# OIL CONDITION PREDICTION IN DIESEL GENERATORS USING ACOUSTIC SIGNALS

W. A. Hiran Wijesuriya

208810B

Master of Science in Industrial Automation

Department of Electrical Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

September 2025

# OIL CONDITION PREDICTION IN DIESEL GENERATORS USING ACOUSTIC SIGNALS

W. A. Hiran Wijesuriya

208810B

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree
Master of Science in Industrial Automation

Department of Electrical Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

September 2025

# DECLARATION

I declare that this is my own work, and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

09.09.2025

……………………………………….. ………………………………………..

Signature Date

The above candidate has carried out research for the Master's thesis/dissertation under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of Supervisor: **Prof. Buddhika Jayasekara**

09/09/2025

……………………………………......... ………………………………………..

Signature of the Supervisor Date

# ACKNOWLEDGEMENT

First, my sincere thanks must go to my advisor, Prof. Buddhika Jayasekara, for his continuous advice, guidance, encouragement and patience throughout the course of this work. It has been a privilege to work under his guidance. I am also thankful to the course coordinator, Prof. Buddhika Jayasekara and the staff of the Department of Electrical Engineering, University of Moratuwa, for their continuous encouragement.

Furthermore, I extend my sincere gratitude to UTE CAT for their support in the collection of materials and data essential to this thesis. Finally, I am deeply thankful to my parents, whose unwavering belief in my potential has continually inspired me to strive beyond my own expectations.

W. A. Hiran Wijesuriya
208810B
University of Moratuwa.

# ABSTRACT

This research proposes a novel method for predicting the oil condition in CAT diesel generators using acoustic signals, aiming to overcome the limitations of traditional Scheduled Oil Sampling (S·O·S$^{SM}$) testing methods. Conventional S·O·S$^{SM}$ programs often face challenges such as delayed results, sampling errors, and high operational costs. To address these, the study develops a non-invasive acoustic monitoring technique that correlates generator sound patterns with oil health indicators.

Audio samples were collected using a Blue Yeti microphone under controlled conditions, and corresponding physical oil samples were subjected to standard S·O·S$^{SM}$ laboratory analysis. Various time-domain and frequency-domain features, including MFCCs, Zero Crossing Rate, and Spectral Centroid, were extracted using the Librosa library. A neural network model was developed to classify oil condition states into "Action Required," "No Action Required," or "Monitor," based on these acoustic features.

Testing the system on a dataset of 75 samples showed encouraging results, with the second iteration achieving improved classification accuracy. Findings confirm that generator acoustic emissions change with oil degradation, validating the hypothesis that sound can predict oil condition. Conclusions highlight that this method enables timely, cost-effective oil monitoring without operational disruption.

Recommendations for future work include optimizing the neural network through learning rate adjustments, particularly expanding the sample dataset, exploring ensemble methods, and validating performance across different generator types to enhance system robustness and industrial applicability.

**Keywords**: Oil Condition  Prediction, Diesel Generators, Time – domain, Frequency – domain, Machine Learning.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| Abbreviation | Description |
|---|---|
| S·O·S$^{SM}$ | Scheduled Oil Sample |
| MFCCs | Mel Frequency Cepstral Coefficients |
| FT-IR | Fourier Transform Infrared |
| TBN | Total Base Number |
| OEM | Original Equipment Manufacturer |
| ISO | International Organization for Standardization |
| TUT | Tampere University of Technology |
| DCAMS | Detection and Classification of Anomalous Machine Sound |
| BSS | Blind Source Separation |
| MC-FIR | Multichannel Finite Impulse Response |
| LPF | Low Pass Filter |
| IIR | Infinite Impulse Response |
| GKF | General Kalman Filter |
| AE | Amplitude Envelope |
| RMSE | Root Mean Square Energy |
| ZCR | Zero Crossing Rate |
| BER | Band Energy Ratio |
| SC | Spectral Centroid |
| BW | Bandwidth |
| SS | Spectral Spread |
| ML | Machine Learning |
| MRMR | Minimum Redundancy Maximum Relevance |
| FDR | Fisher's Discriminant Ratio |

| | |
|---|---|
| MLP | Multi-Layer Perceptron |
| ReLU | Rectified Linear Unit |
| NN | Neural Network |
| RBFNN | Radial Basis Function Neural Network |
| CNN | Convolutional Neural Network |
| ANN | Artificial Neural Network |
| BN | Bayesian Network |
| SVM | Support Vector Machine |
| FL | Fuzzy Logic |
| MF | Membership Function |
| FIS | Fuzzy Interference System |
| PdM | Predictive Maintenance |
| FCM | Fuzzy C – Means |
| OCSVM | One Class Support Vector Machine |
| RUL | Remaining Useful Life |
| RNN | Recurrent Neural Network |
| SVR | Support Vector Regression |
| FFT | Fast Fourier Transformation |

# LIST OF APPENDICES

# CHAPTER 1

# INTRODUCTION

## 1.1. Background of the study

The diesel generators are essential in providing reliable power supply in a wide range of applications, ranging from industries, telecommunications, and power support systems during emergency situations. The working and lifespan of the generators are mainly dependent on the status of the components, especially the lubricating oil, which happens to be among the most critical. Over time, the oil degrades due to a combination of thermal stress, contamination, and oxidation. The breakdown in oil can lead to increased friction, accelerated wear and tear of engine parts, and, if unchecked, can lead to massive engine failure.

Traditional maintenance approaches to generators mainly rely on preventive and corrective procedures. These methods are usually developed for a long time or usage parameters, which means maintenance actions are scheduled at fixed intervals, often making them ineffective. Such an approach can lead to two unsavory consequences: either the oil is changed unnecessarily, leading to wasted expenses, or it is run farther than it can function, creating a serious threat of serious damage to the generator. The modern trend towards more smart and effective asset management requires a move from such time-dependent approaches in favor of condition-based and predictive