Memory
MD	Mahalanobis Distances
Mis	Rotor Misaligment
MGWO	Modified Gray Optimization
ML	Machine Learning
MLP	Multilayer Perceptron
MS	Mahalanobis Space
MSE	Mean Square Error
NN	Nearest Neighbour
PCA	Principal Component Analysis
PSD	Power Spectrum Density
PVT	Peak Value of Vibration
QLD	Quality Labelled Dataset
QULD	Quality Unlabelled Dataset
R&D	Research & Development
ReLU	Rectified Linear Unit
RF	Random Forest
RMS	Root-mean-square
RNN	Recurrent Neural Networks
RPM	Revolution Per Minute
RS	Random Search
SA	Sensor Analytics
sAIQC	Single-Stage AI-Powered Quality Control
SGD	Stochastic Gradient Descent
SNE	Stochastic Neighbor Embedding
Sound	Vibration Fault Sound
STFT	Short Time Fourier Transform
SVM	Support Vector Machine
TN	True Negative

TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
t-SNE	t-Stochastic Neighbor Embedding
UA	Unweighted Accuracy
Unb	Unbalance Defect
Vib1	Vibration Fault 1
Vib1&2	Vibration Fault 1&2
Vib2	Vibration Fault 2
Vib3	Vibration Fault 3
XGB	Extreme Gradient Boosting
ZCR	Zero-crossing rate

1. INTRODUCTION

The rapid advancement of technology has increased the need for modern industrial systems and facilities. This need led to the beginning of the Industry 4.0 era, called the fourth industrial revolution. It centers around the evolution and variation in manufacturing systems. Industry 4.0 aims at developing intelligent manufacturing systems by adapting technology to production sites and factories. The growth of Industry 4.0 is supported by several new technologies, including Cyber-Physical Systems (CPS), Internet of Things (IoT), Big Data Analytics (BDA), and Cloud Computing (CC). These concepts have some common goals, such as optimizing different manufacturing resources and intelligent decision-making in manufacturing systems [1]. CPS is the technology that aims to associate software with physical components. Its aims for a developed version of coordination with physical objects and computational elements [2]. Furthermore, IoT enables any physical object that includes sensors or has software that makes it possible to connect and transfer data through the internet. IoT and related technologies empowered the human-to-machine, machine-to-machine, and human-to-human connections using small-sized sensors and wireless devices connected to the internet. As a result, real-time data collection systems and the traceability of the manufacturing processes could be realized. At the same time, these large and complex data create Big Data. The developed effective analytical methods called BDA provide aid to many organizations as they improve the decision-making quality and operational efficiency. Enabling traceability and real-time data collection with the help of intelligent sensors, developed software utilizing Machine Learning (ML) driven decision-making, and cloud storage systems pave the way for real-time smart manufacturing [3]. The smart manufacturing concept impacts and causes to change many the manufacturing processes such as quality control, maintenance, logistics, and so on.

Quality 4.0 is an emerging concept and a branch of Industry 4.0. It scopes the context for manufacturers to consider when introducing these new technologies. It additionally explains and categorizes the technologies and practices that Industry 4.0 contains and uses them to develop quality standards in the Industry. The key technologies of Industry 4.0 improve quality management in many ways as they enable digitalization and thus smart production. With the help of data collection and processing, it is now possible to predict and prevent issues related to the quality of the overall process and monitor the maintenance needs. Conventionally, quality control was dependent on sampling plans and inspection. However, after realizing Industry 4.0, quality control is realized by implementing BDA, CPS, and other technologies aimed at automation in manufacturing sites. The more improved technologies help producers catch any defects or delays earlier and take action respectively and increase the reliability of the overall manufacturing process.

Quality 4.0 includes predictive quality analysis, machine learning-powered quality control. The predictive quality analysis provides manufacturers with forecasting the quality of the products. It begins with the BDA; collected data is interpreted with statistical algorithms or machine learning to get insightful information. This information includes anomalies and critical issues in operation and predicts future effects and outcomes. These can be addressed before any quality issues occur, and thus overall quality can be increased [4]. The new generation of quality control systems powered by machine learning and deep learning methods provides the manufacturer with more reliable and generalized test systems without a human expert.

The ML-based quality control abolished limitations such as operator dependency, limited samples, limited calculation power, and knowledge expertise. ML algorithms can make decisions based on collected data, and they have learning capabilities based on mimicking the operator's decision-making process. And thus, it can advance human-level intelligence using the dataset. These intelligent systems took most of the work performed by operators. Most of the tasks are linked to CPS and AI based systems, and as a result, human errors were eliminated [5]. Thus, the product's quality perception increases from the customer side. With this perspective, Arçelik tries to use machine learning supported systems more widely to make the production process more efficient. It attaches importance to increasing the brand's quality perception of the customers and maintaining its competitive power by making the production stages smarter. Therefore, it aims to make human-independent and more reliable tests with the help of machine learning algorithms. This context seeks to fully automate the vibration and acoustic quality control tests of BLDC motors. Thus, this dissertation covers the vibration and acoustic quality control test of BLDC produced in the Arçelik Electric Motors factory independent from human beings with modern machine learning techniques.

1.1. Current Quality Control System

Arçelik produces BLDC motors for washing machines. The drum of the washing machine is where the clothes are washed, rinsed, and dried. BLDC motors are used to turn drums while clothes are washed. After the production of the motor, a set of tests are performed to be sure of the engine's performance. Vibration and acoustic control is one of the test methods currently used at the Arçelik plant, and passed motors are lastly controlled by operator check to make the final decision based on its quality. The detection of mechanical faults of the engine is the aim of this quality control step.

The existing functional vibration and acoustic test system consists of two parallel steps: automatic testing and listening test, a manual inspection stage by the operator. Due to the automated system's lack of accuracy and analytic power, listening by a human expert is required. The average proportion of faulty motors only detected by the operator at the vibration sound control point is 40%. The motor model investigated in this study covers nearly 50% of the annual output. The several production challenges have also been solved. The amount of motors returned from Arçelik Washing Machine Factory, a domestic customer and end-user, is extremely low. On the other hand, the human-dependent decision-making system violates Industry 4.0 standards. At the same time, the lack of a quantitative evaluation methodology, the age-related changes in human hearing ability and range, the fact that judgments concerning motor health differ from person to person, and the absence of specialized staff make it difficult for reliability.

The automated system is currently being used to determine each motor's spatial power spectrum energies of different frequency ranges by running it for a short period of time, almost seven seconds. When the engine reaches a steady-state, vibration analysis starts. After that, the projected energy values are compared to the R&D vibration team's established thresholds. In parallel, the operator mounts an accelerometer to the motor to start above mentioned vibration analysis and listens to the produced sound during the automatic test period in a limited noise isolated cabinet. Operators have the authority to overturn the automated outcomes of vibration analysis, giving them the final decision about the quality of the motor. For example, based on the unpleasant noises detected during the test, the result of the automated system that initially reads pass can be overridden as fail. A fail judgment being reversed to pass by a human operator is also a possibility. The existing system only offers a limited amount of information on the root cause of failures.

Five evaluation criteria are used in the current end-of-line vibration analysis system, including Root-Mean-Square (RMS) energies for four frequency bands and counting local maxima peaks within the given frequency range. The RMS energy between two frequencies is computed by generating the Fast Fourier Transfrom (FFT) of raw vibration signals and calculating the RMS of a specific frequency interval, as to be discussed in detail in Section 4.2.

Based on knowledge gained through experience, the responsible R&D department of Arçelik has established relevant frequency ranges. The first range is 50 to 200 Hz, which includes the signatures of imbalance and cage faults as discussed in Section 4.2. The second estimated value is 50 Hz to 1000 Hz, consisting of all additional bearing fault frequencies. Third, calculated by multiplying the number of rotor magnets, which is 8, with central frequency, 50 Hz to 1575 Hz, is another interval representing rotor dismounting or magnet failures. The fourth and final value-based RMS computation is 50 Hz to 2000 Hz. The number of peaks between 2000 and 6400 Hz employed for high vibration motor detection is the last criterion. The fourth parameter's threshold is set to a pretty high value, and it may be claimed that it is rarely used.

1.2. Problem Statement

A fully automatic test bench capable of performing noise and vibration quality control tests on the entire production would be desirable for motor manufacturers. However, such a station is challenging to deploy because of the significant background noise and vibration generated by conveyors and other moving machinery along the line, and the production cycle is another limitation for signal acquisition and processing. For BLDC motors, noise and vibration measurements are directly connected; but for others, they are simply complementary.

The primary goal of this thesis is to develop a machine-learning-driven quality control systems of BLDC motors that incorporates all of the useful features gleaned from domain-expert and academic literature knowledge. Furthermore, the other objectives of this study are to make a feasibility study of machine learning techniques in this field and analyze the comparison of their performance. For this purpose, an installed test bench is used to automatically gather vibration and current signals from the motors at the end of the production line. The test bench is low-cost and not isolated from ambient noise. Therefore, preprocessing methods have been developed to handle difficulties affecting the quality of the received data. At the same time, inconveniences are observed while automatically mounting the accelerometer into the motor body. Since the improper mounting problem can not be eliminated by preprocessing, a double-stage method has been studied to tackle this problem.

The rest of the thesis is summarized as follows: Chapter 2 introduces the construction of BLDC motors and the possible defects of engines. Then, the fault detection, diagnosis, and classification studies in this field are reviewed. The machine learning models and model evaluation techniques utilized are described. Moreover, the data visualization tool is summarized. In Chapter 3, data collection infrastructure and encountered challenges, the procedures for data exclusion, and preprocessing techniques on vibration signals are explained. The extracted features and created datasets are given in detail. Proposed AI-powered quality control methods are described in Chapter 4. Experiments are conducted based on these datasets provided. In addition, the results and discussions of the experiments are discussed in Chapter 4. The final discussion is summarized in Chapter 5. Main conclusions and contributions are given in Chapter 6.

2. BACKGROUND

The brief information about the dynamic parts of the BLDC motor and their possible mechanical faults and the possible reasons during their production are introduced in Section 2.1. State-of-the-art baseline methods for machinery diagnostic are described in Section 2.3. The machine learning methods and evaluation techniques are summarized in Section 2.4. The visualization method is described in Section 2.5.

2.1. Brushless DC Motors

Brushless and brushed DC motors are used to transfer electrical energy into mechanical energy. Their mechanical structural are similar. The BLDC motor is comparable to a BDC motor, but unlike a BDC Motor, a BLDC uses electronic commutation rather than brushes for commutation.

Brushes deliver power to the rotor in traditional BDC motors as they turn in a fixed magnetic field. Brushed motors have permanent magnets in the stator and coils on the main shaft. Brushes are used to apply electric current to the coils on the rotor shaft. The magnetic field formed in the coils reacts to the applied electric current so that it always coincides with the magnetic field of the magnets, moving the shaft. The voltage between its two terminals is adjusted to control the motor's speed. It is required to reverse the current regularly to sustain rotation.

Electronic commutation combined is employed in BLDC motors, which eliminates the need for mechanical brushes. In contrast to a BDC motor, the permanent magnets are on the rotor, and the coils are on the stator. As current runs through the stator coils, the poles on the rotor rotate in proportion to the electromagnetic poles formed within the stator. While the permanent magnet located in the rotor rotates, changes in the magnetic fields and generate rotation. The amount and direction of the current entering the coils located on the stator to regulate the spin can control the rotation.

BLDC motors are driven by DC voltage by switching one stator windings to the other, and current flows through one of the other three stator windings. Thus it generates alternating trapezoidal or quasi-square signal forms.

Brushless DC motors were developed to eliminate mechanical friction, sparks, and electrical faults caused by brushes and improve motor reliability [6]. Moreover, they offer benefits such as high efficiency, high torque, and ease of maintenance, low noise, and high speed [7]. BLDC motors are used commonly due to their lower energy consumption and maintenance requirements: automotive, household appliances, white goods, industrial controls, automation, and so on [7]. They are used for white household appliances that required high-speed control and dynamic response to varying loads, such as washing machines and dryers.

2.1.1. Construction

BLDC motors are consisting of two critical parts. They are stator and rotor. The motor's operation is accomplished by pushing and pulling forces that occur as magnetic forces between the rotor and the stator vary. Rolling element bearings help to transfer loads from the shaft to the motor frame. To provide non-stop rotation of the rotor, the sequence of energized windings of the stator is critical. To know rotor position and determine the energizing series of the windings, sensors or sensorless detection methods are used. The main parts are given in following figure[1] .

[1] Arçelik, İstanbul, TR

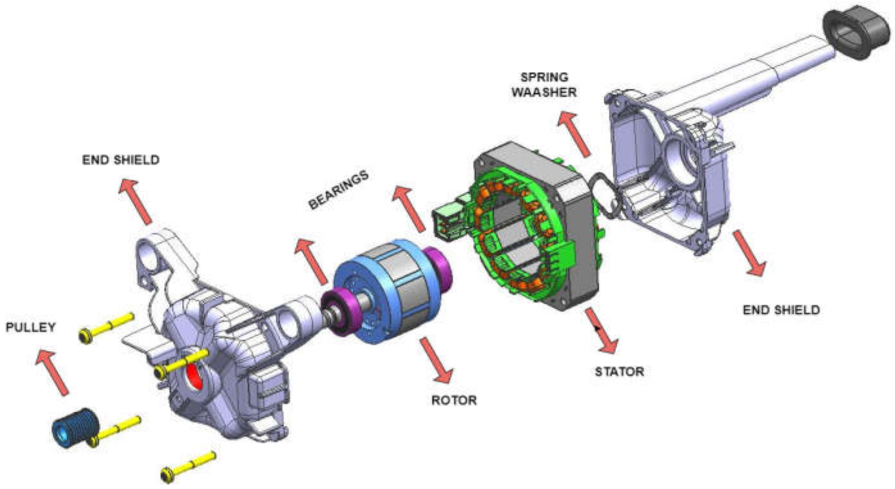

Figure 2.1. The BLDC motor and its components are studied within the scope of this thesis and produced by Arcelik Electric Motors.

- Stator: The stator is located in the stationary electrical component. The stator generates the revolving magnetic field by the windings. A BLDC motor's stator comprises stacked steel laminations with windings positioned in slots cut axially along the inner periphery [7]. Most of the BLDC motors are three-phase. Each phase switches on in turn for the rotor's rotation.
- Rotor: The rotor is the motor's rotating component. The rotor is comprised of permanent magnets with two to eight pole pairs facing the stator poles. The rotor is positioned on the motor's shaft and is housed within the stator. Rotor magnet cross-sections come in a variety of shapes and sizes. Rotor magnet cross-sections come in a variety of shapes and sizes [8]. In the following figure depicts cross-sections of different arrangements of magnets.

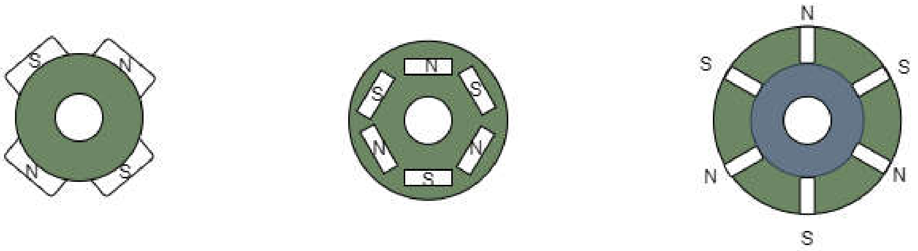

Figure 2.2. The cross-section of rotor magnets.

- Bearing: Rolling element bearings are used to stabilize the most reliable working condition of the motor with low noise, low erosion risk. The sliding motion replaces the rolling motion by using bearings. In this way, friction is significantly reduced, enabling high-speed rotation with minimal power loss. The most used roller types in bearing are ball, cylindrical, tapered, spherical [9]. Ball bearings are a common choice in electric motors. They have low friction and can operate at high speeds with quiet running. They are appropriate for both locating and non-locating positions since they can carry radial, axial, and combination loads. Cylindrical roller bearings are used to meet high load capacity where heavy radial loads prevail, such as belts-driven motors. Due to the high load-carrying capability of spherical roller bearings, they are preferred in large electric motors. Tapered roller bearings can support both radial and axial forces due to their assembly. It has a cup and cone arrangement. The outer ring makes up the cup, while the inner ring, rollers, and cage make up the cone assembly. This bearing design can handle multiple loads and has reduced friction while motors are running.

2.2. Possible Mechanical Defects & Their Vibration Signatures

Any defects on the engine's dynamic rolling and spinning parts as abovementioned are the primary abnormal vibration and noises generators. The reasons for mechanical failures in BLDC motors' production can be that they are not perfectly aligned, improper installation, material fatigues, insufficient lubrication, bearing & shaft currents. The commonly occurred mechanical defects are listed below:

- Eccentricity: The center of the rotor should be aligned the same as the stator geometrical center [10]. If an uneven air gap between the rotor and the stator occurs, the motor begins to wobble during running. Stator ovality, unbalanced bearing assembly on the rotor shaft, or rotor and stator mounting can be causes of this defect. Furthermore, these defects cause current leakages flowing through the stator windings, and the motors are heated [11]. Rotor misaligment and unbalance problems are counted in this category.
- Bearing Faults: Flaking, pitting, spalling or corrosion, brinelling defects can occur as material fatigues of the bearing. These faults create detectable vibrations and increased noise levels. Moreover, these damages are caused by current leakages to bearing surfaces [11].

Vibration monitoring indicates different frequencies of mechanical vibrations that correspond to distinct sources or failures in the machine. Even if all rotational components of the motor are free of flaws and correctly aligned, vibration is still generated. Any defect in the parts causes the vibration pattern to shift. If any rotational component has a defect or fault, impulses will be created periodically. When there is no fault, the rotational elements vibrate at their resonance frequency; however, when there is a fault, the vibration begins to demonstrate various frequencies as the rolling elements pass over it.

The extracted features from the vibration signal can serve as indicators of approaching failure. For example, the temporal and spectral characteristics of motors can define their health condition. The calculated features are explained in Section 2.2.1 and Section 2.2.2.

2.2.1. Temporal Features

Defective rotational components in the motor can be detected using root-mean-square (RMS), kurtosis, number of peaks, or amplitude of peaks of the vibration signal. Other possible fault indicators are elaborated as follows [12].

- The RMS indicates the amount of vibration energy and represents the overall vibration level. It is the square root of the arithmetic mean of the squares of the values and it is can be written as

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}, \tag{2.1}$$

where x is sample data with n observation sample.

- The variance is a metric for determining how variable something is. It corresponds to how far the signal deviates from its mean value. It displays the data set's spread for the signal. σ^2 can be obtained as

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2, \tag{2.2}$$

where $\bar{x}$ is the mean of sample data.

- Shock is detected by looking at the peak amplitude of the vibration signal (PVT) and it can be found as

$$PVT = \max(|x_i|). \tag{2.3}$$

- A localized fault causes the short-duration bursts to significantly affect the peak level but a negligible impact on the RMS level [13]. The crest factor is used to define the amount of influence of the shock across the present signal and to find harmonic distortion of the signal. It can only apply on steady signals. It can be calculated by using PVT and RMS values of a vibration signal as follows

$$\frac{PVT}{RMS} = \frac{\max(|x_i|)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}}. \tag{2.4}$$

- Kurtosis is a statistical parameter that is used to define a signal's characteristics. Kurtosis determines whether the data are heavy-tailed or light-tailed in comparison to a normal distribution. The high kurtosis means that a high proportion of data is outliers and low kurtosis tend to have lack of outliers. It quantifies the frequency of occurrences of significant peaks in a signal and it can be obtained as

$$\frac{\sum\limits_{i=1}^{n}(x_i-\bar{x})^4}{n\times\sigma^4}-3. \tag{2.5}$$

If the peaks are increasing fastly, it means that the product began to fail. It is used to determine the large peak accelerations into the random vibration signal.

- The clearance factor can be obtained as

$$\frac{PVT}{\left(\frac{1}{n}\sum\limits_{i=1}^{n}\sqrt{x_i}\right)^2}. \tag{2.6}$$

It has high values for healthy bearings and diminishes defective balls, defective outer races, and defective inner races.

- The impulse factor is used to determine how much impact is generated by the bearing failure. It can be ontained with division of the maximum absolute value by the mean of absolute value. It is computed as

$$\frac{PVT}{\frac{1}{n}\sum\limits_{i=1}^{n}|x_i|}. \tag{2.7}$$

- A shape factor is a variable influenced by the shape of an object but is unrelated to its dimensions [14]. It is obtained as

$$\frac{RMS}{\frac{1}{n}\sum\limits_{i=1}^{n}|x_i|}. \tag{2.8}$$

- The line integral is the sum of absolute differences between successive points and it is calculated as

$$\sum_{i=0}^{n}|x_{i+1}-x_i|. \tag{2.9}$$

- Peak-to-peak gives the obtaining the maximum displacement information, and it can be computed as

$$\max(x_i) - \min(x_i). \tag{2.10}$$

- Entropy measures the uncertainty and unpredictability associated with a sampled set of vibration data. The Shanon Entropy can be calculated as

$$\sum_{i=0}^{n} p_i \log p_i, \quad p_i = \frac{x_i^2}{\sum\limits_{i=0}^{n} x_i^2}. \tag{2.11}$$

- A skewness value indicates how vibrational data deviates from symmetric behavior. It is qualified as

$$\frac{\frac{1}{n}\sum\limits_{i=1}^{n}(x_i - \bar{x})^3}{\left(\sqrt{\frac{1}{n}\sum\limits_{i=1}^{n}(x_i - \bar{x})^2}\right)^3}. \tag{2.12}$$

  Faults can affect the symmetry of the distribution and thus increase the skewness level.

When the faults on rotational elements of the motor grow serious, the vibration signal becomes random, and the defect indicator is hidden [12,15]. Therefore, features in the time domain may not be adequate to detect the error in some circumstances; that's why spectral features are also used, as described in Section 2.2.2.

2.2.2. Spectral Features

FFT based solutions to find defects on mechanical faults are the most common techniques. Different analysis forms of FFT such as Short Fast Fourier Transform (STFT), Power Spectrum Density (PSD) are used in health monitoring applications. The power or RMS values of specific frequencies or frequency bands are used. Some particular frequencies to detect bearing faults, unbalance, or misalignment problems are calculated before in Taylor's work [16]. Most bearing manufacturers provide these signature frequencies to their users. Also, many prognostic and diagnostic applications in this field have widely used these frequencies, such as IFM real-time motor monitoring system in their VSE series [17]. If the geometry of bearings, the RPM of the motor f_{shaft} is known, the following critical frequencies in Hz can be computed as follows:

$$\textit{Ball pass frequency inner race(BPFI)} = \frac{N_r}{2} \times f_{shaft} \times \left(1 + \frac{B_d}{P_d} \times \cos\phi\right), \tag{2.13a}$$

$$\textit{Ball pass frequency outer race(BPFO)} = \frac{N_r}{2} \times f_{shaft} \times \left(1 - \frac{B_d}{P_d} \times \cos\phi\right), \tag{2.13b}$$

$$\textit{Fundamental train frequency(FTF)} = \frac{f_{shaft}}{2} \times \left(1 - \frac{B_d}{P_d} \times \cos\phi\right), \tag{2.13c}$$

$$\textit{Ball pass frequency(BPF)} = \frac{P_d}{2B_d} \times f_{shaft} \times \left[1 - \left(\frac{B_d}{P_d} \times \cos\phi\right)^2\right], \tag{2.13d}$$

where N_r denotes the number of rotating elements, f_{shaft} is the motor's main frequency, P_d is the pitch diameter, B_d is the rolling element diameter, and ϕ is the contact angle [12]. In other words, the unbalance and misalignment faults, for our cases it is eccentricity, can be observed at the following frequencies in Hz. They are defined as

$$\textit{Unbalance} = f_{shaft}, \tag{2.14a}$$

$$\textit{Misaligment} = 2 \times f_{shaft}, \tag{2.14b}$$

where f_{shaft} is the rotation frequency of rotor shaft of the engine.

2.3. State-of-the-Art Machinery Diagnostics

Examination of the literature on quality control systems of BLDC motors reveals that limited research has been conducted. The reason for this is twofold: The first is that datasets for these studies are not being shared much yet, and the available datasets are more suitable for the conduction of long-term, machine diagnosis and machine condition monitoring-predictive maintenance much more. So the main threads that have been concentrated in the past and today's work are fault detection, diagnosis, and machine monitoring. The vast of the datasets are created in a controlled environment, and no study is working with the dataset collected from the production line in this field. In the studies in the literature, current, vibration, and sound data have been used commonly. In the vast majority of studies with current signals, the motor is started after loaded. The proposed methods in the literature about the above-mentioned topics can be grouped into two. The first group is that the conventional methods have the formulas obtained from previous studies on the individual failures of dynamic spinning parts used in the engine. The systems constructed in the light of these formulas are mainly based on statistical methods for classification or detection. The second group is machine learning or deep learning-powered methods for machine diagnostic, prognostics, or fault detection using known features from previous studies or by learning informative features from raw signals.

Conventional methods can be summarized as follows; they calculate the spectral or temporal statistical features of collected signals, and then they compared them with the healthy condition of the engine by a method, which could be distance, index, ranges, or scores, ie. Concettoni, Cristialli, and Serrafini suggested a quality control method, even though not a BLDC, but for small DC motors [18]. They proposed a semi-automatic test bench prototype to collect current, noise, and vibration data on a small sample of good and faulty motors and develop a data analysis method. They collected signals while the motor ran at all possible assemble positions. The RMS of vibration and current signal of the engine and spectral features are calculated for each 100 ms frame. A damage score is computed using extracted features and is compared with a determined threshold to classify the product as healthy or faulty.

In mechanical terms, bearing defects lead to the preferentially alternating load torque. Trajin and colleagues have compared vibration and stator current analysis to detect faults on the bearing [19]. To do so, on artificially damaged bearings, a vibration spectrum analysis was proposed to detect and diagnose the faults, and it was compared to classical techniques of scalar indicators and advanced signal processing methods. Moreover, it has been reported that the consequences of rotating load torque oscillations on stator current have been recognized. A stator current spectral analysis-based automatic detector has been proposed and successfully implemented on localized faults and bearing damage.

Studies on fault diagnostics, based on vibration analysis, have focused on which vibrations produce which faults. For example, Agoston [20] provides a good summary of these results. She identifies the types of vibrations regarding the electrical motors as motor base vibrations, bearing vibrations, broken rotor bar vibrations, rotor bar passing frequency vibrations, twice line frequency vibrations, and motor unbalance. She also suggests detection and classification of the faults using vibration analysis.

Another study on fault classification for rotary machines, Mahalanobis-Taguchi system [21], a health index mainly used for data classification. The anomalies and faults in the cooling fan and induction motor were detected using feature data set extracted from the vibration signal, and the Mahalanobis Distances (MD) were calculated to determine different health conditions. Normal/healthy rotary machines have low MDs and are within the defined Mahalanobis Space (MS). Indicating that a fault or incipient fault has occurred if the MDs are larger and out of the MS. A recent study conducted by Kudelina focused on the bearing faults that were relatively less investigated in the literature [11]. They proposed researching bearing faults and diagnosing them. The most common bearing failures were discussed, along with their causes. This study proposes using acceleration sensors to detect bearing faults in BLDC motors. The spectrum can be used to determine the type and stage of development of the damage. The fundamental vibration frequency component's amplitude can be used to set a threshold. They compare the benefits and drawbacks of commonly used diagnostic techniques such as FFT, STFT, Wavelet Transform.

Machine learning algorithms are used frequently in recent studies. Glowacz, in his attempt to introduce a new method for bearings, stator, and rotor fault diagnostic methods of a single-phase induction motor [22]. The acoustic signals of the induction motor are used in his study. Five different defect types are analyzed. Differences of frequency spectra of acoustic signals and finding discriminative frequencies of states by comparing a threshold were developed and implemented to feature extraction from acoustic signals. The Nearest Neighbour (NN) classifier was used to perform the classification step. In another research, Verdugo and colleagues offering a prototype for an automatic quality control test system for brushed DC motors indicate that four different quantities are monitored to perform fault diagnosis, namely acoustic noise, mechanical vibration, voltage, and current [23]. The study utilized a three-step feature extraction, a dimensional reduction, and a neural network classifier. A dataset consisting of 66 motors and 92.4 % of the tested motors are predicted correctly.

Li [24] combined a model of Modified Gray Wolf Optimization algorithm (MGWO) and Support Vector Machine (SVM) to diagnose the fault conditions of the engines by using collected current signals from BLDC motors. The optimization of the hyper-parameters of SVM to increase the diagnostic accuracy of the model, MGWO is applied. The performance of proposed method reaches at least 90 % accuracy for each different fault types.

Lee [25] used the Hilbert–Huang transform (HHT) to extract the features of four different fault types of brushless DC motors' hall position sensor. The features are selected by using a method based on distance discriminant. After choosing the most relevant features related to different fault types, a Backpropagation Neural Network (BPNN) and Linear Discriminant Analysis (LDA) are used as classifiers. The fault diagnosis system obtained an accuracy rate of 96 %.

Shifat and Hur, in their two studies regarding the condition monitoring of BLDC motors, revealed important findings [26]. In the first study, predictive maintenance of a BLDC motor by conducting a thorough analysis of vibration signals that combined signal processing, statistical characteristics, and machine learning techniques. It is tried to find transition states from one operating condition to another. They determined three operating states of a lifetime of BLDC: healthy, near to failure, and failure. To properly decompose the signal and locate the states, Ensemble Empirical Mode Decomposition (EEMD) is used. The similarity index is used to select the Intrinsic Mode Function (IMF). Following this, selected IMF is processed using Continuous Wavelet Transform (CWT) for improved fault localization. To identify various defect situations, some motor health statistic features are also extracted. Later, the dimensionality of the features was decreased using the principal component analysis (PCA) technique and categorized using k-Nearest Neighbor classification (KNN). They can classify motors operating conditions with 98 % of accuracy. In their second study [27], they engaged an artificial neural network to understand fault characteristics of BLDC motors. A mechanical fault is created by getting a hole in the rotor body, while an electrical fault is created by shorting two neighboring windings together. A vibration signal is used to detect the mechanical fault, and the third harmonic motor current is used as an indicator of electrical fault. Their abovementioned previous study analyzes vibration signals. The main difference between the two studies the transition point between failure conditions is determined by the current signal and, an Artificial Neural Network (ANN) is utilized to categorize various fault features. The ANN model succeeded in classifying three distinct health conditions: healthy, mechanical failure, and electrical failure, with 98 percent accuracy.

There are many studies on machine health monitoring with powered deep learning methods are increased recently. The primary preferred deep learning models are Auto-encoder (AE), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) [28]. Sun [29] proposed to classify induction motor defects, a single AE-based neural network layer. In another study, Ince suggested a 1-D CNN, which can merge motors fault detection into a single learning system during extraction and classification [30]. Xiao and his friends [31] proposed that without any feature engineering, an LSTM (Long Short-Term Memory) neural network is used directly on raw acceleration data to acquire meaningful representations for classifying the health status of asynchronous motors. Another fault diagnostic and classification study belongs to Zimnickas [32]. His study is about identifying stator health conditions and categorization of fault types of the stator of BLDC motors. The spectral entropy of the vibration signals is calculated. After the standardization of these features, an LSTM model is trained with both raw vibration signals and extracted features that standardized spectral entropy. He observed that the model's performance trained with features could classify all working conditions and faults accurately.

2.4. Machine Learning Models Utilized & Their Evaluations

The fault diagnosis algorithms in literature can be roughly divided into two groups as mentioned in Section 2.3; rule-based and machine learning-based methods. Rule-based methods achieve their decision by a predefined set of rules. On the other hand, machine learning methods work by learning these rules based on given data. ML methods can be examined in two groups; classical methods and deep learning methods. Classical machine learning methods conduct the prediction or classification process based on extracted features of the input data. Besides, deep learning methods are commonly trained with raw vibration signal or frequency spectrum rather than a feature set. This study focuses on machine learning algorithms; the proposed methodology is tested and compared with classical and deep machine learning methods. The machine learning models utilized is described as follows, the methods for evaluating and optimizing hyper-parameters of these models are summarized in Section 2.4.5

2.4.1. Support Vector Machines

Support Vector Machines (SVM) is one of the fundamental methods in classical machine learning algorithms. It is a supervised learning method which classifies the n-dimensional data by *(n-1)*-dimensional separator. This separator is called a hyperplane which divides the learning data into two classes linearly. Finding the optimum hyperplane is the learning process. A hyperplane can be defined as group of points satisfying $\mathbf{W_s}^T\mathbf{x} + w_{0s} = 0$. Where $\mathbf{W_s}$ is the normal vector to hyperplane and w_{0s} determines the offset of the hyperplane.

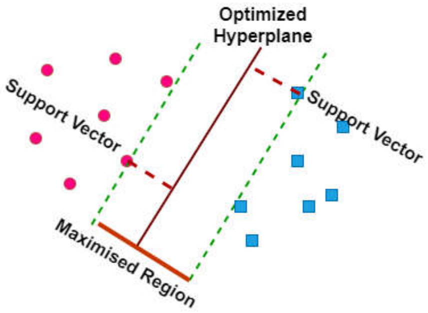

Figure 2.3. The hyperplane and support vectors are illustrated. Pink and blue points represent two classes. The optimum hyperplane can be found by maximizing the margin so that the distance from the closest sample from each class.

If the samples are linearly separable, the optimum hyperplane can be found by maximizing the margin so that the distance from the closest sample from each class. The hyperplane and margin are illustrated in Figure 2.3. In the cases where the data can not be linearly separable, the kernel trick is applied. The input data is mapped to a higher dimensional space with a kernel function, and the hyperplane is found in this space. One common approach for kernel function is the Gaussian radial basis function. Two hyper-parameters emerge for this kernel function γ and C. γ determines how far a single training example affects, and C is the punishment for the classified training samples. A high γ value will increase the variance of the Gaussian kernel function; hence the effect of the training sample would expand. On the contrary, a low value of γ decreases the impact area of a single training sample. C acts as a regularization parameter for SVM. High values of the C mean lower margins are acceptable. Low values would yield more smooth decision boundaries.

Effect of large values of γ, it becomes the dominant component and causes overfitting. If the γ is too low classifier can not grasp the shape of the data. While high C values will increase the fitting time, lower values lead to more support vector; hence the prediction time will increase. Thus C represents a trade-off between fitting.

2.4.2. Random Forest

Random Forest (RF) is an ensemble machine learning method that is based on decision trees. Decision trees build a predictive model based on observed data. Although they are very simple and highly effective in representing complex data, they also suffer from inaccuracy for the unseen data. RF overcomes this problem by an ensemble of week decision trees.

The underlying idea of the RF is to decide based on the majority of the multiple simple classifiers. To achieve this goal, RF creates many weak decision trees with bagging data. The term bagging stands for bootstrapping and aggregating the training data. Bagging data is constructed by randomly selecting the training samples of the same size as the training data. One entry in the training data can be selected multiple times for the bootstrapped data. A new set of data is created for each decision tree. The bootstrapping method yields a better performance since it decreases the variance without increasing the bias. Even if a single tree is sensitive to the noise on its training data, the average of trees will not be if the trees are uncorrelated. To break correlation between trees, Random Forest selects a random set of features for each tree. In a typical project, one-third of the training data will not end up in any bootstrapped data. This set of data is called out-of-bag data ant it is used to cross-validation the performance of the model.

Selected hyperparameters of the RF are the number of trees, the maximum number of features to consider split at each node, the maximum depth of a tree, and the criteria to split a node of a tree. Increasing the number of trees in the forest would create a more robust model with less variance at the cost of a longer training time. This parameter should be selected large enough to represent the feature space effectively and be computationally feasible.

The number of features to be considered while splitting a node is depended on how much the data is noise-free. For clean data, a lesser number of features would be sufficient. If the data is noisy, this value must be set to higher values to increase high-quality features during the tree creation. Small values of the number of the features will decrease the variance of the model at the cost of higher individual tree bias. Higher values would decrease the bias of the model at the cost of a longer training time.

The depth of a decision tree determines how detailed the model represents the data. Therefore the larger value of the depth will result in overfitting.

Gini index and entropy are two options available for the criterion on how to split a node while using the Random Forest for classification. They are the measures for impurity of a node and calculated as follows:

$$Gini = 1 - \sum_{i=1}^{n} p^2(c_i), \tag{2.15a}$$

$$Entropy = 1 - \sum_{i=1}^{n} -p(c_i)log_2p(c_i), \tag{2.15b}$$

where $p(c_i)$ is probability of the class c_i in a node. The performance of both metrics may vary depending on the data.

2.4.3. Extreme Gradient Boosting

Gradient boosting is a machine learning algorithm for regression and classification, consisting of an ensemble of week prediction methods. Boosting is a method to improve the performance of machine learning algorithms by combining series of weak decisions. In the case of gradient boosting, the weak prediction method is a decision tree. Different from Random Forest gradient boosting uses the boosting algorithm rather than the majority led decision. Extreme Gradient Boosting (XGB) is an enhanced version of gradient boosting with additional steps and methods. In gradient boosting, regression trees are used to create a model even if the final goal is classification. These trees are built using residuals instead of the actual class of the samples. The residual represents the error between the predicted and actual values of the sample. The same strategy is applied for XGBoost. Trees are restricted with a certain value of depth. The maximum size of the trees prevents the overfitting of training data. Trees are combined sequentially step-by-step with a learning rate at each step. The learning rate regularizes the contribution of each step. Hence more gradually and steady improvement is achieved.

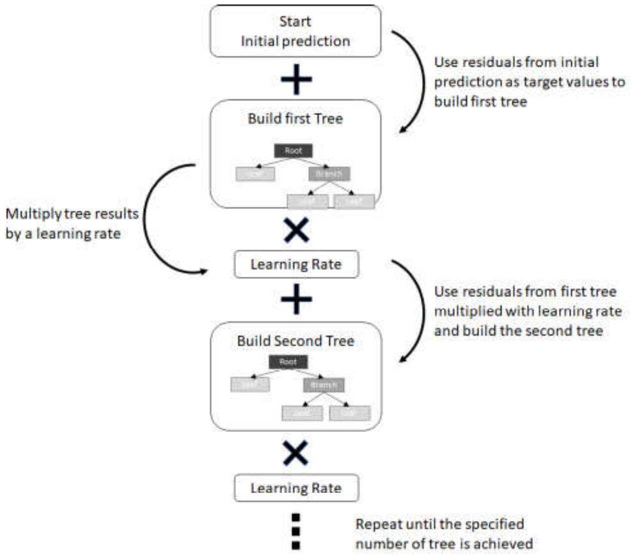
Start
Initial prediction
+
Use residuals from initial prediction as target values to build first tree
Build first Tree
Root
Branch
Multiply tree results by a learning rate
×
Learning Rate
+
Use residuals from first tree multiplied with learning rate and build the second tree
Build Second Tree
Root
Branch
×
Learning Rate
Repeat until the specified number of tree is achieved

Figure 2.4. XGBoost algorithm.

The visual interpretation of the algorithm is given in Figure 2.4. Gradient boosting algorithm uses Mean Square Error (MSE), which is a standard metric for representing the error of the prediction that can be calculated as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Observed_i - Predicted_i)^2. \tag{2.16}$$

While splitting the nodes, a split point is selected where the least MSE is obtained. On the contrary, XGB uses a different method to split the nodes. First similarity score given is calculated for each node according to the equation:

$$Similarity = \frac{\sum_{i=1}^{n}(R_i)^2}{\sum_{i=1}^{n}[Pp_i \times (1 - Pp_i)] + \lambda}, \tag{2.17}$$

where R_i is the residual which is the difference between actual and predicted values of the data, Pp_i is the previous probability which is the output of the previous step. While building the model, it starts with an initial guess. λ is the regularization parameter that classic gradient boosting does not offer. A gain score is calculated as

$$GAIN = Left\_Leaf_{similarity} + Right\_Leaf_{similarity} - Root_{similarity}, \tag{2.18}$$

for each leaf node. A split point is selected where the gain is highest. XGB provides a hyperparameter which is denoted as γ. The nodes which are gain score is less than γ is pruned. Hence the overfitting of the model is prevented.

2.4.4. Multilayer Perceptron

Perceptron is an atomic building block of an Artificial Neural Networks (ANN). ANNs are the computing model inspired by biological neural networks. In this analogy, a perceptron corresponds to a neuron in a neural network, an activating unit when the stimulus is higher than a threshold value. Neural networks consist of the connection of perceptrons. These connections are formed into layers. Multilayer Perceptron (MLP) consists of at least three layers; input, output, and hidden layers. More complex models can be achieved by increasing the number of hidden layers or the number of perceptrons in a hidden layer at the cost of a longer training time.

A perceptron stores a set of values that are called weights. It multiplies the weight with its inputs and generates an output. If the output is higher than a threshold value, the node is activated and generates output accordingly. This relation is represented as

$$f(x) = w_0 + Wx_{input}^T, \tag{2.19}$$

where x_{input} is the input data of the perceptron, W is the weights, w_0 is the bias value that is used to set the threshold. The resulting multiplication is fed to an activation function to obtain the on/off behavior of a perceptron. Thus using nonlinear activation functions, neural networks can be used to solve nontrivial problems. The most commonly used activation function is Rectified Linear Unit (ReLU) function. In this case, the Equation (2.19) becomes

$$f(x) = max(0, w_0 + Wx_{input}^T). \tag{2.20}$$

It is widely used because it is easy to compute compared to sigmoid or *tanh* functions. It is less likely to occur vanishing gradients; thus, it provides better gradient performance. On the other hand, being not differentiable at zero and being not zero-centered are the downsides of the ReLU. A different type of activation function is used for the output layers, which is called softmax. The softmax function gives a probability distribution over predicted classes.

Gradient descent algorithm is an iterative search method that tries to find a given loss function based on its gradient. It is used to train neural networks. The loss function is an evaluation of a candidate solution of the model. The cross-entropy function is the most commonly used method for the classification problem. For binary classification, the cross-entropy function is given as

$$C_e = \frac{1}{n}\sum_{i=1}^{n}(y_i log(P_i) + (1 - y_i) log(1 - P_i)), \tag{2.21}$$

where n is the size of the data, y_i is the actual label, and the P_i is the output of the softmax function for i^{th} sample.

Gradient descent updates the network weights based on the gradient of loss function multiplied with a learning rate. A randomly selected subset of the dataset is preferred to calculate the gradient. The subset is called a batch, and its size is a hyperparameter. This method is called Stochastic Gradient Descent (SGD). It eases the calculation burden of the training and achieves faster iterations. A commonly used version of SGD is Adam optimizer. It uses the first and second moments of the gradient to calculate individual learning rates for different parameters. Scanning all of the training datasets by the optimizer is called the epoch. Optimizer runs until the error is under a specified value or the specified number of epochs is achieved.

MLP could also face the overfitting problem like other machine learning algorithms. Dropout is a method to prevent overfitting. It achieves a more generalized model by disabling randomly selected nodes during training. The dropout rate is another hyperparameter to be set.

2.4.5. Cross Validation and Hyper-Parameter Tuning

The purpose of the Cross Validation (CV) is to see if the model can predict new data that was not used in the estimation and see if it would generalize to a different dataset. CV is the process of splitting a dataset into subsets that are used for training, validation, and testing. If the data is divided into a training set and a validation set, the models' performance metrics are highly dependent on those two sets. The performance depends on one evaluation and varies when trained and evaluated on different data subsets. That is why multiple subsets are picked. Thus, the model's final estimate of its predictive power results from accumulating all evaluations' results. Thus, the variability of the prediction performance of the trained model can be eliminated in this way.

Hyper-parameters are configurable parameters of a machine learning model to be specified for a task or dataset. Unlike other learned parameters, hyper-parameters can be manually set to direct and optimize the learning process for a given dataset. The optimization is started with defining a search space which is volume to search, and each dimension corresponds to a hyper-parameter; each point is a vector for a model configuration. Random Search (RS) is an algorithm that randomly selects points inside the bounded search space. RS are more efficient due to not all hyperparameters require equal tuning [33]. The models are trained with hyper-parameters selected randomly in search space and evaluate with a determined number of cross-validation folds. This concept is repeated for all randomly selected vectors in search space, and the final best-performing parameter combination is returned.

2.5. Dimensionality Reduction

The dimension reduction approaches are used for visualizing the results and datasets. t-Distributed Stochastic Neighbor Embedding (t-SNE) is a dimension reduction technique preferred to visualize our dataset.

In 2008, Laurens van der Maatens and Geoffrey Hinton developed it [34]. It is based on Stochastic Neighbor Embedding (SNE). SNE starts with calculating the probability that items s and l are similar, as follows. Firstly, the high-dimensional Euclidean distances are transferred into similarity conditional probabilities. It is calculated as

$$p_{j|i} = \frac{e^{\frac{-\left\|s_i - s_j\right\|^2}{2\sigma_i^2}}}{\sum_{k \neq i} e^{\frac{-\|s_i - s_k\|^2}{2\sigma_i^2}}}, \tag{2.22}$$

where N-dimensional observation s_i to s_N and bandwidth of the Gaussian kernel function σ_i, the similarity probability, $p_{j|i}$, of s_j and s_i. The similarity conditional probability $p_{j|i}$, which is the similarity of s_j to s_i, is high if they are close samples. Vice versa, it will be nearly infinitesimal for samples that are substantially separated. SNE uses the Gaussian distribution to determine the similarity of low-distinction samples [12]. The conditional probability $qj|i$ can be obtained as

$$q_{j|i} = \frac{e^{-\|l_i - l_j\|^2}}{\sum_{k \neq i} e^{-\|l_i - l_k\|^2}}, \tag{2.23}$$

where l_i and l_j are low-dimensional equivalents of high-dimensional points s_i and s_j.

Kullback-Leibler (KL) divergence between a joint probability distribution, P, in the high-dimensional space, Q and it is minimized with

$$KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{2.24}$$

Student t-distribution is utilized to measure similarity between locations of low dimension to let unrelated objects stand apart from each other on the map. The joint distribution q_{ij} can be written as

$$q_{ij} = \frac{(1 + \|l_i - l_j\|_2^2)^{-1}}{\sum_{k \neq l}(1 + \|l_k - l_l\|_2^2)^{-1}}, \tag{2.25}$$

while $(1 + \|l_i - l_j\|_2^2)^{-1}$ approaches an inverse square law for large pairwise distances $\|l_i - l_j\|$ in the low-dimensional space. The gradient of Equation (2.24) is

$$\frac{\delta KL}{\delta l_i} = 4 \sum_j (p_{ij} - qij)(l_i - l_j)(1 + \|l_i - l_j\|^2)^{-1}. \tag{2.26}$$

3. DATA COLLECTION

An automated test bench collects current and vibration signals of BLDC motors from the end of the production line. The related test bench is explained in Section 3.1, the encountered challenges during data collection are stated in Section 3.2.

3.1. Test Bench

The study aims to create a fully automated test system in the production line without affecting the production cycle time, which is 12 sec. That's why data are collected from the automated test bench installed at the end of the production line.

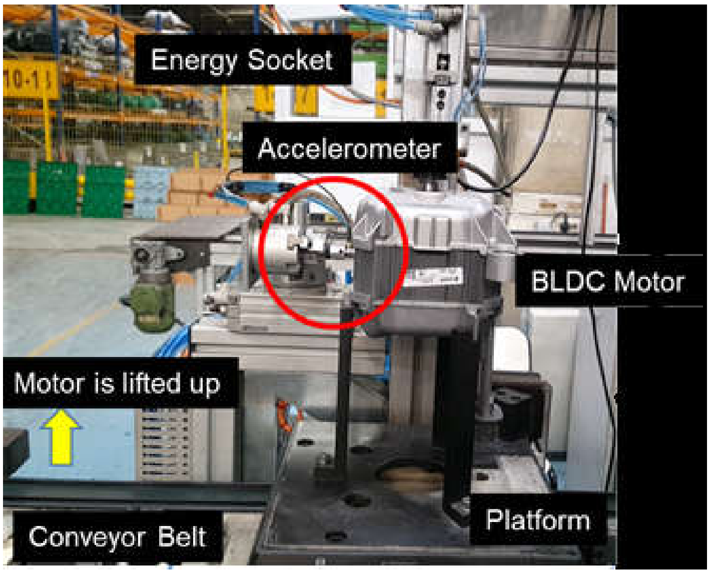

Figure 3.1. The data collection test bench.

The installed setup is given in Figure 3.1. The measurement and data collection process follows the procedures below.

(i) The testing procedure is triggered when the motor reaches above bench.
(ii) The barcode on the motor is registered to PLC.

(iii) The motor is isolated from the production line by a lifting mechanism during data collection to eliminate additional noises from the conveyor.

(iv) The male connector plugs into the female counterpart of the socket after lifting the engine. Thus, the motor becomes ready to be energized.

(v) Male connector is released to disconnect any interaction with its grabber.

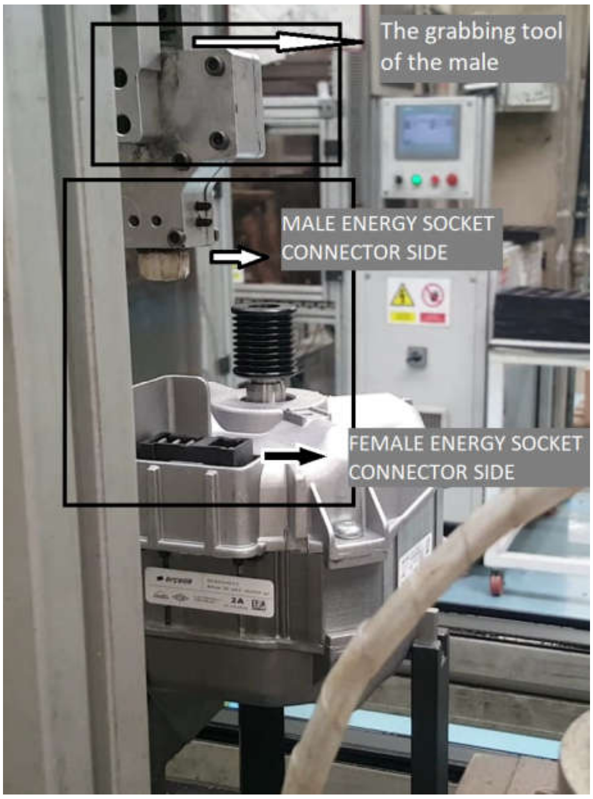
The grabbing tool of the male
MALE ENERGY SOCKET CONNECTOR SIDE
FEMALE ENERGY SOCKET CONNECTOR SIDE

Figure 3.2. Another view of energy socket connection: A grabbing mechanism holds the male side of the energy socket. The male connector plug in the female connector on the motor when the motor is lifted. Then, the grabber releases the male connector not to transfer on it.

Thus, the vibration transfer through the grabbing tool is prevented. Grabber and two counterparts of power socket are depicted in Figure 3.2.

(vi) The accelerometer is mounted on the motor.

(vii) Motor is started.

(viii) Data acquisition process is started. The signals of vibration and 3-phase-current are recorded for seven seconds.

(ix) The collected data is stored with the motor barcode as time series under a filename.

(x) Lifting mechanism goes down.

(xi) Motor is transferred to the conveyor belt.

All signals were collected with a sampling frequency of 50 kHz. ADCR 100 series data acquisition card and its user interface iProbe Platform[2] was used to collect data. ADCR is hardware developed by Ascenix Corporation to provide high-speed and continuous collection and transfer of current, voltage, and vibration data to the processing point.

The PCB 352C34 ceramic shear ICP® accelerometer [35] is installed. It is mounted onto the motor's outer body perpendicular to the rotation direction of the engine. Magnetic mounting bases are used to attach the accelerometer to the motor's surfaces. The magnetic bases are screwed to the accelerometer with a thin layer of silicone between the base and the sensor.

3.2. The Challenges

Various mechanical issues arise during data collection. The encountered difficulties are listed below.

- The accelerometer should be mounted parallel to the main sensitivity axis, perpendicular to the rotation direction of the engine. A gap between the sensor and the motor can occur due to improperly mounting. During the data collection process, this issue arose frequently. Improper mounting occurs when the 3-finger mechanism that mounts the sensor on the motor fails to release the sensor, and hence the sensor does not attach to the engine with full contact. Another reason for this issue is the differences in motor holding mechanisms. For example, if the height of each leg of the mechanism that holds the motor is not identical or if the fastening screws of any leg are loose, in this case, the targeted sensor mounting

[2] Ascenix Corporation, Washington DC, USA

point may slide. The accelerometer can not be adequately mounted on the motor. These differences cause the accelerometer to attach with an angle or not to stick on the engine at all.

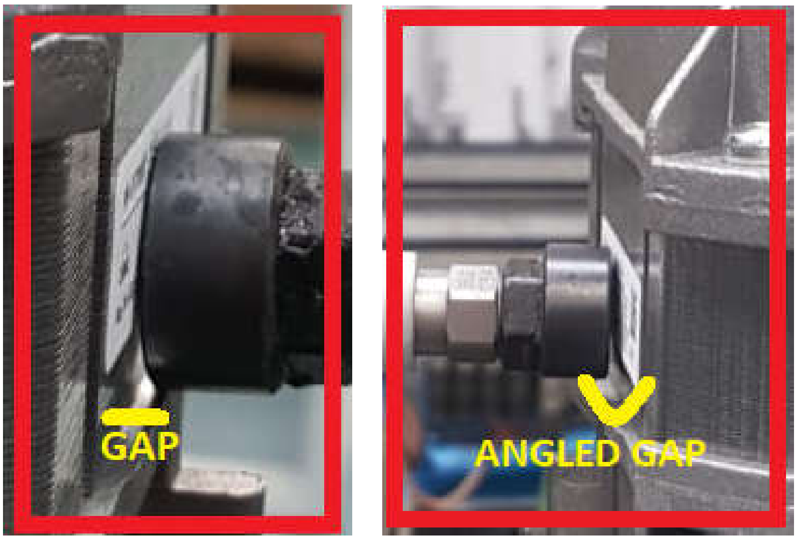

Figure 3.3. Two examples of the sensor mounting problems: The angles that occur as a result of trying to mount the sensor on the protruding surface instead of the flat surface of the motor (left). The angle is formed because it cannot cut its connection with the three-finger mechanism holding the sensor (right).

Most common examples about sensor mounting problem are given in Figure 3.3. These gaps are caused the distortion on the vibration signal.

- The mechanism that separates the motor from the production conveyor belt may drop the engine during its upward movement. In this case, the data collection is interrupted.
- Energizing problems can happen during data collection due to the power socket not being fully inserted into the motor's connector.
- Differences in the height of each leg of the motor holding mechanism and the width of the notch in which the motor is placed would affect the vibration signals as it will change the swing amount of the engine while it is running.

The measurements that have these issues can be removed from the dataset as to be discussed in Section 4.1. However, the problems about improper mounting of the accelerometer on the motor's body is still an issue after preprocessing.

4. PREPROCESSING

Data preprocessing involves noise suppression, data exclusion, consistency check, and reformatting to maximize machine learning models' performance. It is a crucial step before extracting features from all raw recorded signals.

The details of the elimination of erroneous recordings, temporal windowing, and noise filtering are given in Section 4.1. Temporal and spectral features are mentioned in Section 4.2. Finally, the datasets are introduced in Section 4.3.

4.1. Data Exclusion and Steady State Frame Detection

The data exclusion is a removal step of improperly recorded cases from the dataset, which may well be caused by not energizing the motor due to improper placement of the energy socket originating from the mechanical parts and not contacting the accelerometer to the motor body. Short-Time Fourier Transform (STFT) is applied to observe the time-dependent variations of engine operating status, ramp-up, ramp-down and steady-state.

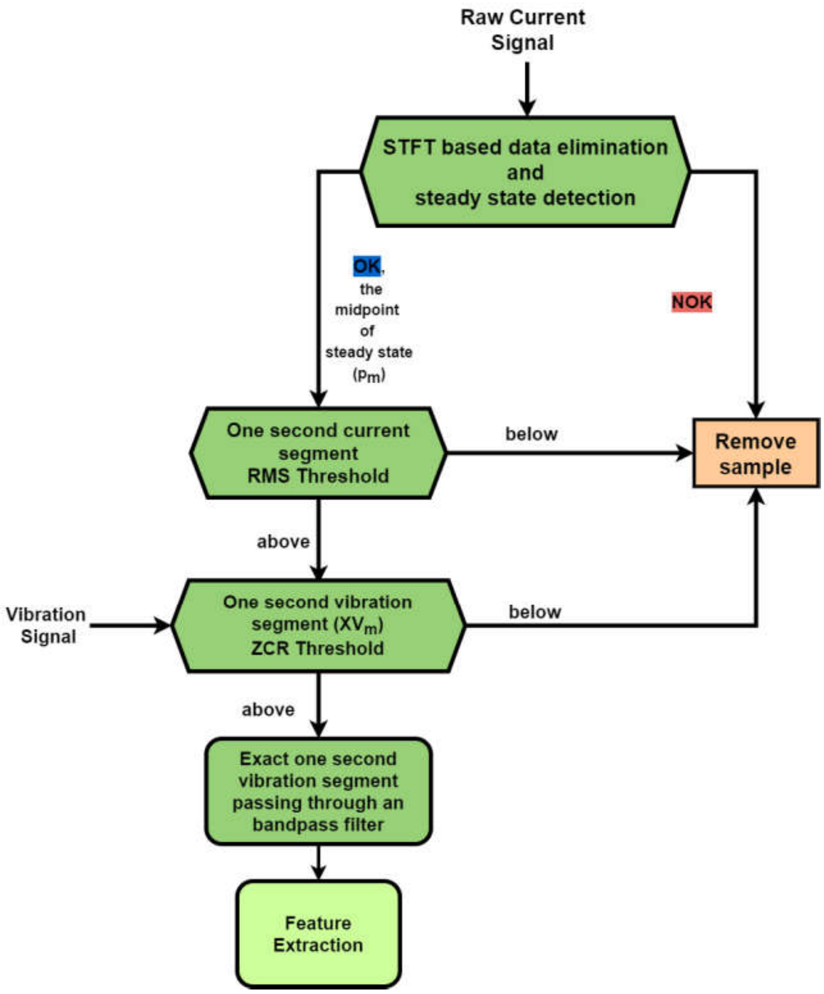
Raw Current Signal
STFT based data elimination and steady state detection
OK, the midpoint of steady state (Pm)
NOK
One second current segment RMS Threshold
below
Remove sample
above
Vibration Signal
One second vibration segment (XVm) ZCR Threshold
below
above
Exact one second vibration segment passing through an bandpass filter
Feature Extraction

Figure 4.1. The preprocessing steps: Raw current signals are used to detect the transition points where the motor transitions from the ramp-up to steady-state by applying STFT based algorithm. Then, the RMS of the one-second current signal is computed and compared with a threshold. The calculated ZCR of the one-second vibration signal is compared with the threshold. Finally, the feature extraction process is started by using the vibration data of the samples that have passed all stages.

The whole preprocessing steps are listed in Figure 4.1. The transition points where the motor switches from ramp-up to steady-state are detected using an STFT-based technique on raw current measurements. The RMS value of the one-second current signal is computed and compared to a threshold. If the calculated value is lower than the threshold, all signals of the motors are removed from the dataset. Another threshold is based on the estimated Zero-Crossing Rate (ZCR) of the one-second vibration signal. The signals that can manage to pass all preprocessing and removal stages are used for the feature extraction process.

4.1.1. STFT Based Steady State Detection and Data Removal

One of 3 phase current signals is used in this stage. Although each motor's ramp up, ramp down, and steady-state timings should be the same, they are not aligned perfectly across all cases due to several delays. Hence, individual ramp-up, ramp-down and steady-state periods need to be determined, and the sample that can not reach steady-state or is not energized should be removed from datasets. All features are extracted from the steady-state phase, where the motor runs at a constant speed.

The BLDC motors are driven with Mitsubishi FR-A820-00046-E1N6 model inverter from FR-A800-E series [36]. Their output frequency ranges from 0.2 to 590 Hz. In our study, the generated current signal by the inverter should be at least 585 Hz. Thus, the motor can run at 8800 RPM.

The frequency of the provided current is increased over time until the motor reaches the targeted speed. Thus, STFT analysis is chosen to find time-varying frequency information. The transition points between motors' operating conditions over the current signal are found using STFT. For computing STFTs, a longer time signal should be divided into equal-length segments. Each segment is multiplying with a window function. As the window is moved along the time axis, the Fourier transform of the resulting signal is obtained, as explained in Figure A.1. As a result, the signal is represented in two dimensions. Then, the maximum amplitude and its corresponding frequency are found for each STFT frame. The detected frequencies are stored in an array. Then, the first and last points of the array greater than or equal to 585 are determined.

```
input  : Current Signal: $\chi_c = \{x_{c1}, x_{c2}, \ldots, x_{cn}\}$,
         Window length: $wlen$, Rectangle Window: $Win$
         The number of samples between each successive FFT window: $hop$,
         The number of FFT points: $nfft$, Sampling Rate: $fs$
output: The mid-point of steady state: $p_m$, the status of measurement: $status$
begin
    compute $STFT$ as matrix(using Algorithm given Figure A.1)
    take the number of rows $Le$ of $STFT$
    for $l \leftarrow 0$ to $Le$ do
        find maximum peak and its index $i$ of $STFT_{*,l}$, set $MaxIndex(l) = i$
    end
    find the first $i_f$ and the last point $i_l$ where $MaxIndex >= 585$
    if $(i_f = 0)$ or $(i_l = 0)$ then
        $status = improper$ and $p_m = 0$ break
    end
    $SteadyFrame = MaxIndex_{i_f, i_l}$        $FrameIndex = \{r_1, r_2, \ldots, r_m\}$
    compute difference $Rdiff = [r(2) - r(1)r(3) - r(2) \ldots r(m) - r(m-1)]$
    if $\sum_{k=1}^{m} Rdiff_k$ is not equal to 0 then
        $status = improper$ and $p_m = 0$ break
    end
    if $i_l - i_f <= fs/hop$ then
        $status = improper$ and $p_m = 0$ break
    end
    $status = proper$ and $p_m = \frac{i_l + i_f}{2}$
end
```

Figure 4.2. Steady-state detection and data removal algorithm. The transition points between motors' operating conditions over the current signal are found using STFT. The maximum amplitude and its corresponding frequency are found in each STFT frame. The first and last values of the array that are greater than or equal to 585 are detected. Then, the difference between successive array elements is calculated.

The given algorithm in Figure 4.2 provides the details about the algorithm to remove improper recording and detect steady-state frame. If any detected points are 0, the engine can not run properly, and all recordings of this instance should be removed from the dataset. After then, the part of the array between the first and last points is taken. The difference between successive array elements of the part is calculated. If the motor runs with constant speed, the sum of the calculated difference should be 0. Otherwise, the sample should be excluded from the datasets. Three things are checked. First and foremost, it determines whether the motor is running, if it has run, if it can reach the target revolution per minute (RPM), and if it is capable of working at that RPM for at least one second. It is removed from the dataset as an improper measurement if the aforementioned initial conditions are not met. The 1000 msec time window at the center of the thus detected steady-state period is marked for feature extraction.

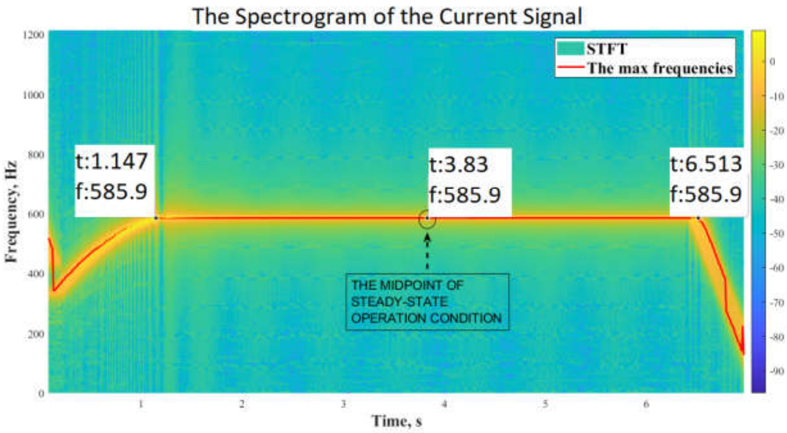

Figure 4.3. The STFT of the current signal, which is belonged to the motor running properly, and the highest amplitude frequency array, which is obtained by detecting max amplitude and corresponding frequency for each STFT segment.

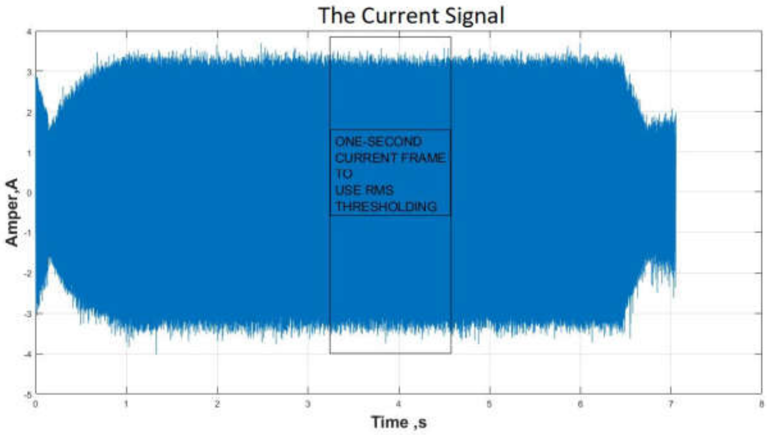

Figure 4.4. The current signal in time domain and the detected steady-state frame.

The exact midpoint is calculated by using the seconds when the motor starts and ends with constant RPM, as illustrated in the below figure and stored. A one-second frame from the current, as given in Figure 4.4 and vibration signal are taken around the detected mid-point, as described in Figure 4.2. The correctly collected signals and their STFT visualizations are given in Figure 4.3.

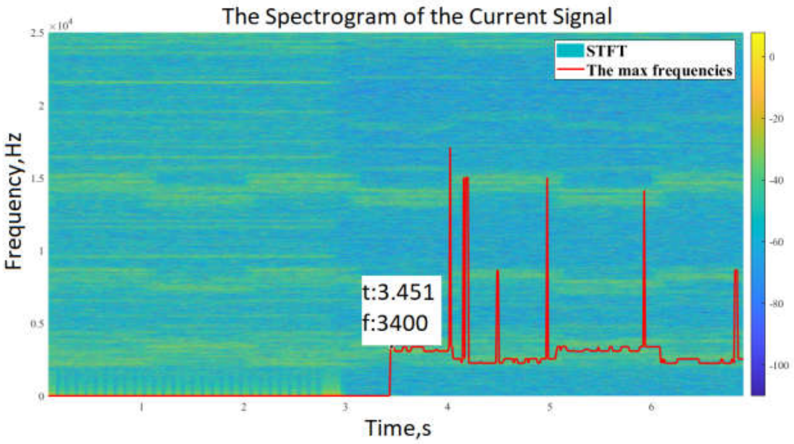

Figure 4.5. The STFT illustration of the current signal of the improperly run engine.

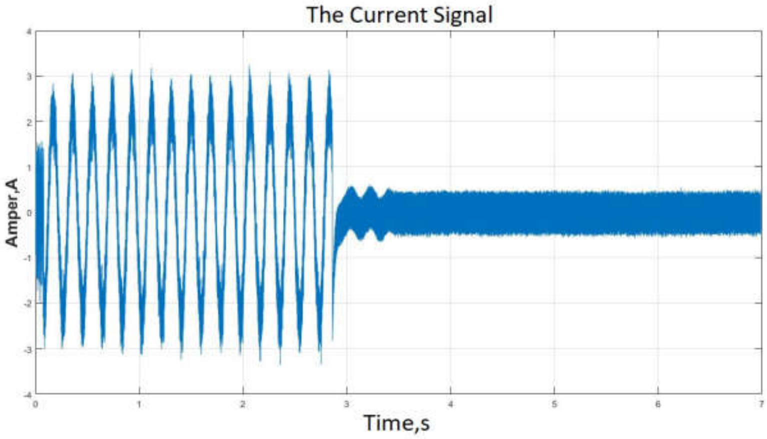

Figure 4.6. The visualization of the current signal in case the power socket can not plug into the motors' energy socket.

The current signal samples, in which the motor can not run properly, are given in Figure 4.6. Figure 4.5 shows the STFT illustration of the current signal if the power socket can not plug into the motors' energy socket.

4.1.2. RMS & ZCR Thresholding

The minimum desired current RMS value under steady-state conditions is determined by the R&D department of Arcelik as 1.7 A. All recordings with current RMS less than 1.7 A are excluded from the dataset. Moreover, the three-finger mechanism that attaches the accelerometer to the motor is another cause of improper sensor mounting. All fingers must open after the accelerometer has been mounted on the engine to isolate itself from the accelerometer. The accelerometer will not fully contact the motor if all three fingers do not open, resulting in an improper measurement.

The ZCR of vibration signals are used to determine improper signal recordings. The ZCR is calculated over all vibration signals that can pass from STFT based data removal steps as mentioned in Section 4.1.1. The minimum ZCR threshold is applied on one-second steady-state vibration signals. The lower limit is calculated by subtracting the standard deviation 2σ from the mean of the ZCR across the dataset and it is 0.08.

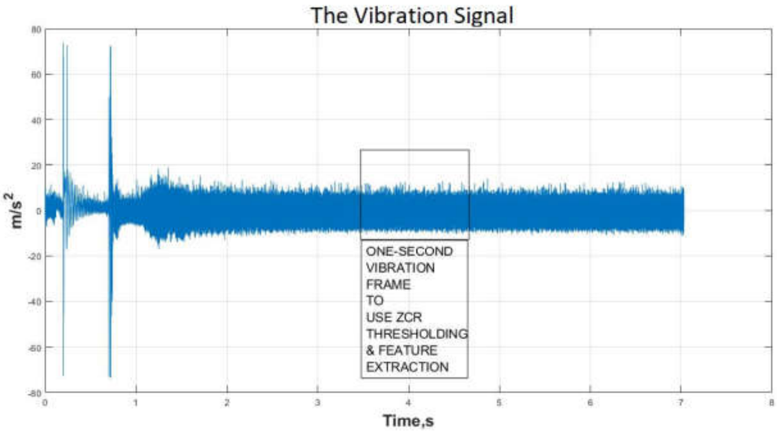

Figure 4.7. The properly collected vibration signal.

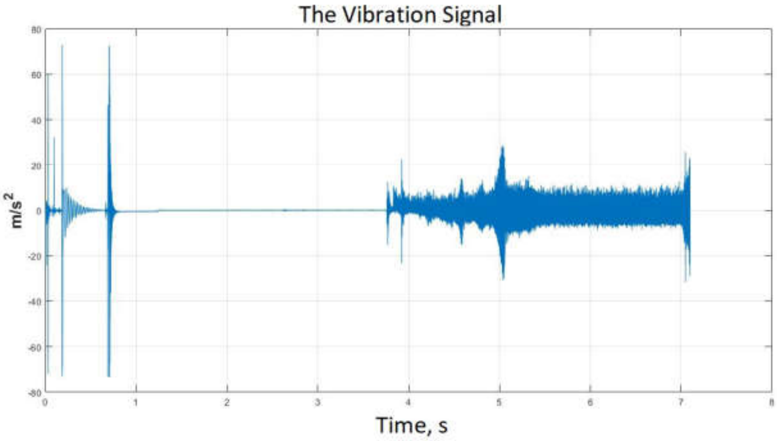

Figure 4.8. The improperly collected vibration signal in time domain.

The properly collected vibration signal, as shown in Figure 4.7, and improperly collected vibration signal are given in Figure 4.8. In this case, the accelerometer moves and drops after data acquisition starts.

4.1.3. Removing DC Component & BandPass Filtering

The mean of the vibration signal, which is the Direct Current (DC) component, is subtracted from the signal. Then, a bandpass filter is used to remove the spikes on the vibration signal and minimize the effects of the additional mechanical vibrations from the production process on the signal and DC components. Finally, a bandpass filter is used to remove the spikes on the vibration signal, minimize the effects of the additional mechanical vibrations. The low cutoff of the filter was 40 Hz since the minimum frequency of interest was line frequency 50 Hz. Because the accelerometer's frequency range is limited to 10 kHz, it is employed as the filter's high cutoff. Finally the bandpass filter works between 40 Hz to 10kHz frequency range.

4.2. Feature Extraction

The one-second vibration signals, where the motor is running at a steady-state condition, are used during the feature extraction process. The one-second frame is determined as discussed in Section 4.1.

Table 4.1. Characteristics of bearing in usage.

Rolling element per row N_r	8
Pitch diameter P_d	25.26 mm
Roller diameter B_d	6 mm
Contact angle phi	0,2617 rad

Twelve temporal features mentioned in Section 2.2.1 are extracted from the determined one second vibration recordings and also the critical frequencies are calculated by using Equations (2.13a), (2.13b), (2.13c) and (2.13d) by taking the motor rotation frequency as 146.7 Hz. The calculated characteristic frequencies of the faults on bearing are given in Table 4.1. The related frequencies about bearing faults are computed. BPFI is 721.27 Hz, BPFO is 452.06 Hz, FTF is 56.51 Hz, BPF is 584.96 Hz. The unbalance and misalignment faults are also computed.

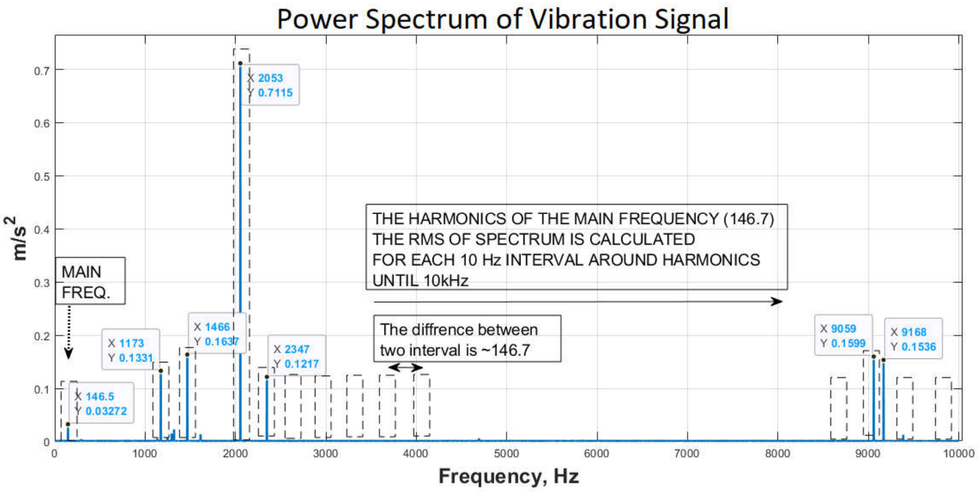

Figure 4.9. The spectral frequencies of main frequency f_{shaft}, 146.7 Hz. The frequency ranges are calculated as 10 Hz intervals around the mentioned particular frequencies and all harmonics.

Some frequency bands, which are used already in the current semi-automatic quality control system, are determined by the Arcelik R&D department, and other frequency ranges are calculated as 10 Hz intervals around above mentioned particular frequencies and all harmonics of main frequency f_{shaft}, 146.7 Hz, as depicts in Figure 4.9.

After determination of to be calculated frequency ranges, power spectrum density $P(f)$ of the vibration signal is computed with pwelch method [37] as following. Signal is $\chi = \{x_1, x_2, \ldots, x_n\}$ divided into segments, $\chi_k(j)$ in which $j = 0, \ldots, L-1$ and $k = 1, \ldots, K$. This segments are possibly overlapping, and the length of segments is L. They slips D units until the end of the signal, suppose K segments that include the all signal. The periodograms of each segment can be obtained as [37]

$$A_k(n) = \frac{1}{L}\sum_{j=0}^{L-1} \chi_k(j) \times W(j)e^{-kj2\pi\frac{n}{L}}, \qquad k = 1, 2, \ldots, K, \tag{4.1}$$

where $\chi_k(j)$ is the kth segments and jth frame of the signal and $W(j)$ is the selected window. K periodograms [37] are obtained as

$$I_k(f_n) = \frac{L}{U}|A_k(n)|^2, \qquad k = 1, 2, \ldots, K, \tag{4.2}$$

where [37]

$$f_n = \frac{n}{L}, \qquad n = 0, 1, \ldots, \frac{L}{2}, \tag{4.3}$$

and

$$U = \frac{1}{L}\sum_{j=0}^{L-1} W^2(j). \tag{4.4}$$

The power spectra is calculated as [37]

$$P(f_n) = \frac{1}{K}\sum_{k=1}^{K} I_k(f_n). \tag{4.5}$$

Then, the RMS of the periodogram of the signal in determined frequency ranges is computed as

$$V_{RMS} = \sqrt{\frac{1}{r}\sum_{r=f_{low}}^{f_{high}} P(r)}. \tag{4.6}$$

The current parameters, which are used for the semi-automatic system as aforementioned in Section 1.1, and the 10 Hertz intervals around particular frequencies are computed. They are explained in detail below:

- *First*: The RMS value of the power spectrum between 50 and 200 Hz exceeds the determined threshold. This frequency range contains the main frequency (146.67 Hz), dependent on the motor's rotation speed (8800 RPM).
- *Second*: The second defect checker is the RMS value of the power spectrum between 50 and 1000 Hz.
- *Third*: The third criterion is the computed RMS value of the power spectrum between 50 and 1575 Hz.
- *Fourth*: Computing the logarithmic sum of the power spectrum gives the fourth parameter.

All harmonics of the main frequency up to 10 kHz are calculated. A total of 76 frequency bands are used as spectral features.

Table 4.2. The computed frequency ranges.

Abbreviation	**Detail**	**Low Frequency - High Frequency**
First	First Band	50 Hz - 200 Hz
Second	Second Band	50 Hz - 1000 Hz
Third	Third Band	50 Hz - 1575 Hz
Fourth	Total Band	50 Hz- 10000 Hz
BPFI	Inner Raceway	716.27 Hz - 726.27 Hz
BPFO	Outer Raceway	447.06 Hz - 457.06 Hz
FTF	Cage Defect	51.51 Hz - 61.51 Hz
BPF	Ball Defect	579.96 Hz - 589.96 Hz
Unb	Unbalance Defect	141.67 Hz - 151.67 Hz
Mis	Rotor Misalignment	288.33 Hz - 388.33 Hz
H	Remaining Harmonics	146.67*k - 5 Hz k=3,4,. . . ,68 - 146.67*k + 5 Hz k=3,4,. . . ,68

The number of used frequency ranges are given in Table 4.2. First, Second, Third, and Fourth frequency ranges are coming from the standard quality control procedure applied at the manufacturing plant and others are commonly used frequencies to detect failures in literature.

4.3. BLDC Motor Datasets

Vibration and current signals are captured by using the automated data collection system as meantioned in Section 3. After all, recordings where the motor failed to start are excluded. The samples in which the recordings are acquired under improper conditions where the vibration sensor has been unable to attach to the body of the motor properly can not be excluded by using the preprocessing stage in Section 4.1.

Two datasets are created. While the first one has recordings from 750, the second dataset has recordings from 32120 motors. The first set, 750 recordings, were manually labeled as proper and improper with checking how the accelerometer mounts to the engine's body. This subset is called Quality Labelled Dataset (QLD), while the other is called Quality Unlabelled Dataset (QULD).

Apart from the binary labeling of the recording quality in the QLD, both Quality Labelled and Quality Unlabelled recordings are labeled following the standard quality control procedure applied at the manufacturing plant. These labels are Healthy/Pass, Vib1, Vib2, Vib3, Vib1&2, Vib, Sound. Except for Pass labeled instances, all other failure types are labeled as Fail during binary classification. They are explained in detail below:

- *Vib1*: First computed spectral feature, 50 and 200 Hz, exceeds the determined threshold. This frequency range detects the unbalance and cage failures.

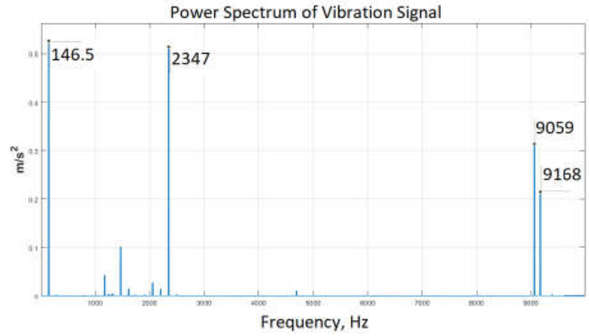

Figure 4.10. The power spectrum of a Vib1 motor.

The power spectrum of the vibration signal of the motor labeled as Vib1 is shown in Figure 4.10.

- *Vib2*: The second calculated spectral feature between 50 and 1000 Hz exceeds the second criterion. In this range, we can see the typical fault frequencies for bearings and rotor misalignment.

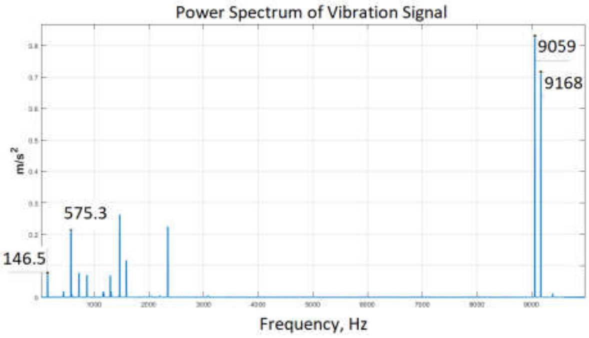

Figure 4.11. The power spectrum of a Vib2 motor.

Figure 4.11 shows the power spectrum of the Vib2 labeled motor.

- *Vib3*: The third obtained spectral feature, 50 and 1575 Hz is higher than the threshold. Finally, the 8th harmonic of the central frequency is also utilized to discover rotor defects because the motor's rotor contains eight poles.

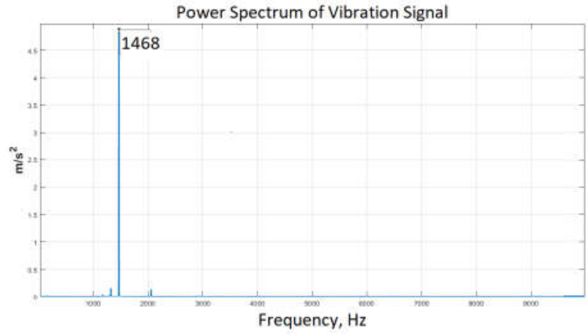

Figure 4.12. The power spectrum of a Vib3 motor.

The power spectrum of the Vib3 labeled engine is given in Figure 4.12.

- *Vib1&2*: The first two calculated values are above the threshold values.

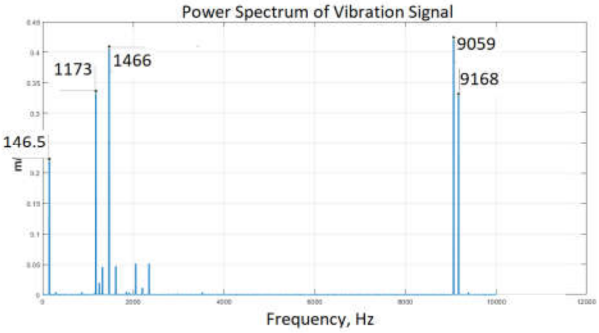

Figure 4.13. The power spectrum of a Vib1&2 motor.

The power spectrum of the vibration signal of the Vib1&2 motor is given in Figure 4.13.

- *Vib*: If the threshold values for the fourth and fifth criterion are surpassed, the operator gives a label to the motor as Vib. Counting the number of peaks in the power spectrum yields the fourth parameter and computing the fourth spectral feature. These numbers are utilized as an indicator of unusual motor noises. However, these two parameters are unable to detect all sound-related failures. Finally, while the motor is operating, an experienced operator listens to the engine's sound and determines if it is pass or fails.

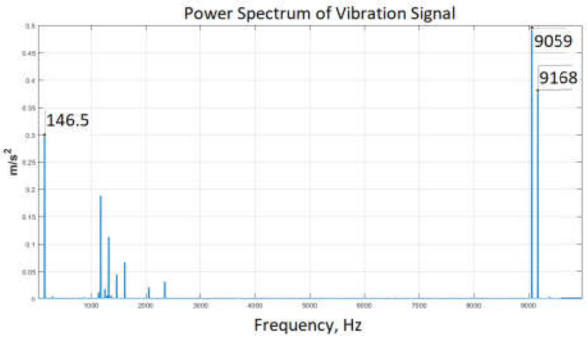

Figure 4.14. The power spectrum of a Vib motor.

Figure 4.14 shows the power spectrum of the vibration signal labeled as Vib.

- *Sound*: Some faults are not detectable by the vibration control system, and as a result, the operator must make the final choice. The operator must listen to the motor noise before making the final pass/fail judgment.

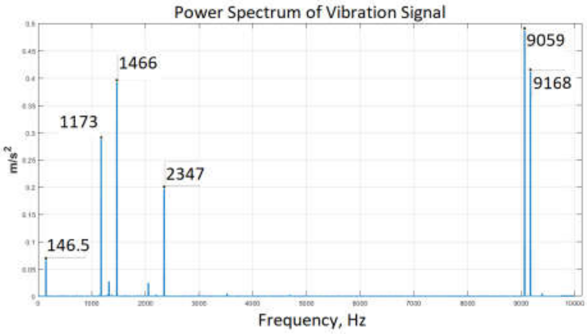

Figure 4.15. The power spectrum of a Sound motor.

The power spectrum of this fault is shown in given in Figure 4.15. This failure label is applied to samples that have been identified as failures by only the operator as they transit through the measuring system.

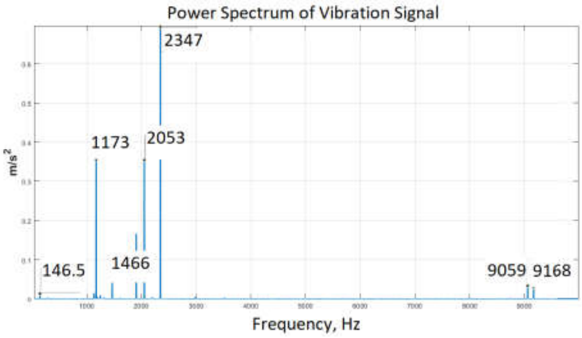

Figure 4.16. Figure illustrates the power spectrum of a healthy motor.

Figure 4.16 depicts the power spectrum of the vibration signal of a healthy motor. The main harmonics can reach a limited amplitude, and high-frequency noises almost disappear.

4.3.1. Quality Labelled Dataset

Vibration and current signals are collected from 750 motors. The samples with energizing issues are removed by applying the proposed preprocessing stage. After then, only 671 acquired signals remain. However, the vibration sensor improper placement problem is still present in the gathered signals. Therefore, the data is separated into two groups depending on whether the accelerometer could be properly placed on the motor. Properly collected data, i.e., proper sensor attachment, is called proper, improper recordings where the sensor was misplaced are called improper. QLD has more than one label. One of them is related to the engine's health status, pass or fail, and another one is related to the mounting of the accelerometer properly during data gathering, proper or improper.

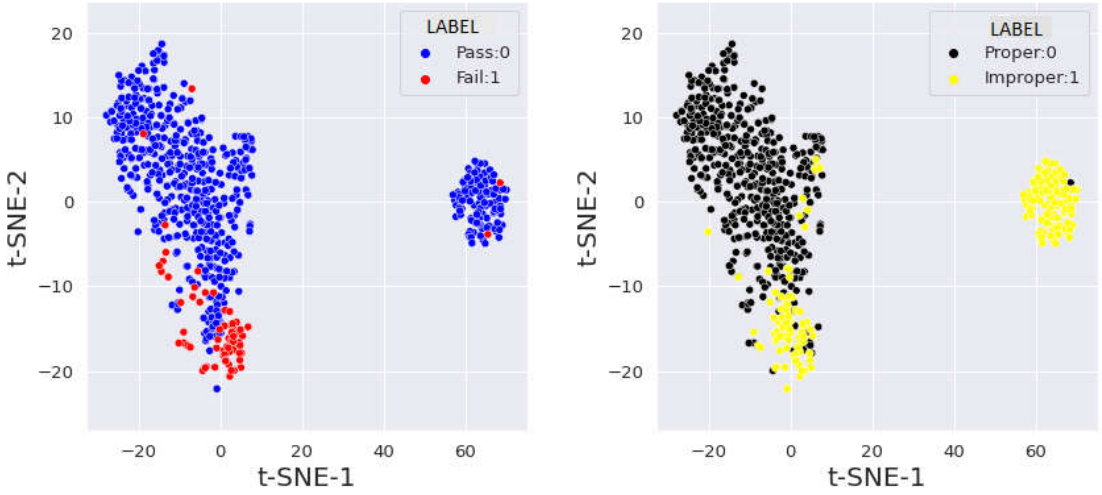

Figure 4.17. t-SNE visualizations of QLD: The distribution of QLD according to the engine's health status (left). The distribution of QLD in the aspect of data quality (right).

Figure 4.17 shows the distribution of QLD according to health status labels and data quality labels.

Table 4.3. Quality Labeled Dataset(QLD) descriptions

Health Status	**Mounting Status of Accelerometer**	**Label**	**Label Detail**	**Number of Samples**
Healthy	Proper	Pass	-	438
Faulty	Proper	Fail	Vib1	0
			Vib2	3
			Vib3	0
			Vib1&2	19
			Vib	1
			Sound	7
Healthy	Imroper	Pass	-	162
Faulty	Improper	Fail	Vib1	0
			Vib2	4
			Vib3	0
			Vib1&2	28
			Vib	1
			Sound	8

Table 4.3 depicts the number of samples according to data quality and motor's health status. Properly collected data, i.e., with proper sensor attachment, is called proper. Improper recordings where the sensor was misplaced are called improper. Binary motor health status is labeled as pass, healthy motor, and fail unhealthy or faulty motor.

4.3.2. Quality Unlabelled Dataset

Vibration and current signals from 32120 motors are collected for a month. In this dataset, there is no label about the improper conditions where the vibration sensor cannot mount to the engine properly. Only 28194 engines were able to be run during data collection, and only 25580 signals remain after our preprocessing stage, as explained in Figure 4.1 and Section 4.1. Data exclusion criteria do not include sensor misplacement; hence the 25580 remaining recordings can potentially have improperly recorded signals, but they are not labeled in QULD.

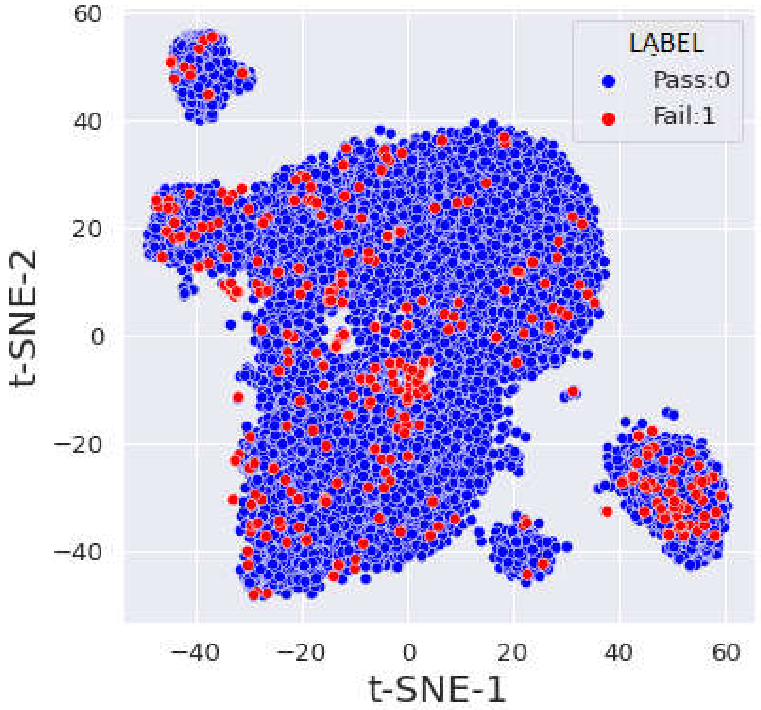

Figure 4.18. t-SNE visualizations of QULD according to the engine's health status.

Figure 4.18 shows the distribution of QULD according to its health status label. Healthy motors are labeled as pass, while all other types of failure are labeled fail. The pass label is assigned to the negative class and represented with red dots, while the fail label is assigned to the positive class and represented with blue color.

Table 4.4. Quality Unlabeled Dataset (QULD) descriptions. Binary motor health status is labeled as Pass, healthy motor, and Fail, unhealthy or faulty motor.

Health Status	**Label**	**Label Detail**	**Number of Samples**
Healthy	Pass	-	25180
Faulty	Fail	Vib1	65
		Vib2	58
		Vib3	5
		Vib1&2	83
		Vib	110
		Sound	79

The fault types and their distribution are given in Table 4.4. The vast majority of samples have a pass label, and other classes are under-represented. The imbalance ratio of the QULD is 1:63. It is highly imbalanced in terms of the motor health labels, which poses an extra challenge for the pursued machine learning approaches.

5. AI-POWERED QUALITY CONTROL (AIQC)

Two AI-assisted BLDC motor quality control methods are described. The first method is Single-Stage AI-Powered Quality Control (sAIQC) which only tries to classify data as healthy (pass) or faulty (fail), blind to data acquisition quality, which is related to how the accelerometer is mounted on the motor. The second proposed method is Double-Stage AI-Powered Quality Control (dAIQC) which is composed of 2 stages. The first stage makes a binary classification based on data quality as proper or improper, and the second stage is a binary classification as pass or fail. For both cases, multi-classification is not preferred due to problems with data quality and the limited number of samples in some failure types. Instead, binary classification is favored.

The preprocessing steps are applied to all datasets to exclude erroneous recordings, suppress external noises, and detect the transition points of steady-state operation, as summarized in Figure 4.1. The steady-state phase, in which the motor runs at a constant speed, provides all of the features. And, as explained in Section 4.2, temporal and spectral features are extracted for both dataset, QLD and QULD, as mentioned in Section 4.3. All of the above stages are employed as standard processes in both sAIQC and dAIQC.

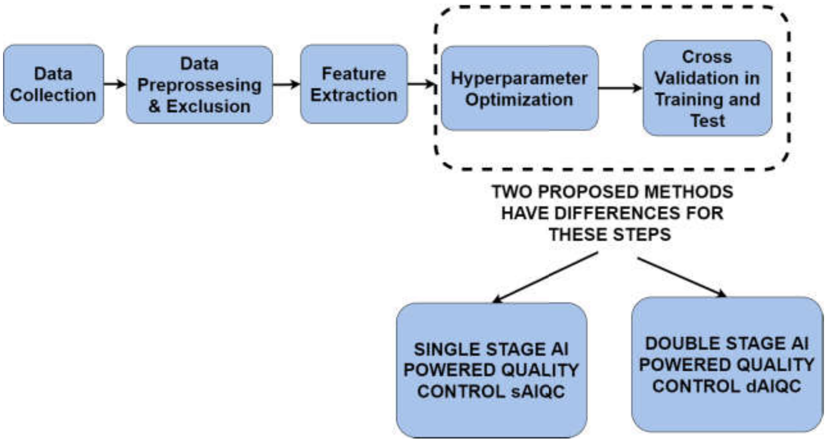
Data Collection
Data Preprossesing & Exclusion
Feature Extraction
Hyperparameter Optimization
Cross Validation in Training and Test
TWO PROPOSED METHODS HAVE DIFFERENCES FOR THESE STEPS
SINGLE STAGE AI POWERED QUALITY CONTROL sAIQC
DOUBLE STAGE AI POWERED QUALITY CONTROL dAIQC

Figure 5.1. The schema of the following steps of sAIQC and dAIQC.

Each method uses hyper-parameter optimization and cross-validation for the evaluation process. Figure 5.1 shows the schema of applied steps. Data collection, preprocessing, feature extraction are common stages of both sAIQC and dAIQC. Hyperparameter optimization and cross-validation are also used in both methods. But according to the structure, how many times these blocks should be used may vary, and labels can change.

Hyperparameter optimization of each classifier is another essential step in both sAIQC and dAIQC before the actual training process begins. To choose the right combination of hyper-parameters based on the performance of the feature set, the random search method is used as described in Section 2.4.5. The random search algorithm picks a random combination and evaluates the model using 3-fold cross-validation and getting the best hyper-parameters. Balanced accuracy is used as the scoring parameter of this process. After performing optimization, the top-performing parameters of the classifier are fixed.

The efficiency of the suggested models has been demonstrated using three different ML models, which are support vector machine (SVM), random forest (RF), extreme gradient boosting (XGB), and multilayer perceptron (MLP). The details of these classifiers explained in Section 2.4. Radial Basis Function (RBF) is used as a kernel function of SVM, and C and γ parameters are optimized. The hyper-parameters of RF are the number of trees in the forest, *n_estimators*, the maximum depth of the tree, *max_depth*, the criterion for splitting attributes, *criterion*. Learning rate, η, the maximum depth of a tree, *max_depth*, the number of trees, *n_estimators* and subsample percentages, *subsample* parameters of XGB are optimized. Finally, the size of each hidden layer, the drop-out rates, batch size, the number of epochs, and the learning rate of the optimizer are selected for hyper-parameter optimization of MLP. MLP consists of three hidden layers. Binary cross-entropy is used for the loss function, Adam optimizer is preferred for compiling the model.

Cross-validation is used for performance assessment of the classifiers with optimum hyper-parameters in both sAIQC and dAIQC. Stratified k-fold cross-validation is prefered [38]. The dataset is split into three folds. The class ratio and the distribution of defect types are preserved in each fold. Two sets of folds are used for training, whereas one set is used for the test. In this case, the entire dataset becomes test data in any cross-validation fold. The fixed hyper-parameters are used for each training CV fold, with model parameter initialization for each fold. All predicted test results are stored, and after the cross-validation process is completed, the final evaluations of the classifier are made on this stored data.

While training, class imbalance is always taken into consideration. The class weights are distributed inversely proportional to the number of samples in classes. In other words, unweighted class weights are used for each classifier.

Finally, True Positive Rate (TPR or Sensitivity), True Negative Rate (TNR or Specificity), and Unweighted Accuracy (UA) are used as performance metrics of the classifier. They are chosen to deal with the imbalanced class ratio. It should be known which label represents the positive and negative class to calculate performance indicators, which are True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN). In pass and fail classification, the pass class is considered the negative class and labeled as 0. The fail class is set as the positive class and labeled as 1. In proper and improper recording classification, proper recordings are labeled as 0, which is the negative class, and improper recordings are labeled as 1, which is the positive class. TPR can be obtained as

$$\textit{TPR - Sensitivity} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{5.1}$$

TNR is calculated as

$$\textit{TNR - Specificity} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}. \tag{5.2}$$

Finally, UA is the average of the recall scores for each class. It can be computed as

$$\textit{Unweighted Accuracy} = \frac{\mathrm{TPR} + \mathrm{TNR}}{2}. \tag{5.3}$$

5.1. Single-Stage AIQC

5.1.1. Method

The proposed sAIQC method aims to separate the samples as pass or fail regardless of data quality for both QLD and QULD. To compare the performance of sAIQC's classifiers and discuss the effects of data quality on the classifier's performance, the classifiers are also trained with only pre-labeled proper and improper labeled instances of QLD separately. Due to the lack of data quality labels in QULD, only QLD are used to compare the performance of classifiers.

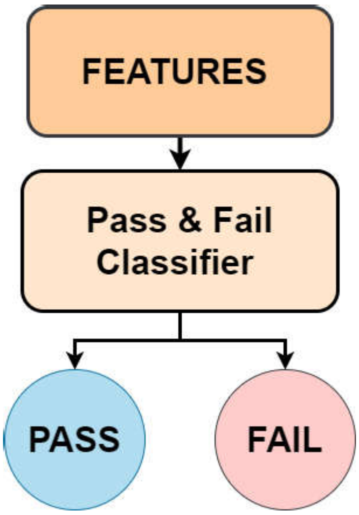

Figure 5.2. The structure of sAIQC which aims to separate the samples as pass or fail regardless of data quality.

Figure 5.2 shows the structure of sAIQC method. After the abovementioned steps are passed, the feature sets of two datasets, QLD and QULD, become ready for classification. Before the training step, hyper-parameters of four selected classifiers are optimized by employing 3-fold cross-validation. Cross-validation is also used for model evaluation, as mentioned above.

Whole QLD samples, the proper labeled subset of QLD, and the improper labeled subset of QLD are used for training and testing separately. Hyperparameter optimization is also applied with these datasets respectively.

sAIQC are analyzed and compared for the defined problem in the following section. The experiments of QLD are presented in Section 5.1.2. Section 5.1.3 gives the training details. All results of the experiments are shown in Section 5.1.4. The results of the experiments are discussed on various aspects in Section 5.1.5.

5.1.2. QLD Experiments

Three cross-validation folds are used for both optimizations of hyper-parameters and model evaluation. Each folds are created with keeping class balance ratio and the distribution of defect types. As mentioned in Section 4.3.1, all defect types are labeled as fail.

Table 5.1. The content of a cross-validation fold of QLD for the sAQIC.

	Pass	**Fail**					
	Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**
Train Fold	400	0	4	0	31	2	10
Test Fold	200	0	3	0	16	1	5
Total	600	0	7	0	47	3	15

A cross-validation fold of QLD is shown in Table 5.1. After finding optimal parameters of classifiers, the training folds are fitted to the frozen classifier, respectively, and the test fold is predicted. Before passing another train fold, test results are kept. The following values are used to optimize the hyper-parameters of classifiers trained with QLD:

- SVM: The C parameter is 100 and the γ parameter is 0.1.
- RF: The *n_estimators* is 313, the *max_depth* is 10 ,*criterion* is entropy.
- XGB: The *n_estimators* is 3400 , the *max_depth* is 9 ,the *subsample* is 0.368 ,the η is 0.03.
- MLP: The size of hidden layer one is 64 , drop-out rate one is 0.5, the size of hidden layer two is 32, drop-out rate two is 0.4, the size of hidden layer three is 8, the learning rate of optimizer is 0.001. The number of epoch is 500 and batch size 16.

Table 5.2. The content of a cross-validation fold of the proper labeled subset of QLD for the sAQIC.

	Pass	**Fail**					
	Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**
Train Fold	292	0	2	0	13	1	5
Test Fold	146	0	1	0	6	0	2
Total	438	0	3	0	19	1	7

A training and test fold content of the proper labeled subset in QLD is given in Table 5.2. Healthy samples are labeled as pass, and faulty motors are labeled as fail. The distribution of defect types in the fail class is preserved for each fold. The following list summarizes the hyper-parameters of classifiers trained on only proper labeled instances of QLD:

- SVM: The C parameter is 1 and the γ parameter is 0.1.
- RF: The *n_estimators* is 117 ,the *max_depth* is 98, *criterion* is entropy.
- XGB: The *n_estimators* is 1800 , the *max_depth* is 3,the *subsample* is 0.243 ,the η is 0.08.
- MLP: The size of hidden layer one is 64 , drop-out rate one is 0.4, the size of hidden layer two is 32, drop-out rate two 0.3, the size of hidden layer three is 8, the learning rate of optimizer is 0.001. The number of epoch is 100 and batch size 16.

Table 5.3. The content of a cross-validation fold of the improper labeled subgroup of QLD for the sAQIC.

	Pass	**Fail**					
	Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**
Train Fold	108	0	3	0	19	1	5
Test Fold	54	0	1	0	9	0	3
Total	162	0	4	0	28	1	8

The improper labeled subset in QLD has a training and test fold content, as shown in Table 5.3. The pass label represents healthy instances, and it is a negative label, 0. Fail label shows the faulty samples with the positive label, 1. The hyper-parameters of classifiers trained with only improper labeled instances are shown below:

- SVM: The C parameter is 1 and the γ parameter is 0.01.
- RF: The *n_estimators* is 451, the *max_depth* is 27, *criterion* is entropy.
- XGB: The *n_estimators* is 733, the *max_depth* is 9,the *subsample* is 0.458,the η is 0.1.
- MLP: The size of hidden layer one is 64 , drop-out rate one is 0.4, the size of hidden layer two is 16, drop-out rate two 0.3, the size of hidden layer three is 8, the learning rate of optimizer is 0.01. The number of epoch is 300 and batch size 8.

5.1.3. QULD Experiments

As mentioned in Section 4.3.2, this data set does not have data acquisition quality labels. Thus sAIQC can be applied to the whole dataset. The standard hyper-parameter optimization step as mentioned in Section 4.3.2 is also employed before training.

Table 5.4. The detailed content of a cross-validation fold of QULD.

	Pass	**Fail**					
	Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**
Train Fold	16787	43	39	3	55	73	53
Test Fold	8393	22	19	2	28	37	26
Total	25180	65	58	5	83	110	79

The details of a cross-validation fold of QULD about the pass and fail distribution and the number of failure types in the fail class are also shown in Table 5.4. The hyper-parameters of classifiers trained with QULD are given below:

- SVM: The C parameter is 100 and the γ parameter is 0.0001.
- RF: The *n_estimators* is 460 ,the *max_depth* is 5, *criterion* is entropy.
- XGB: The *n_estimators* is 1266 , the *max_depth* is 3,the *subsample* is 0.672 ,the η is 0.0001.
- MLP: The size of hidden layer one is 64 , drop-out rate one is 0.6, the size of hidden layer two is 32, drop-out rate two 0.3, the size of hidden layer three is 16, the learning rate of optimizer is 0.0001. The number of epoch is 500 and batch size 128.

5.1.4. Results

The evaluation results of each classifier trained with all different dataset and their subset in the sAIQC method, which are TPR, TNR, and UA of classifiers, are summarized with following tables.

Table 5.5. The final evaluation results with SVM, RF, XGB, and MLP classifiers for QLD in sAIQC. All classifiers have similar UAs. RF, which is colored blue, has the highest unweighted accuracy among other classifiers.

Classifier	**Sensitivity**	**Specificity**	**Unweighted Accuracy**
SVM	0.859	0.810	0.835
RF	0.887	0.810	**0.849**
XGB	0.873	0.777	0.825
MLP	0.955	0.676	0.816

The evaluation results of each classifier trained with the whole dataset QLD in the sAIQC method are given in Table 5.5. RF is the best performing classifier trained with QLD. It reaches 0.849 unweighted accuracy.

Table 5.6. The results of RF as a sAIQC classifier trained with QLD, which have the highest unweighted accuracy, are given in the below table. True predicted instances are shown with blue color, and false predicted samples are represented with red color.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	486	0	2	0	4	1	1	**TNR** 0.810
	Fail	114	0	5	0	43	1	14	**TPR** 0.887
		TNR 0.810	**TPR** -	**TPR** 0.714	**TPR** -	**TPR** 0.915	**TPR** 0.500	**TPR** 0.933	**UA** 0.849

Table 5.6 shows a detailed analysis of the result of RF classifier. TPR and TNR are computed for each defect type. TPR shows the percentage of actual fails which are correctly identified, and TNR is the outcome where the model correctly predicts the pass motors. The fail samples are taken positive class, and vice versa, pass instances are the negative class as abovementioned in Chapter 5. Furthermore, the prediction details of other classifiers are given in Table B.1, Table B.2, Table B.3, respectively, in Appendix.

Table 5.7. The final evaluation results of SVM, RF, XGB, and MLP classifiers trained with only proper samples in sAIQC. MLP, which is colored with blue, is the best-performing classifier with 0.936 unweighted accuracy.

Classifier	Sensitivity	Specificity	Unweighted Accuracy
SVM	0.870	0.967	0.916
RF	0.870	0.967	0.916
XGB	0.799	1.0	0.900
MLP	0.933	0.938	**0.936**

Table 5.7 depicts the results of each classifier trained with only proper collected samples. Sensitivity, Specificity, and Unweighted Accuracy are given as the performance metrics.

Table 5.8. The results of the best-performing classifier trained with samples labeled as proper, MLP, are given. Blue-colored samples are correctly predicted samples, and red-colored instances represent incorrect predictions.

		Actual							
		Pass	Fail						
		Pass	Vib1	Vib2	Vib3	Vib1&2	Vib	Sound	Performance metrics
Predicted	Pass	411	0	0	0	0	1	1	TNR 0.938
	Fail	27	0	3	0	19	0	6	TPR 0.933
		TNR 0.938	TPR -	TPR 1.000	TPR -	TPR 1.000	TPR 0.000	TPR 0.857	UA 0.936

The detailed results of MLP trained with proper samples are shown in Table 5.8. TPR and TNR are computed in the same way as described before. Another experiment results belong to classifiers trained with only improper collected samples.

Table 5.9. The final evaluation results with SVM, RF, XGB, and MLP classifiers trained with only improper samples of QLD in sAIQC.

Classifier	**Sensitivity**	**Specificity**	**Unweighted Accuracy**
SVM	0.976	0.722	**0.849**
RF	0.951	0.728	0.840
XGB	0.878	0.728	0.803
MLP	0.951	0.722	0.837

The sensitivity, specificity, and unweighted accuracy of each classifier trained with only improper collected samples are shown in following table. The sensitivities, specificities, and unweighted accuracies of the classifiers are shown. SVM has the highest unweighted accuracy with 0.849.

Table 5.10. The results of SVM trained with improper labeled instances are given. The correct predictions are shown with blue and incorrect predictions are represented with red color.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	117	0	0	0	1	0	0	**TNR** 0.722
	Fail	45	0	4	0	27	1	8	**TPR** 0.976
		TNR 0.722	**TPR** -	**TPR** 1.000	**TPR** -	**TPR** 0.964	**TPR** 1.000	**TPR** 1.000	**UA** 0.849

The prediction results of SVM trained with only improper collected instances are given in Table 5.10. The calculated unweighted accuracy of SVM is 0.849. The details of the result on other faulty types are also given. TPR and TNR are calculated for each fault type separately.

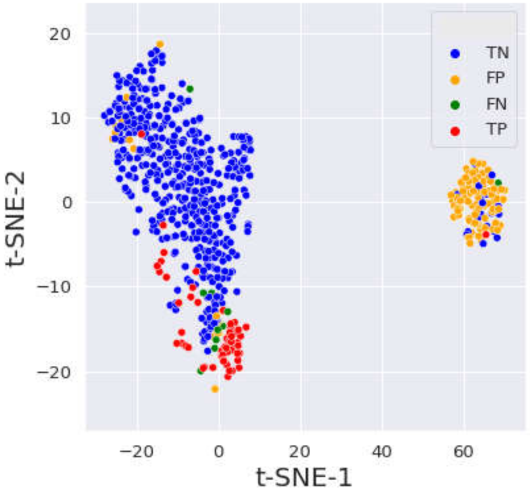

Figure 5.3. t-SNE visualizations of the prediction outcomes of the best-performing classifier RF trained QLD with 0.849 unweighted accuracy.

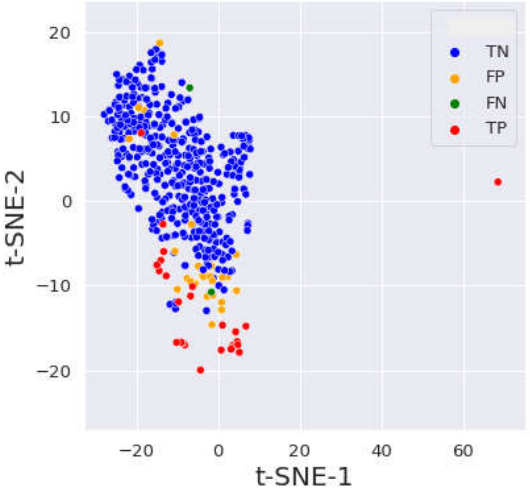

Figure 5.4. t-SNE visualizations of the prediction outcomes of MLP trained with proper subset of QLD, which has the highest unweighted accuracy with 0.936.

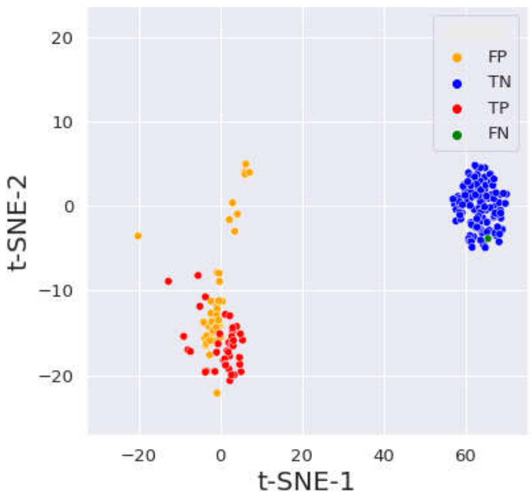

Figure 5.5. t-SNE visualizations of the prediction outcomes of SVM trained with improper subset of QLD, which has the highest unweighted accuracy with 0.849.

Figure 5.3, Figure 5.4 and Figure 5.5 show the predicted labels of best-performing classifiers trained respectively with QLD, properly labeled samples, and improper collected instances. TP, FP, TN, FN results are demonstrated at plots. TP corresponds to the fail class having been correctly predicted by the model. Also, TN is the model's predictions where the pass class is correctly predicted. FP is an outcome where the model predicts the fail class when it should instead have predicted the pass class. FN are situations where the model predicts the pass class, even though the pass class is, in fact, not present. The dots represent the same measure in Figure 5.3, Figure 5.4 and Figure 5.5. Blue points represent TN. Orange points symbolize FP. Green points are FN. Red dots correspond to TP. Figure 5.3 shows the prediction results of best-performing classifier RF trained with properly labeled samples with 0.849 unweighted accuracy. Figure 5.4 depicts the predicted outputs of MLP, which has the highest unweighted accuracy with 0.936. Figure 5.5 describes the prediction outputs of SVM with 0.849 unweighted accuracy, which classifier best suits on improperly collected instances.

Table 5.11. The evaluation results with SVM, RF, XGB, and MLP classifiers trained with QULD in sAIQC. XGB is the best-performing classifier with 0.747 unweighted accuracy.

Classifier	**Sensitivity**	**Specificity**	**Unweighted Accuracy**
SVM	0.665	0.787	0.726
RF	0.508	0.918	0.713
XGB	0.640	0.853	**0.747**
MLP	0.708	0.713	0.710

Table 5.11 depicts the outputs of the evaluation results, which are sensitivity, specificity, and unweighted accuracy of each classifier trained with QULD. Sensitivities, Specificities and Unweighted Accuracies of the classifiers are shown. XGB is the best-performing classifier with 0.747 unweighted accuracy.

Table 5.12. The outputs of XGB trained with QULD in sAIQC are given. TPR, TNR an UA metrics are given.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	21487	8	18	3	12	50	53	**TNR** 0.853
	Fail	3693	57	40	2	71	60	26	**TPR** 0.640
		TNR 0.853	**TPR** 0.877	**TPR** 0.690	**TPR** 0.400	**TPR** 0.855	**TPR** 0.545	**TPR** 0.329	**UA** 0.747

The prediction results of XGB trained with QULD in Table 5.12. The true predicted and false predicted samples in each class and the prediction outputs of faulty types are shown in Table 5.12. TPR and TNR of each subgroup are also depicted. In addition, the prediction details of other classifiers are given in Table B.4, Table B.5, Table B.6, respectively, in Appendix.

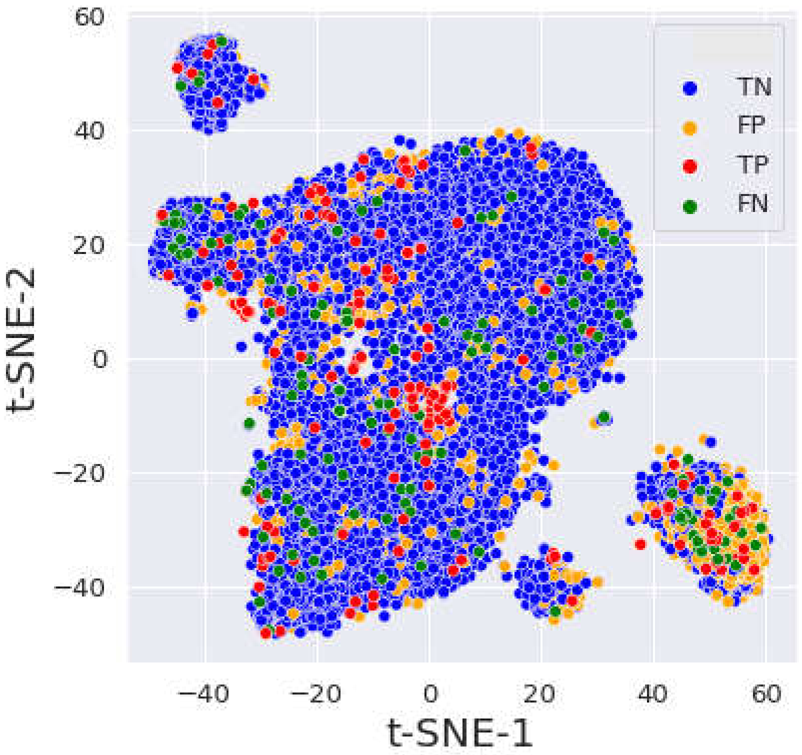

Figure 5.6. t-SNE visualization of the prediction results of XGB trained with QULD in sAQIC. TP is shown with red dots. Additionally, TN is represented by blue dots. FPs are indicated by orange dots. FNs are shown with green dots.

Figure 5.6 depicts the predicted results of best-performing classifier (XGB with 0.745 unweighted accuracy) trained with QULD. TP, FP, TN, and FN results are shown in plots.

5.1.5. Discussion

The sAIQC method is applied on QLD, its proper labeled subset, improper labeled subset, and QULD. Table 5.5 depicts the performance of each classifier trained with the whole dataset QLD. The unweighted accuracy is 0.849 as a result of the RF. The RF's sensitivity is 0.887, and its specificity is 0.810. Specificity values for each classifier are less than their sensitivities in Table 5.5. It means that all classifiers tend to label this data as positive, which means fail. This suggests that recordings mimic failure to a great extend. TPRs are not evenly distributed across different failure types in Table 5.6. Vib failure types are harder to detect, while the Vib1&2, Sound, and Vib2 failure can be predicted correctly. This is probably due to the limited number of samples in Vib defect type. Figure 5.3 shows that the vast majority of FP are clustered around 70 on the t-SNE 1 (x-axis) and 0 on the t-SNE 2 (y-axis) and standing apart throughout the dataset. This cluster in Figure 5.5 consists of improper labeled samples, and the majority of it is pass samples. In this case, it is seen that the majority of FPs gather in this cluster. Moreover, the fail samples overlapping with pass instances are predicted as pass, and most of the FNs occur around the border of two classes.

Table 5.7 displays the performance of the classifier trained on properly labeled QLD samples. Each classifier's performance is significantly improved. While the sensitivity values increase, the most significant improvement is seen in the classifiers' specificity values in Table 5.7. The best performing classifier MLP with 0.936 UA, 0.933 TPR and 0.938 TNR in Table 5.8 achieve to classify each class properly. Vib failure type could not be detected in this case either, and one sample from sound failure type is predicted as pass. This is likely to be caused by incorrectly labeled samples. After all, those labels are generated automatically by the standard (commercial) quality control system and have not been confirmed manually. Listed below is the prediction performance of the MLP, as shown in Figure 5.4. When similar features and overlaps are found in samples of various classes, these samples lead to misclassification. Most of the FPs are found around decision boundaries between pass and fail classes.

Table 5.9 list the performance of the classifier trained with improper labeled instances. The sensitivities of the classifiers are higher than the specificities, and each classifier tends to label as positive. The highest unweighted accuracy belongs to SVM with 0.849. TNR value of this classifier is 0.722, and its TPR is 0.976. Table 5.10 shows the prediction result of SVM in detail. The vast majority of fails are predicted correctly, SVM and other classifiers struggle to classify improperly collected pass samples due to lower specificity values than sensitivities. Figure 5.5 depicts the prediction results of SVM. All pass samples placed near 0 on the t-SNE 1 (x-axis) and -25 to 5 on the t-SNE 2 (y-axis), close to fail samples, are mispredicted. On the other hand, other samples are correctly predicted except for FPs because they are linearly separable. The reason for the occurrence of FPs, in this case, is that those labels are then provided manually by observation during data collection and can be incorrectly labeled.

If all results in Figure 5.3, Figure 5.4 and Figure 5.5 are considered together, if the instances from different data quality are not the same training group, samples become linearly separable. Especially pass samples that collect improperly have a major degradation effect on the performance of the classifier.

To sum up, if the training data are properly collected, the best-fitting classifier's (MLP) performance can reach 0.936 unweighted accuracy, and it can separate the classes correctly with high sensitivity of 0.933 and high specificity of 0.938. However, improperly labeled instances, especially pass samples, gets harder to classify. This problem causes the performance degradation of classifiers. The classifier's (SVM) sensitivity is 0.976, but its specificity is only 0.722. Finally, if the data quality is not known, the classifier's performance reaches only 0.849, and the fundamental limiting factors on the model's performance have pass samples with improper labels and class overlapping problems. Hence, rather than classify the improperly recorded data, it may be preferable to detect improper data and flag it without performing motor quality control (healthy/failure). It is understood from here that when there is improperly recorded data in the training set, classification becomes difficult, and prediction performance decreases. sAIQC performance gets stuck at one point if the quality of the data is unknown.

Table 5.11 shows the performance metrics of each classifier trained with another dataset, QULD. The best-fitted classifier on this dataset is XGB with 0.747 UA. The classifier's low sensitivity score of 0.640 shows that it is more likely to label datasets as pass. This is most likely due to class imbalance and the overlapping of samples from different classes. These samples cause uncertainty in decision boundary determination, affecting classification performance. TPRs in Table 5.12 are not equally distributed among various types of failures. Vib2, Vib3, Vib, and Sound failures are difficult to detect, whereas Vib1&2 and Vib1 failures can be predicted correctly. This is probably related to limited Vib3 samples, and that different failure types and pass samples may share similar characteristics. Moreover, as mentioned above, improper collected pass samples can cause misclassification of fail samples. Besides that in Figure 5.6, four clustered subsets can be observed. The cluster around 50 on the t-SNE 1 (x-axis) and -30 on the t-SNE 2 (y-axis) that distinct from the rest of the dataset seem to have high FPs and FNs. The number of FPs in this cluster is higher than other narrow clusters, as shown in Figure 4.18. This is likely caused by data quality, which is missing in QULD.

5.2. Double-Stage AIQC

5.2.1. Method

The proposed dAIQC is a two-stage classification method. First, binary classification is made based on data quality, and then pass or fail classification processes are made based on the result of the first stage. The second stage comprises two different binary classifiers. One of them, Pass & Fail Classifier 1, tries to classify the samples, which are predicted as proper, as pass or fail. The second classifier, Pass & Fail Classifier 2, categorizes the improper predicted instances into pass and fail.

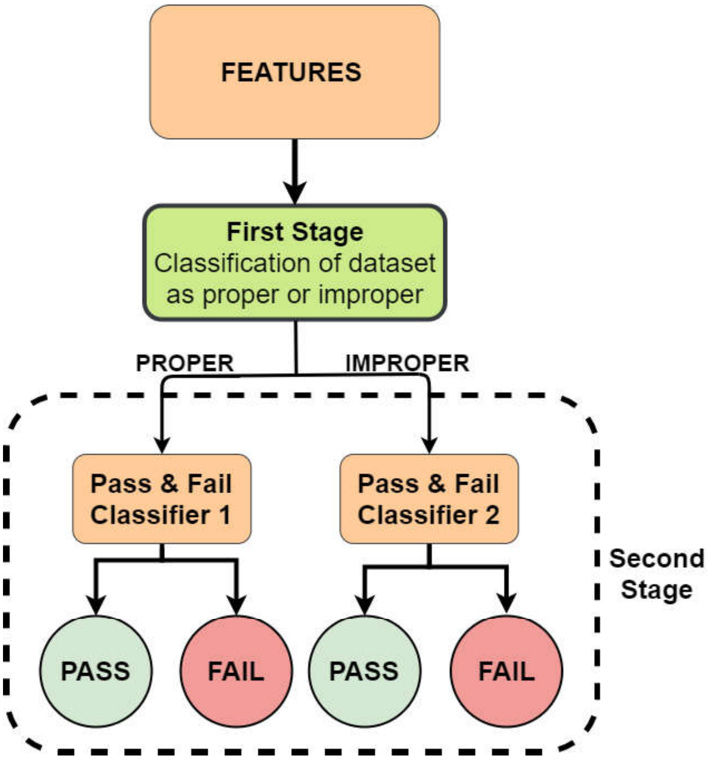

Figure 5.7. The structure of dAIQC: The first step of this method is to categorize the data into its quality labels, proper or improper at First-Stage Classifier. The separated subsets are tried to classify as pass or fail. Pass & Fail Classifier 1 categorizes the proper predicted data into the pass and fail. Pass & Fail Classifier 2 is to classify improper predicted data as pass and fail.

The proposed method structure is shown in Figure 5.7. Due to the absence of QULD quality labels, the first stage of dAIQC can only be trained using QLD.

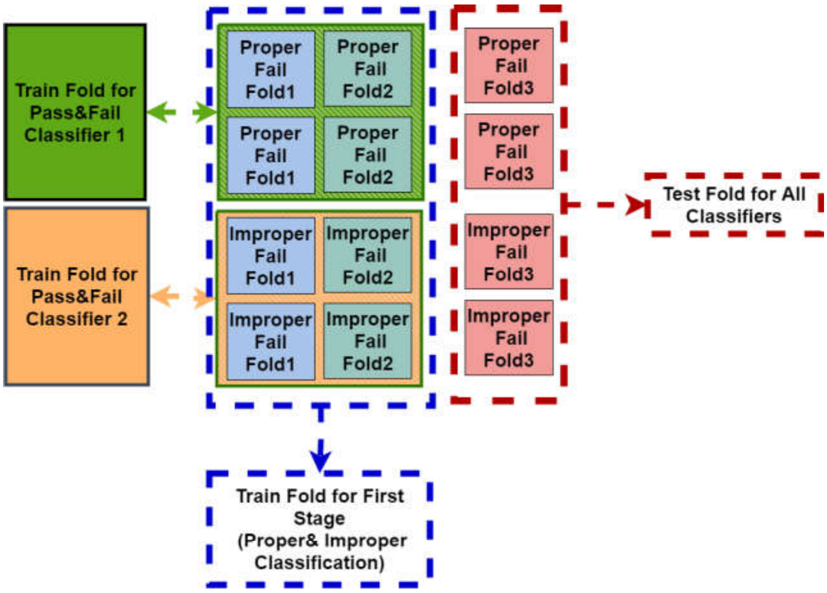

Figure 5.8. The content of a cross-validation fold for QLD. Each fold equally contains samples with both proper and pass labeled, both proper and fail labeled, both improper and pass labeled, and both improper and fail labeled, which are divided with preserving the balance between each subset in the data set.

The prepared feature set of QLD is split into three cross-validation folds as shown in Figure 5.8. In this case, each fold equally contains samples with both proper and pass labeled, both proper and fail labeled, both improper and pass labeled, and both improper and fail labeled, which are divided with preserving the balance between each subset in the data set. The first stage classifier and each second stage classifier are trained with different subsets on the same training set. Firstly, the whole samples in a training fold are used to train the First-Stage Classifier using proper and improper labels. The Pass & Fail Classifier 1 is trained with proper labeled samples in this training fold using the pass and fail labels. Likewise, the other Pass & Fail Classifier 2 is trained with improper labeled samples in the training fold. After training processes are finished, the test fold, as shown in figure with red dashed box, is predicted at the first stage classifier. The proper or incorrect predicted samples are used to test the second stage classifiers separately, depending on the outcome of the first stage. This procedure is repeated for each fold of cross-validation.

The hyper-parameters of the First-Stage Classifier are tuned by using three folds in Figure 5.8. The classifiers, SVM, RF, XGB, and MLP, as abovementioned in Section 5 are considered for each classification process. The optimized hyper-parameters of second-stage classifiers are already found in sAIQC for QLD. The proper and improper labeled samples are used for optimizing classifiers in sAIQC separately.

QULD is classified as proper or improper by the First-Stage Classifier trained with QLD. QULD is considered as a test set. Following that, second-stage classifiers are trained with the prediction results of the first stage of dAIQC separately.

5.2.2. QLD Experiments

Table 5.13. The content of training and test fold of QLD for dAIQC. The entire training fold is used for training of First-Stage Classifier, which categorizes the samples as improper or proper. While Pass & Fail Classifier 1 is trained with the proper labeled subset of the training fold, Pass & Fail Classifier 2 is fitted on the improper labeled part of the training fold.

		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Total**
Training Fold	**Proper**	292	0	2	0	13	1	5	313
	Improper	108	0	3	0	19	1	5	136
Test Fold	**Proper**	146	0	1	0	6	0	2	155
	Improper	54	0	1	0	9	0	3	67

The content of a cross-validation fold is given in Table 5.13. The fold contains proper and improper labeled samples while preserving the pass/fail distribution. The First-Stage Classifier uses the entire training fold. Pass & Fail Classifier 1 is trained on the training fold's proper labeled samples, while Pass & Fail Classifier 2 is fitted on the training fold's improper labeled samples. After training, the testing process is made end to end. The test fold is predicted as proper or improper at the First-Stage Classifier. Based on the results of the first-stage classifier, second-stage classifiers are evaluated on the relevant subset. The hyper-parameters of First-Stage Classifier are listed below:

- SVM: The C parameter is 1 and the γ parameter is 0.1.
- RF: The *n_estimators* is 333, the *max_depth* is 49, *criterion* is entropy.
- XGB: The *n_estimators* is 1266, the *max_depth* is 3,the *subsample* is 0.672, the η is 0.0001.
- MLP: The size of hidden layer one is 32 , drop-out rate one is 0.2, the size of hidden layer two is 32, drop-out rate two 0.2, the size of hidden layer three is 16, the learning rate of optimizer is 0.001. The number of epoch is 300 and batch size 32.

The hyper-parameters of second-stage Pass & Fail Classifier 1, which classify proper predicted samples into pass or fail, is shown below:

- SVM: The C parameter is 1 and the γ parameter is 0.1.
- RF: The *n_estimators* is 117 ,the *max_depth* is 98, *criterion* is entropy.
- XGB: The *n_estimators* is 1800 , the *max_depth* is 3,the *subsample* is 0.243 ,the η is 0.08.
- MLP: The size of hidden layer one is 64 , drop-out rate one is 0.4, the size of hidden layer two is 32, drop-out rate two 0.3, the size of hidden layer three is 8, the learning rate of optimizer is 0.001. The number of epoch is 100 and batch size 16.

The hyper-parameters of second-stage Pass&Fail Classifier 2 is given below:

- SVM: The C parameter is 1 and the γ parameter is 0.01.
- RF: The *n_estimators* is 451, the *max_depth* is 27, *criterion* is entropy.
- XGB: The *n_estimators* is 733, the *max_depth* is 9,the *subsample* is 0.458,the η is 0.1.

- MLP: The size of hidden layer one is 64 , drop-out rate one is 0.4, the size of hidden layer two is 16, drop-out rate two 0.3, the size of hidden layer three is 8, the learning rate of optimizer is 0.01. The number of epoch is 300 and batch size 8.

5.2.3. QLD Results

Table 5.14. The evaluation results of First-Stage Classifiers, which are SVM, RF, XGB, and MLP, trained with QLD. RF is the best-performing classifier with 0.839 unweighted accuracy.

Classifier	**Sensitivity**	**Specificity**	**Unweighted Accuracy**
SVM	0.729	0.912	0.821
RF	0.704	0.974	**0.839**
XGB	0.690	0.972	0.831
MLP	0.768	0.885	0.827

The First-Stage Classifier of dAIQC's prediction results, which include sensitivity, specificity, and unweighted accuracy, are summarized in Table 5.14.

Table 5.15. The outputs of RF in the first stage of dAIQC. The samples are classified as improper and proper. The performans metrics are given.

		Actual		
		Proper	**Improper**	**Performance metrics**
Predicted	**Proper**	456	60	**TNR** 0.974
	Improper	12	143	**TPR** 0.704
		TNR 0.974	**TPR** 0.704	**UA** 0.839

The detailed prediction results for RF, the model with the highest unweighted accuracy of 0.839, are shown in Table 5.15. TPR represents the ratio of correctly predicted improper samples to given actual proper instances.TNR denotes the proportion of proper labeled samples that are correctly identified. The average of TNR and TPR represents the classifier's unweighted accuracy. Moreover, the prediction results of SVM, XGB and MLP is shown in Table B.7, Table B.8 and Table B.9, respectively.

Table 5.16. The prediction performances of the SVM, RF, XGB, and MLP classifiers for Pass & Fail Classifier 1. TPR, TNR and UA of the classifiers trained with the proper subset of QLD and tested with samples predicted as proper from the best-performing First-Stage Classifier, RF, are shown. MLP is the best-performing classifier with 0.927 UA.

Classifier	**Sensitivity**	**Specificity**	**Unweighted Accuracy**
SVM	0.878	0.968	0.923
RF	0.732	0.973	0.853
XGB	0.805	0.994	0.899
MLP	0.927	0.926	**0.927**

The evaluation results of each classifier for Pass & Fail Classifier 1 at the second stage are given in the Table 5.16. The samples of test fold, where the best performing classifier RF in the first stage of dAIQC predicts as proper, are predicted as pass or fail.

Table 5.17. The prediction outcomes of MLP as best-performing Pass & Fail Classifier 1. True predicted samples are indicated in blue. Incorrectly predicted instances are highlighted in red.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	440	0	1	0	0	1	1	**TNR** 0.926
	Fail	35	0	3	0	22	1	12	**TPR** 0.927
		TNR 0.926	**TPR** -	**TPR** 0.750	**TPR** -	**TPR** 1.000	**TPR** 0.500	**TPR** 0.923	**UA** 0.927

With 0.927, MLP achieves the highest unweighted accuracy value. The results of MLP's prediction are summarized in Table 5.17. TPRs for each defect type in the fail class and TNR are calculated using the best-performing Pass & Fail Classifier 1, MLP. In addition, the prediction results of SVM, RF and XGB is shown in Table B.10, Table B.11 and Table B.12, respectively.

Table 5.18. The performance metrics of each classifier for Pass & Fail Classifier 2 at the second stage of dAIQC. MLP has the best performance with 0.950 unweighted accuracy.

Classifier	**Sensitivity**	**Specificity**	**Unweighted Accuracy**
SVM	0.933	0.936	0.935
RF	0.867	0.992	0.929
XGB	0.900	0.992	0.946
MLP	0.900	1.000	**0.950**

The outcomes of each classifier for Pass & Fail Classifier 2 at the second stage of DAIQC. Performance metrics are summarized in Table 5.18. Sensitivities, specificities, and unweighted accuracies of the classifiers trained with the improper subset of QLD and tested with samples predicted as improper from the best-performing first-stage classifier, RF, are shown.

Table 5.19. The prediction results of MLP as best-performing Pass & Fail Classifier 2. True predicted samples are indicated in blue. Incorrectly predicted instances are highlighted in red.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	125	0	0	0	3	0	0	**TNR** 1.00
	Fail	0	0	3	0	22	0	2	**TPR** 0.900
		TNR 1.000	**TPR** -	**TPR** 1.000	**TPR** -	**TPR** 0.880	**TPR** -	**TPR** 1.000	**UA** 0.950

The detailed results of MLP's performance as the best classifier for Pass & Fail Classifier 2 are shown in Table 5.19. The performance metrics for the pass and each failure type in the fail class are calculated. Furthermore, the prediction details of SVM, RF and XGB is shown in Table B.13, Table B.14 and Table B.15, respectively.

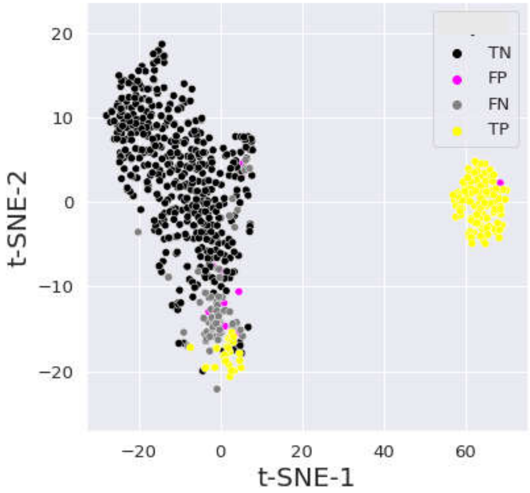

Figure 5.9. t-SNE visualizations of the prediction outcomes of the best-performing First-Stage Classifier, RF, which categorize QLD as improper or proper, with 0.839 unweighted accuracy. Black points denote TNs. Pink points represent FPs. Gray points indicate FNs. Yellow dots mark TPs .

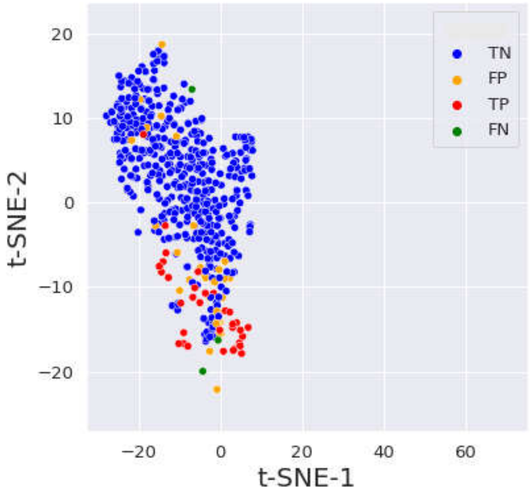

Figure 5.10. t-SNE visualizations of the prediction outcomes of MLP as Pass & Fail Classifier 1, which has the highest unweighted accuracy with 0.927. TNs are denoted by blue dots. Orange points are used to denote FPs. FNs are denoted by green points. Red dots indicate TPs.

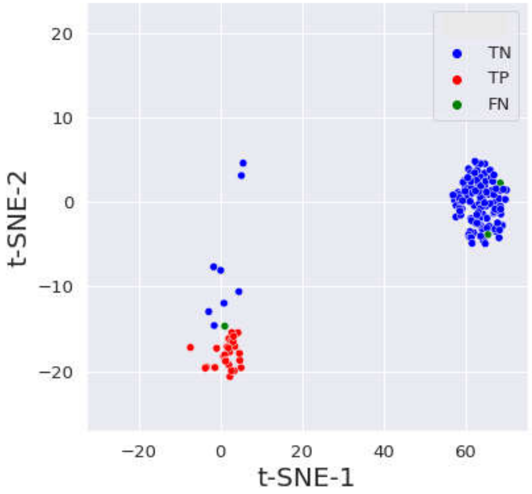

Figure 5.11. t-SNE visualizations of the prediction outcomes of MLP as Pass & Fail Classifier 2 with 0.950 UA. TNs are denoted by blue dots. Orange points are used to denote FPs. FNs are denoted by green points. Red dots indicate TPs.

Figure 5.9, Figure 5.10, Figure 5.11 depict the prediction outputs of best-performing classifiers for each stage classifier. Plots illustrate the TP, FP, TN, and FN results. For the result of the First-Stage Classifier, TPs correspond correctly predicted improper labeled samples. Also, TN is that correctly predicted proper instances. FP is an outcome where the model predicts the improper class when it should instead have predicted the proper category. FN are situations where the model predicts the proper class, even though it is from an improper class. For second stage classifiers, TPs are for true predicted fails, TNs are used for true predicted passes. FNs represent incorrectly predicted fails, and FPs shows incorrectly predicted passes.

Table 5.20. Summary table of results of Pass & Fail Classifiers at the second stage of dAIQC for QLD.

	Pass	Vib1	Vib2	Vib3	Vib1&2	Vib	Sound	Performance metrics
Pass & Fail Classifier 1	**TNR**	**TPR**	**TPR**	**TPR**	**TPR**	**TPR**	**TPR**	**UA**
	0.926	-	0.750	-	1.000	0.500	0.923	0.927
Pass & Fail Classifier 2	**TNR**	**TPR**	**TPR**	**TPR**	**TPR**	**TPR**	**TPR**	**UA**
	1.00	-	1.000	-	0,880	-	1,000	0.950
Total	**TNR**	**TPR**	**TPR**	**TPR**	**TPR**	**TPR**	**TPR**	**UA**
	0.942	-	0,857	-	0,936	0.500	0.933	0.929

Suppose the prediction outcomes of Pass & Fail Classifiers at the second stage of dAIQC in Table 5.17 and Table 5.19 are combined. In that case, the total performance of dAIQC for the pass or fail binary classification is given in Table 5.20.

5.2.4. QULD Experiments

Due to the lack of quality labels in QULD, the best-performing First-Stage Classifier trained with QLD, RF, is utilized to make data quality predictions on QULD. According to the prediction results of the first stage classifier, the dataset is split into proper or improper labeled subsets. Hyperparameter optimization procedures are applied on these two subsets to find the best hyper-parameters of second-stage classifiers by using 3-fold cross-validation. The model evaluation processes are completed.

Table 5.21. The prediction results of the First-Stage Classifier RF that trained with QLD on QULD. As quality labels are not available at QULD, the most efficient First-Stage Classifier with QLD, RF, is used to make predictions of data quality for QULD. After all, QULD is divided into two quality classes as improper and proper.

Quality	**Pass**	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**
proper	16897	56	39	1	77	55	49
	16897	277					
improper	8283	9	19	4	6	55	30
	8283	123					

The prediction results of QULD in the first stage classifier trained with QLD are shown in Table 5.21. The majority of the samples in QULD are predicted as proper, while %33 of the dataset are predicted as improper. After samples of QULD are grouped at the first stage of dAIQC, train and test folds are created for second-stage classifiers.

Table 5.22. The content of a training and test fold of QULD for Pass & Fail Classifier 1 of dAIQC.

	Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**
Train Fold	11265	37	26	1	51	37	33
Test Fold1	5632	19	13	0	26	18	16

A training and test fold of QULD for Pass & Fail Classifier 1 are given in Table 5.22. All fault types in fail and pass classes are depicted below. The following are the optimized hyper-parameters for the second-stage Pass & Fail Classifier 1 that are used to classify predicted samples as pass or fail:

- SVM: The C parameter is 1000 and the γ parameter is 0.0001.
- RF: The *n_estimators* is 460, the *max_depth* is 5, *criterion* is entropy.
- XGB: The *n_estimators* is 1266, the *max_depth* is 3, the *subsample* is 0.672, the η is 0.0001.
- MLP: The size of hidden layer one is 64 , drop-out rate one is 0.6, the size of hidden layer two is 16, drop-out rate two 0.5, the size of hidden layer three is 16, the learning rate of optimizer is 0.0001. The number of epoch is 500 and batch size 128.

Table 5.23. The content of a training and test fold of QULD for Pass & Fail Classifier 2 of dAIQC.

	Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**
Train Fold	5522	6	13	3	4	37	20
Test Fold1	2651	3	6	1	2	18	10

Figure 5.23 contains the training and test folds of QULD for Pass & Fail Classifier 2. The following diagram illustrates all fault types in the fail and pass classes. For the second-stage Pass & Fail Classifier 2, the following are the tuned hyper-parameters that are used to classify predicted samples as pass or fail:

- SVM:The C parameter is 1 and the γ parameter is 0.01.
- RF: The *n_estimators* is 460, the *max_depth* is 5, *criterion* is entropy.
- XGB: The *n_estimators* is 1266, the *max_depth* is 3, the *subsample* is 0.672, the η is 0.0001.
- MLP: The size of hidden layer one is 32 , drop-out rate one is 0.2, the size of hidden layer two is 32, drop-out rate two 0.2, the size of hidden layer three is 8, the learning rate of optimizer is 0.001. The number of epoch is 200 and batch size 512.

5.2.5. QULD Results

Table 5.24. The prediction performances of the SVM, RF, XGB, and MLP classifiers for Pass & Fail Classifier 1 for QULD. Sensitivities, specificities, and unweighted accuracies of the Pass & Fail Classifier 1 trained with proper predicted samples in the First-Stage Classifier are demonstrated.

Classifier	**Sensitivity**	**Specificity**	**Unweighted Accuracy**
SVM	0.874	0.963	0.918
RF	0.614	0.987	0.801
XGB	0.823	0.939	0.881
MLP	0.906	0.948	**0.927**

The evaluation results of each classifier as Pass & Fail Classifier 1 for QULD are listed in Table 5.24. MLP can reach the best performance metrics with 0.927 UA.

Table 5.25. The prediction results of MLP as best-performing Pass & Fail Classifier 1 of dAIQC for QULD. Correctly predicted samples are shown as blue-colored. Incorrectly predicted instances are red-colored.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	16012	2	3	1	3	12	5	**TNR** 0.948
	Fail	885	54	36	0	74	43	44	**TPR** 0.906
		TNR 0.948	**TPR** 0.964	**TPR** 0.923	**TPR** 0.000	**TPR** 0.961	**TPR** 0.782	**TPR** 0.898	**UA** 0.927

MLP has the highest unweighted accuracy value of 0.927. Figure 5.25 contains predictions for MLP and calculated TPRs for each defect type in the fail class and TNR. In addition, the prediction results of SVM, RF and XGB is shown in Table B.16, Table B.17 and Table B.18, respectively.

Table 5.26. The performance metrics of the SVM, RF, XGB, and MLP classifiers for Pass & Fail Classifier 2 for QULD.

Classifier	**Sensitivity**	**Specificity**	**Unweighted Accuracy**
SVM	0.683	0.892	0.788
RF	0.195	0.959	0.577
XGB	0.512	0.865	0.689
MLP	0.707	0.880	**0.794**

In Table 5.26, each classifier's evaluation results as Pass & Fail Classifier 2 for QULD are listed. Sensitivities, specificities, and unweighted accuracies of the Pass & Fail Classifier 2 trained with proper predicted samples in the First-Stage Classifier are shown.

Table 5.27. The prediction results of MLP as best-performing Pass & Fail Classifier 2 of dAIQC for QULD. Blue-colored values represent true predicted samples. Incorrectly predicted instances are highlighted in red.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	7289	3	9	1	3	10	10	**TNR** 0.880
	Fail	994	6	10	3	3	45	20	**TPR** 0.707
		TNR 0.880	**TPR** 0.667	**TPR** 0.526	**TPR** 0.750	**TPR** 0.500	**TPR** 0.818	**TPR** 0.667	**UA** 0.794

The prediction details of best-performing classifier with 0.794 unweighted accuracy, MLP, are given Table 5.27. The predicted, actual labels and performance metric TPRs, TNR are shown in detail. Blue-colored values represent true predicted samples. Incorrectly predicted instances are highlighted in red. Moreover, the prediction details of SVM, RF and XGB is shown in Table B.19, Table B.20 and Table B.21, respectively.

Table 5.28. Summary table of outputs of Pass & Fail Classifiers at the second stage of dAIQC for QULD.

	Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Pass & Fail	**TPR**	**TNR**	**TNR**	**TNR**	**TNR**	**TNR**	**TNR**	**UA**
Classifier 1	0.948	0.964	0.923	0.000	0.961	0.782	0.898	0.927
Pass & Fail	**TPR**	**TNR**	**TNR**	**TNR**	**TNR**	**TNR**	**TNR**	**UA**
Classifier 2	0.880	0.667	0.529	0.750	0.500	0.818	0.667	0.794
Total	**TPR**	**TNR**	**TNR**	**TNR**	**TNR**	**TNR**	**TNR**	**UA**
	0.925	0.923	0.793	0.600	0.928	0.800	0.810	0.895

The average performance metrics of best fitted pass-fail classifiers are summed up and the results of Pass & Fail Classifiers are summarized in Table 5.28. The total prediction performance are increased significantly.

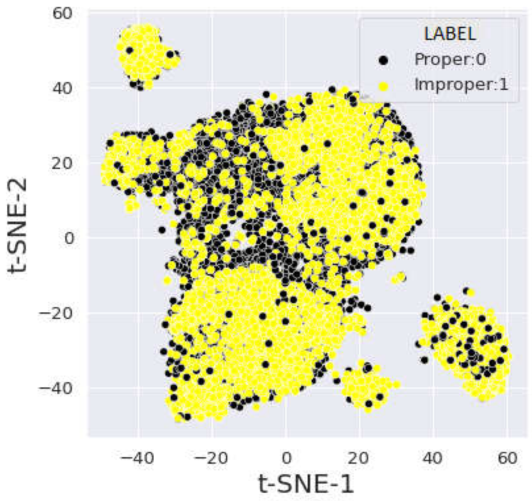

Figure 5.12. t-SNE visualizations of the prediction results of the best-performing First-Stage Classifier trained with QLD, RF, and tested on QULD. Black points denote the predicted samples as proper. Instances predicted as improper are marked by yellow dots.

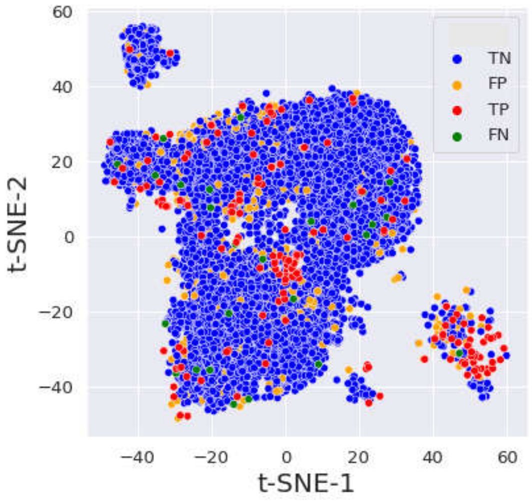

Figure 5.13. t-SNE visualizations of the prediction outcomes of MLP as Pass & Fail Classifier 1 trained with QULD subset, which has the highest unweighted accuracy with 0.927. TNs are denoted by blue dots. Orange points are used to denote FPs. FNs are denoted by green points. Red dots indicate TPs.

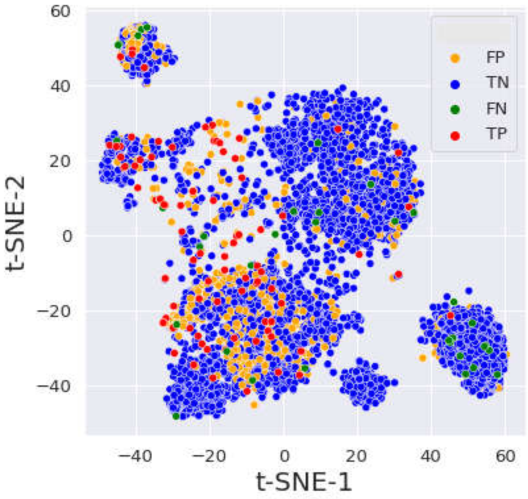

Figure 5.14. t-SNE visualizations of the prediction outcomes of MLP's output prediction as Pass & Fail Classifier 2 trained with QULD subset with 0.794 UA. TNs are denoted by blue dots. Orange points are used to denote FPs. FNs are denoted by green points. Red dots indicate TPs.

Figure 5.12 shows the evaluation outputs of First-Stage Classifier trained with QLD and tested on QULD. The outputs of best-performing classifiers for each second-stage classifier are depicted in Figure 5.13 and Figure 5.14. TP, FP, TN, and FN are represented in plots. TPs for true predicted fail samples, TNs for true predicted passes, FNs for incorrectly predicted fails, and FPs for false predicted passes are marked with red, blue, green, and orange, respectively.

5.2.6. Discussion

The first stage classification step of dAIQC can only be trained with QLD, and it is tested on QULD. Table 5.14 shows the performance metrics of classifiers trained with QLD in the first stage; RF has the highest unweighted accuracy with 0.839. The calculated specificity values in Table 5.14 greater than sensitivities indicates that all classifiers tend to label as proper. The vast of FNs is located near to TNs in Figure 5.9. By examining Figure 5.10, it can conclude that most of the FNs are pass samples from improper classes that overlapping with proper class. Thus, the issue of mounting the sensor at an angle to the motor body can be tolerated up to a point. A few proper labeled motors overlapping with the improper class are only confused with a low number of FPs. Almost all instances from the small-sized cluster, which clusters around 70 on the t-SNE 1(x-axis) and 0 on the t-SNE 2 (y-axis), are classified as improper in Figure 5.9.

Table 5.16 lists the performance of Pass & Fail Classifier 1 on QLD, which categorizes the samples predicted as proper into the pass and fail. MLP is the most accurate classifier, with a unweighted accuracy of 0.927, a sensitivity of 0.927, and a specificity of 0.926. Although improper labeled samples are predicted as proper at the first stage, the actual true prediction rate of Pass & Fail Classifier 1 is still sufficient. The distribution of TPRs across different failure types is not uniform in Table 5.17. Vib failures are still more difficult to detect, whereas Vib1&2, Sound failures can be accurately predicted. This is probably due to the limited number of samples in Vib defect type. If the data is collected properly, it can be concluded that the distinguishing features between the two classes are clarified. The majority of FPs in Figure 5.10 are seen borders pass and fail classes. The number of FNs is quite low, and fail samples near pass instances cannot be accurately predicted.

Table 5.18 shows the prediction performance of Pass & Fail Classifier 2 on QLD, which categorizes the samples predicted as improper into the pass and fail. MLP is the best-performing classifier with a unweighted accuracy of 0.950, a sensitivity of 0.900, and a specificity of 1.000. Due to reducing the number of samples after the first stage classifier and eliminating the overlapping samples by predicting as proper at the first stage, fail and pass classes become linearly separable. The vast of the pass samples locate far away from fail motors in Figure 5.11. Thus, the final prediction performance of the classifier makes significant progress. Only three samples from the Vib1&2 failure type are mispredicted in Table 5.19, and these FNs in Figure 5.11 are seen close to clustered pass samples, where locates around 70 on the t-SNE 1(x-axis) and around 0 on the t-SNE 2 (y-axis).

As seen in Table 5.20, MLP within each group with separately trained classifiers shows the best classification performance due to a clarification of distinctive features after separation of the dataset according to data quality. In the overall performance summary in Table 5.20, TPRs are making considerable progress except for the Vib failure type due to its limited number of instances.

Table 5.21 shows the predicted data quality labels of QULD. Sixty-seven percent of the dataset is predicted as proper. Although difficult to visualize the dataset with t-SNE, most of the samples from the cluster, where places around 50 on the t-SNE 1(x-axis) and -30 on the t-SNE 2 (y-axis), another cluster, where locates around 20 on the t-SNE 1(x-axis) and -40 on the t-SNE 2 (y-axis), and the cluster, where locates around -40 on the t-SNE 1(x-axis), and 50 on the t-SNE 2 (y-axis), are predicted as improper, as shown in the Figure 5.12.

The performance metrics of each classifier for Pass & Fail Classifier 1 on QULD are listed in Table 5.24. MLP achieves to reach the highest unweighted accuracy with 0.927 as same as unweighted accuracy of MLP trained with QLD at the second stage of dAIQC. TPRs are not evenly distributed across failure types in Table 5.17. The Vib failure type is still more difficult to detect, whereas the Vib1&2, Sound, Vib1 failure can be correctly predicted. This is likely caused by incorrect labeling during label generation from the current commercial quality control system. In Figure 5.13, FPs are distributed close to fail classes, and FNs are located among pass samples.

Table 5.26 shows the evaluation outcomes of each classifier as Pass & Fail Classifier 2 on QULD. All classifiers' specificity is lower than their sensitivities, and they are inclined to predict this subset as pass. MLP has the highest prediction performance among other classifiers, with a unweighted accuracy of 0.794. The details of MLP is given in Table 5.27. TPR of the classifier is 0.707, and except Vib failure type, all fail samples are confused with the pass. In Figure 5.13, the number of FP is given as 994 in Table 5.27, and the vast majority of FPs are clustered between -40 and 20 on the t-SNE 1(x-axis) where the fail samples are also located in this range. Most of the samples in the cluster around 50 on the t-SNE 1 (x-axis) and around -30 on the t-SNE

2 (y-axis) are predicted as pass, and this cluster can be same characteristics with the cluster in Figure 5.11 where locates apart from the remaining subset of QLD.

The summary of dAIQC on QULD is depicted in Table 5.28. As in QLD, MLPs are the best performing classifiers of the second stage of dAIQC in QULD. Although Pass & Fail Classifier 2 has low TPR, which is 0.707 since the majority of the fail class is predicted as proper in the First-Stage Classifier and the total TNR of the Pass & Fail Classifier 1 is high, with 0.906, the overall TNR with 0.845 is not decreased much. Especially the samples that are predicted as proper in the first stage can be classified as pass and fail easily. In this case, it can be said that the first stage classifier trained with QLD can detect proper labeled samples among QULD; on the other hand, the samples predicted as improper can predict limited. This indicates other problems affecting the data quality apart from improper mounting of the accelerometer on the body of the motor.

6. DISCUSSION

Two methods are proposed, sAIQC and dAIQC, in this thesis. It has been tried to overcome the difficulties in collecting sensitive data with high sampling in mass production conditions. sAIQC is a classical classification method for categorizing a motor's health status as pass and fail. dAIQC is composed of 2 stages. The first stage makes a binary classification based on data quality, and the second stage is a binary classification as healthy or faulty. dAIQC employs two different second-stage classifiers, one for each group as identified by the first stage.

In Table 6.1, proper labeled samples of QLD are separable with 0.938 TNR, 0.933 TPR, and 0.936 UA. On the other hand, both improper labeled samples and the whole QLD have the same UA value, with 0.849. Thus, if the sAIQC method is applied to the dataset without knowing the data quality label and there are samples in the data that degrade the data quality, we see that the performance of the classifiers is limited. This limited performance of sAIQC is that the data set contains all or part of the improper labeled data. Especially pass (healthy) instances are quite similar to the fail labeled if they are not collected properly. This resulted in significant reductions in the TNR with 0.810. Therefore, after separating the data according to the data quality, the increase in the TNR value is the main factor contributing to classifier performance improvement. The TNR of sAIQC on QLD increases from 0.810 to 0.942 after dAIQC is applied on QLD. Likewise, TPR rises from 0.887 to 0.915. Although sAIQC with improper labeled samples of QLD has a 0.722 TNR value, some pass samples in this subset are predicted as proper after the division of this subset at the first stage of dAIQC improper predicted subset became a linearly separable dataset with 1.0 TNR. Thus, it can be concluded that the classifier can tolerate problems in data quality up to a point. Even though The TPR value, which was 0.976 in the sAIQC method, decrease to 0.900 in the dAIQC method. This is because the fail labeled samples in the subset, which is predicted as improper, have decreased. Thus, it is said that fail samples show different characteristics than pass instances, whether it is quality label proper or improper.

Table 6.1. The comparison of sAIQC and dAIQC on QLD. The total results of sAIQC on QLD, a proper labeled subset of QLD, and improper labeled subset of QLD are given, respectively. After splitting of QLD at the first stage of dAIQC, each second stage classifiers of dAIQC and total evaluation results are shown.

Method	**Dataset**	**Pass TNR**	**Vib1 TPR**	**Vib2 TPR**	**Vib3 TPR**	**Vib1&2 TPR**	**Vib TPR**	**Sound TPR**	**Performance Metrics**
sAIQC	**QLD**	0.810	-	0.714	-	0.915	0.500	0.933	**TPR = 0.887** **TNR = 0.810** **UA = 0.849**
	Actual Proper QLD	0.938	-	1.000	-	1.000	0.000	0.857	**TPR = 0.933** **TNR = 0.938** **UA = 0.936**
	Actual Improper QLD	0.722	-	1.000	-	0.964	1.000	1.000	**TPR = 0.976** **TNR = 0.722** **UA = 0.849**
dAIQC	**Predicted as Proper QLD**	0.926	-	0.750	-	1.000	0.500	0.923	**TPR = 0.927** **TNR = 0.926** **UA = 0.927**
	Predicted as Improper QLD	1.000	-	1.000	-	0.880	-	1.000	**TPR = 0.900** **TNR = 1.000** **UA = 0.950**
	Summary of dAIQC	0.942	-	0.857	-	0.936	0.500	0.933	**TPR = 0.915** **TNR = 0.942** **UA = 0.929**

The proposed two methods are applied on QULD, and the comparison of the results is shown in Table 6.2. A significant increase in unweighted accuracy is observed from sAIQC with 0.747 UA to dAIQC with 0.895 UA. The sAIQC method on this dataset has 0.853 TNR and 0.640 TPR. Thus, sAIQC on QULD tends to label this dataset into pass due to class overlapping, imbalance class ratio, and absence of data quality labels. After the First-Stage Classifier of dAIQC, which is trained on QLD, divides the QULD with labels improper and proper, the performance of second-stage classifiers is improved for each group. The main progress among performance metric is that TPR of sAIQC increase 0.640 to 0.845 in dAIQC when dAIQC results are summarized. While the change in the TNR came to the fore when switching from the sAIQC method to the dAIQC method on the QLD, the increase in the TPR in the QULD essentially improved the performance. This is probably due to having a high-class imbalance ratio from QLD. The samples predicted as proper, which is 67 percent of QULD, can be classified with 0.927 UA, as same as the result of QLD in Table 6.1. Therefore, it is possible to say that the samples separated as proper in the first stage of dAIQC are valid in QULD by looking at the similar performance results, which have the same UA with 0.927, in both datasets.

Table 6.2. The comparison of two methods on QULD.

Method	**Dataset**	**Pass**	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance**
		TNR	**TPR**	**TPR**	**TPR**	**TPR**	**TPR**	**TPR**	**Metrics**
sAIQC	*QULD*	0.853	0.877	0.690	0.400	0.855	0.545	0.329	**TPR = 0.640** **TNR = 0.853** **UA = 0.747**
dAIQC	*Predicted as Proper QULD*	0.948	0.964	0.923	0.000	0.961	0.782	0.898	**TPR = 0.906** **TNR = 0.948** **UA = 0.927**
	Predicted as Improper QULD	0.880	0.667	0.526	0.750	0.500	0.818	0.667	**TPR = 0.707** **TNR = 0.880** **UA = 0.794**
	Summary of dAIQC	0.925	0.923	0.793	0.600	0,.928	0.800	0.810	**TPR = 0.845** **TNR = 0.925** **UA = 0.895**

On the other hand, the evaluation results with the samples of QULD predicted as improper are close to the outcomes of sAIQC. This is likely to be caused that this subset still contains some factors affecting data quality, which are not labeled before. The rate of correct prediction of Vib and Sound failure types that are detected by an operator and have not been confirmed manually increase.

To sum up, the results of the methods on both datasets show that the data can be classified if the data quality problem is exceeded or the factor that affects the data quality is labeled.

At the same time, when we compare the dAIQC method with the currently used vibration analysis system, the TPR of the dAIQC method is over %90 in both datasets when the data is collected properly. In contrast, the TPR of the current vibration analysis system is %60. The TPR of the classification made under the condition that the data is not collected properly is greater than %60. Therefore, it is said that the currently working quality control system can detect limited errors. At the same time, the operator dependency of the quality control process is reduced with the dAIQC system. The operator ceases to be a part of the quality control step. If the mechanical problems related to the data quality are overcome and the sensor is mounted properly, the vibration quality control test can be fully automated, and the labor allocated for this process can be reduced.

7. CONCLUSIONS AND FUTURE WORKS

This thesis focuses on machine-learning-driven quality control systems of BLDC motors despite the negative factors affecting data quality. Background noise effects can be eliminated in the laboratory environment during data collection. However, it is necessary to install high-cost infrastructures to collect high-precision vibration signals from a manufacturing plant by preserving data quality. These vibration data collection infrastructures should be acoustically isolated from the environment with an acoustic anechoic chamber, mechanically isolated from any vibration that may come from the production line, and vibration measurement is taken non-touch. In systems with this infrastructure, the proposed sAIQC method will classify the motors' health status correctly. The experiments show that data quality has a significant role in the accuracy of the classifier. The method answers with at least 0.934 of unweighted accuracy on the subset consisting of properly collected samples. The bottleneck of the proposed method is that there is limited tolerance on improper gathered samples, and the robustness of the model is highly dependent on the data quality. This conclusion can be reached by looking at the performance of the sAIQC on whole datasets, both QLD with 0.849 of unweighted accuracy and QULD with 0.747 of unweighted accuracy. Therefore, sAIQC with high accuracy is challenging to achieve in cost-constrained production systems. Besides, even an improper recording can still include meaningful information about the quality of motors. For this reason, despite the mounting problem of the accelerometer on the body of the motor, the creation of automatic and AI-powered quality control systems of BLDC motors are the primary concern in this thesis.

It is not always possible to prevent noise from external factors in production conditions. Many factors can affect the quality of the vibration signal collected from the engine. The most important among these factors is that the accelerometer is not mounted properly on the engine's body. Therefore, adding a data quality label considering this situation and using this label to eliminate the negative effect of the factor on the classification has increased the classifier's performance. In the experiments conducted with the dAIQC method, unweighted accuracies reach 0.929 for QLD and

0.895 for QULD. With this approach, when using the dAIQC method instead of the sAIQC method, the UA performance metric of the QLD increases by 0.08, and the QULD's increases by 0.148. The prediction performance of the second-stage classifiers in dAIQC could not be further improved due to the possible presence of factors affecting the data quality in data sets. Therefore, in case of improper data recording risk, pre-stages should be placed to minimize the effect of factors affecting data quality on classification performance.

To sum up, the usage of the dAIQC in test systems with a limited budget and non-isolation is ideal for the improvement of the accuracy of test results. If the data is below a certain quality, this situation can be given to the user as an alarm. Thus, the system can be arranged, the test can be repeated, or when the system cannot produce a response in an inevitable repetition, the model can be retrained by taking expert opinion. Moreover, a model that makes valuable inputs for the transition from a highly dependent system on the operator to a fully automatic system is also produced. The following conclusions based on our findings can be deduced.

- Data collection quality is curial for AIQC systems.
- The performance of AIQC systems remains limited in low-quality collected data.
- For such an AIQC system to be feasible in the field, either the data collection setup should be very controlled and properly collected data should be guaranteed, or the dataset should be classified according to data quality before being classified healthy or faulty.
- It is observed that the best-performing first stage classifier of dAIQC is RF as a proper or improper data quality classifier.
- In the motor quality classification, which is pass or fail, of groups separated by data quality in dAIQC, the best performance belongs to MLP.

The following studies can be made in the future based on this study.

- If the number of factors affecting the data quality is unknown, the first stage of the dAIQC method can be designed as unsupervised clustering, and in the second stage of dAIQC, Pass & Fail Classifiers can be added to the model as much as the number of clusters. Thus, classification performance can be increased even without labeling data quality.
- If there is known the number of factors affecting the data quality, a multi-class classification can be established in the first stage of the dAIQC method.

REFERENCES

1. Zhong, R. Y., X. Xu, E. Klotz and S. T. Newman, “Intelligent Manufacturing in the Context of Industry 4.0: A Review”, *Engineering*, Vol. 3, No. 5, pp. 616–630, 2017.

2. Gürdür, D., J. El-Khoury, T. Seceleanu and L. Lednicki, “Making Interoperability Visible: Data Visualization of Cyber-Physical Systems Development Tool Chains”, *Journal of Industrial Information Integration*, Vol. 4, pp. 26–34, 2016.

3. Ruppert, T., S. Jaskó, T. Holczinger and J. Abonyi, “Enabling Technologies for Operator 4.0: A Survey”, *Applied Sciences*, Vol. 8, No. 9, 2018.

4. Nurmimaki, M., *Quality 4.0: The 3 Most Important Technologies for Manufacturers*, 2020, `https://www.manufacturingtomorrow.com/article/2020/03/readers-choice-2020-quality-40-the-3-most-important-technologies-for-manufacturers/14914`, accessed in May 2021.

5. Godina, R. and J. C. O. Matias, “Quality Control in the Context of Industry 4.0”, *Industrial Engineering and Operations Management II*, pp. 177–187, Springer International Publishing, Cham, 2019.

6. Jaya, A., E. Purwanto, M. B. Fauziah, F. D. Murdianto, G. Prabowo and M. R. Rusli, “Design of PID-fuzzy for Speed Control of Brushless DC Motor in Dynamic Electric Vehicle to Improve Steady-State Performance”, *2017 International Electronics Symposium on Engineering Technology and Applications (IES-ETA)*, pp. 179–184, 2017.

7. Yedamale, P., “Brushless DC (BLDC) Motor Fundamentals”, *Microchip Technology Inc*, Vol. 20, pp. 3–15, 2003.

8. Kallaste, A., T. Vaimann and A. Belahcen, “Influence of Magnet Material Selection

on the Design of Slow-Speed Permanent Magnet Synchronous Generators for Wind Applications", *Elektronika Ir Elektrotechnika*, Vol. 23, pp. 31–38, 2017.

9. Harris, T. A. and M. N. Kotzalas, *Essential Concepts of Bearing Technology*, CRC press, 2006.

10. Rajagopalan, S., *Detection of Rotor and Load Faults in Brushless DC Motors Operating Under Stationary and Non-Stationary Conditions*, Georgia Institute of Technology, 2006.

11. Kudelina, K., B. Asad, T. Vaimann, A. Belahcen, A. Rassõlkin, A. Kallaste and D. V. Lukichev, "Bearing Fault Analysis of BLDC Motor for Electric Scooter Application", *Designs*, Vol. 4, No. 4, 2020.

12. Bayram, A., *Novelty Detection on Streaming Sensor Data for IIOT Applications*, Master's thesis, Bogazici University, 2019, accessed in September 2020.

13. Mathew, J. and R. Alfredson, "The Condition Monitoring of Rolling Element Bearings Using Vibration Analysis", pp. 447–453, 1984.

14. Caesarendra, W. and T. Tjahjowidodo, "A Review of Feature Extraction Methods in Vibration-Based Condition Monitoring and Its Application for Degradation Trend Estimation of Low-Speed Slew Bearing", *Machines*, Vol. 5, p. 21, 09 2017.

15. Howard, I., *A Review of Rolling Element Bearing Vibration Detection, Diagnosis and Prognosis*, Defence Science and Technology Organization Canberra (Australia), 1994.

16. Taylor, J., *The Vibration Analysis Handbook: A Practical Guide for Solving Rotating Machinery Problems*, VCI, 2003.

17. IFM, *Operating Instructions,Diagnostic Electronics for Vibration Sensors*, 2020, `https://www.ifm.com/mounting/80227555UK.pdf`, accessed in September 2020.

18. Concettoni, E., C. Cristalli and S. Serafini, “Mechanical and Electrical Quality Control Tests for Small DC Motors in Production Line”, *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, pp. 1883–1887, 2012.

19. Trajin, B., J. Régnier and J. Faucher, “Comparison Between Vibration and Stator Current Aanalysis for the Detection of Bearing Faults in Asynchronous Drives”, *IET Electric Power Applications*, Vol. 4, No. 2, pp. 90–100, February 2010.

20. Ágoston, K., “Fault Detection of the Electrical Motors Based on Vibration Analysis”, *Procedia Technology*, Vol. 19, pp. 547–553, 2015, 8th International Conference Interdisciplinarity in Engineering, INTER-ENG 2014, 9-10 October 2014, Tirgu Mures, Romania.

21. Jin, X. and T. W. S. Chow, “Anomaly Detection of Cooling Fan and Fault Classification of Induction Motor Using Mahalanobis-Taguchi System”, *Expert Syst. Appl.*, Vol. 40, No. 15, p. 5787–5795, Nov. 2013.

22. Glowacz, A., “Fault Diagnosis of Single-Phase Induction Motor Based on Acoustic Signals”, *Mechanical Systems and Signal Processing*, Vol. 117, pp. 65–80, 02 2019.

23. Rodriguez Verdugo, F., E. Concettoni, A. Bastari, C. Cristalli and M. Palma, “Automating End-of-Line Quality Control Tests of Small Brushed DC Motors”, *12th IMEKO TC10 Workshop on New Perspectives in Measurements, Tools and Techniques for Industrial Applications 2013*, pp. 126–131, 01 2013.

24. Li, L.-L., J.-Q. Liu, W.-B. Zhao and L. Dong, “Fault Diagnosis of High-Speed Brushless Permanent-Magnet DC Motor Based on Support Vector Machine Optimized by Modified Grey Wolf Optimization Algorithm”, *Symmetry*, Vol. 13, No. 2, 2021.

25. Lee, C.-Y. and C.-H. Hung, “Feature Ranking and Differential Evolution for Feature Selection in Brushless DC Motor Fault Diagnosis”, *Symmetry*, Vol. 13, No. 7, 2021.

26. Shifat, T. A. and J.-W. Hur, “EEMD Assisted Supervised Learning for the Fault Diagnosis of BLDC Motor Using Vibration Signal”, *Journal of Mechanical Science and Technology*, Vol. 34, No. 10, pp. 3981–3990, 2020.

27. Shifat, T. A. and J.-W. Hur, “ANN Assisted Multi Sensor Information Fusion for BLDC Motor Fault Diagnosis”, *IEEE Access*, Vol. 9, pp. 9429–9441, 01 2021.

28. Zhao, R., R. Yan, Z. Chen, K. Mao, P. Wang and R. Gao, “Deep Learning and Its Applications to Machine Health Monitoring”, *Mechanical Systems and Signal Processing*, Vol. 115, 2019.

29. Sun, W., S. Shao, R. Zhao, R. Yan, X. Zhang and X. Chen, “A Sparse Auto-Encoder-Based Deep Neural Network Approach for Induction Motor Faults Classification”, *Measurement*, Vol. 89, pp. 171–178, 2016.

30. Ince, T., S. Kiranyaz, L. Eren, M. Askar and M. Gabbouj, “Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks”, *IEEE Transactions on Industrial Electronics*, Vol. 63, No. 11, pp. 7067–7075, 2016.

31. Xiao, D., Y. Huang, X. Zhang, H. Shi, C. Liu and Y. Li, “Fault Diagnosis of Asynchronous Motors Based on LSTM Neural Network”, *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*, pp. 540–545, 2018.

32. Zimnickas, T., J. Vanagas, K. Dambrauskas, A. Kalvaitis and M. Ažubalis, “Application of Advanced Vibration Monitoring Systems and Long Short-Term Memory Networks for Brushless DC Motor Stator Fault Monitoring and Classification”, *Energies*, Vol. 13, No. 4, 2020.

33. Bergstra, J. and Y. Bengio, “Random Search for Hyper-Parameter Optimization.”, *Journal of machine learning research*, Vol. 13, No. 2, 2012.

34. Maaten, L. v. d. and G. Hinton, “Visualizing Data Using t-SNE”, *Journal of Machine Learning Research*, Vol. 9, No. Nov, pp. 2579–2605, 2008.

35. Piezotronics, P., *PCB 352C34 Product Manual*, PCB Piezotronics, 2000, https://www.pcb.com/contentStore/docs/pcb_corporate/vibration/products/manuals/352c34.pdf, accessed in September 2020.

36. Mitsubishi, *FR-A800 Inverter Instruction Manual*, 2018, https://eu3a.mitsubishielectric.com/fa/en/dl/10248/FR-A800_instruction_manual_detailed_ib0600503_eng_J.pdf, accessed in January 2020.

37. Welch, P., "The Use of Fast Fourier Transform for The Estimation of Power Spectra: A Method Based on Time Averaging over Short, Modified Periodograms", *IEEE Transactions on Audio and Electroacoustics*, Vol. 15, No. 2, pp. 70–73, 1967.

38. He, H. and Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications*, 2013.

APPENDIX A: STFT

The below figure shows the calculation of STFT that are used in Figure 4.2. The STFT divides the time-domain input signal into overlapping frames by multiplying the signal with a window function, W, and then applies the FFT to each block.Finally, the spectral range of the signal over different time intervals is obtained. As a result, the input signal is represented in two dimensions.

```
$\chi_c = \{x_{c1}, x_{c2}, \ldots, x_{cn}\}$
$Win = \{win_1, win_2, \ldots, win_m\}$
The number of samples between each successive FFT window: $hop$
The number of FFT points: $nfft$
Sampling Rate: $fs$
Determination of signal length $x_{clen}$
Determination of window length $w_{len}$
Compute the number of uniq FFT points $N_{up} = \frac{1+nfft}{2}$
Compute the number of frames $L = 1 + \frac{x_{clen} - w_{len}}{hop}$
for $l = 1, ..., L-1$ {For each frame} do
    windowing of signal frame $xw = \chi_c[1 + lhop : w_{len} + lhop]W$
    compute fft $FFT = \sum_{m=0}^{nfft-1} xw_m e^{-j2\pi \frac{mk}{nfft}} \qquad k = 0, \ldots, nfft-1$
    store fft in a STFT matrix $STFT[:, l+1] = FFT[1 : N_{up}]$
end for
compute time vector $t = \{\frac{w_{len}}{2}, \frac{w_{len}}{2} + hop, \ldots, \frac{w_{len}}{2} + (L-1)hop\}$
compute frequency vector $f = \{0, \frac{fs}{nfft}, \ldots, (N_{up} - 1)\frac{fs}{nfft}\}$
```

Figure A.1. Short-Time Fourier Transform Calculation.

APPENDIX B: PERFORMANCE TABLES

B.1. sAIQC

B.1.1. QLD Results

Table B.1. The results of SVM as sAIQC classifier trained with QLD.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	486	0	2	0	4	1	3	**TNR** 0.810
	Fail	114	0	5	0	43	1	12	**TPR** 0.859
		TNR 0.810	**TPR** -	**TPR** 0.714	**TPR** -	**TPR** 0.915	**TPR** 0.500	**TPR** 0.800	**UA** 0.835

Table B.2. The results of XGB as sAIQC classifier trained with QLD.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	468	0	2	0	2	1	4	**TNR** 0.777
	Fail	132	0	5	0	45	1	11	**TPR** 0.873
		TNR 0.777	**TPR** -	**TPR** 0.714	**TPR** -	**TPR** 0.957	**TPR** 0.500	**TPR** 0.733	**UA** 0.825

Table B.3. The results of MLP as sAIQC classifier trained with QLD.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	404	0	0	0	1	1	1	**TNR** 0.676
	Fail	196	0	7	0	46	1	14	**TPR** 0.955
		TNR 0.676	**TPR** -	**TPR** 1.000	**TPR** -	**TPR** 0.979	**TPR** 0.500	**TPR** 0.933	**UA** 0.816

B.1.2. QULD Results

Table B.4. The outputs of SVM trained with QULD in sAIQC.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	19828	5	20	2	11	50	46	**TNR** 0.787
	Fail	5352	60	38	3	72	60	33	**TPR** 0.665
		TNR 0.787	**TPR** 0.923	**TPR** 0.655	**TPR** 0.600	**TPR** 0.867	**TPR** 0.545	**TPR** 0.418	**UA** 0.726

Table B.5. The outputs of RF trained with QULD in sAIQC.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	23104	12	37	4	11	71	62	**TNR** 0.918
	Fail	2076	53	21	1	72	39	17	**TPR** 0.508
		TNR 0.918	**TPR** 0.815	**TPR** 0.362	**TPR** 0.200	**TPR** 0.867	**TPR** 0.355	**TPR** 0.198	**UA** 0.713

Table B.6. The outputs of MLP trained with QULD in sAIQC.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	17947	6	19	3	10	37	42	**TNR** 0.713
	Fail	7233	59	39	2	73	73	37	**TPR** 0.708
		TNR 0.713	**TPR** 0.908	**TPR** 0.672	**TPR** 0.400	**TPR** 0.880	**TPR** 0.664	**TPR** 0.468	**UA** 0.710

B.2. dAIQC

B.2.1. QLD Results

Table B.7. The outputs of SVM ,trained with QLD, in the first stage of dAIQC.

		Actual		
		Proper	**Improper**	**Performance metrics**
Predicted	**Proper**	427	55	**TNR** 0.912
	Improper	41	148	**TPR** 0.729
		TNR 0.912	**TPR** 0.729	**UA** 0.821

Table B.8. The outputs of XGB ,trained with QLD, in the first stage of dAIQC.

		Actual		
		Proper	**Improper**	**Performance metrics**
Predicted	**Proper**	455	63	**TNR** 0.972
	Improper	13	140	**TPR** 0.690
		TNR 0.972	**TPR** 0.690	**UA** 0.831

Table B.9. The outputs of MLP ,trained with QLD, in the first stage of dAIQC.

		Actual		
		Proper	**Improper**	**Performance metrics**
Predicted	**Proper**	414	47	**TNR** 0.885
	Improper	54	156	**TPR** 0.768
		TNR 0.885	**TPR** 0.768	**UA** 0.827

Table B.10. The prediction outcomes of SVM, trained with QLD, as Pass & Fail Classifier 1.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	460	0	2	0	1	1	1	**TNR** 0.968
	Fail	15	0	2	0	21	1	12	**TPR** 0.878
		TNR 0.968	**TPR** -	**TPR** 0.500	**TPR** -	**TPR** 0,955	**TPR** 0.500	**TPR** 0.923	**UA** 0.923

Table B.11. The prediction outcomes of RF, trained with QLD, as Pass & Fail Classifier 1.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	462	0	2	0	3	1	5	**TNR** 0.973
	Fail	13	0	2	0	19	1	8	**TPR** 0.732
		TNR 0.973	**TPR** -	**TPR** 0.500	**TPR** -	**TPR** 0.863	**TPR** 0.500	**TPR** 0.615	**UA** 0.853

Table B.12. The prediction outcomes of XGB, trained with QLD, as Pass & Fail Classifier 1.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	472	0	2	0	0	1	5	**TNR** 0.994
	Fail	3	0	2	0	22	1	8	**TPR** 0.805
		TNR 0.994	**TPR** -	**TPR** 0.500	**TPR** -	**TPR** 1.000	**TPR** 0.500	**TPR** 0.615	**UA** 0.899

Table B.13. The prediction outcomes of SVM, trained with QLD, as Pass & Fail Classifier 2.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	117	0	0	0	2	0	0	**TNR** 0.936
	Fail	8	0	3	0	23	0	2	**TPR** 0.933
		TNR 0.936	**TPR** -	**TPR** 1.000	**TPR** -	**TPR** 0.920	**TPR** -	**TPR** 1.000	**UA** 0.935

Table B.14. The prediction outcomes of RF, trained with QLD, as Pass & Fail Classifier 2.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	124	0	0	0	3	0	1	**TNR** 0.992
	Fail	1	0	3	0	22	0	1	**TPR** 0.867
		TNR 0.992	**TPR** -	**TPR** 1.000	**TPR** -	**TPR** 0.880	**TPR** -	**TPR** 0.500	**UA** 0.929

Table B.15. The prediction outcomes of XGB, trained with QLD, as Pass & Fail Classifier 2.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	124	0	0	0	3	0	0	**TNR** 0.992
	Fail	1	0	3	0	22	0	2	**TPR** 0.900
		TNR 0.992	**TPR** -	**TPR** 1.000	**TPR** -	**TPR** 0.880	**TPR** -	**TPR** 1.000	**UA** 0.946

B.2.2. QULD Results

Table B.16. The prediction results of SVM as Pass & Fail Classifier 1 of dAIQC for QULD.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	16266	3	11	1	5	11	4	**TNR** 0.963
	Fail	631	53	28	0	72	44	45	**TPR** 0.874
		TNR 0.963	**TPR** 0.946	**TPR** 0.718	**TPR** 0.000	**TPR** 0.935	**TPR** 0.800	**TPR** 0.918	**UA** 0.918

Table B.17. The prediction results of RF as Pass & Fail Classifier 1 of dAIQC for QULD.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	16683	10	30	1	7	40	19	**TNR** 0.987
	Fail	214	46	9	0	70	15	30	**TPR** 0.614
		TNR 0.987	**TPR** 0.821	**TPR** 0.231	**TPR** 0.000	**TPR** 0.909	**TPR** 0.272	**TPR** 0.612	**UA** 0.801

Table B.18. The prediction results of XGB as Pass & Fail Classifier 1 of dAIQC for QULD.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	15866	3	8	1	7	18	12	**TNR** 0.939
	Fail	1031	53	31	0	70	37	37	**TPR** 0.823
		TNR 0.939	**TPR** 0.946	**TPR** 0.795	**TPR** 0.000	**TPR** 0.909	**TPR** 0.673	**TPR** 0.755	**UA** 0.881

Table B.19. The prediction results of SVM as Pass & Fail Classifier 2 of dAIQC for QULD.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	7388	3	9	2	3	11	11	**TNR** 0.892
	Fail	994	6	10	2	3	44	19	**TPR** 0.683
		TNR 0.892	**TPR** 0.666	**TPR** 0.526	**TPR** 0.500	**TPR** 0.500	**TPR** 0.800	**TPR** 0.633	**UA** 0.788

Table B.20. The prediction results of RF as Pass & Fail Classifier 2 of dAIQC for QULD.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	7947	8	17	3	6	39	26	**TNR** 0.959
	Fail	336	1	2	1	0	16	4	**TPR** 0.195
		TNR 0.959	**TPR** 0.111	**TPR** 0.105	**TPR** 0.250	**TPR** 0.000	**TPR** 0.291	**TPR** 0.133	**UA** 0.577

Table B.21. The prediction results of XGB as Pass & Fail Classifier 2 of dAIQC for QULD.

		Actual							
		Pass	**Fail**						
		Pass	**Vib1**	**Vib2**	**Vib3**	**Vib1&2**	**Vib**	**Sound**	**Performance metrics**
Predicted	**Pass**	7167	5	13	2	4	21	15	**TNR** 0.865
	Fail	1116	4	6	2	2	34	15	**TPR** 0.512
		TNR 0.865	**TPR** 0.444	**TPR** 0.316	**TPR** 0.500	**TPR** 0.333	**TPR** 0.618	**TPR** 0.500	**UA** 0.689

APPENDIX C: TEZDE KULLANILAN ŞEKİLLER İÇİN İZİN

İlgili izin Figure 2.1 için alınmış olup geriye kalan figürlerin tamamı tarafımca oluşturulan altyapıya dair çekilmiş görüntüleri ve çizilmiş grafikleri içermektedir.

05.08.2021

İlgili Makama

Çalışanımız Sibel Şentürk'ün Boğaziçi Üniversitesi Elektrik Elektronik Mühendisliği yüksek lisans tezi kapsamında kendisine sağlamış olduğumuz BLDC motor datasetlerine ait içerik tablolarının ve veriye ait yapılan görselleştirmelerin, aynı zamanda motora ait parçaların görselinin(Fig2.1) kullanılmasına tez yazımı kapsamında izin verilmiştir.

Turgut Köksal Yalçın

Arçelik Üretim Teknolojileri Direktörlüğü

Endüstriyel Robotik Yöneticisi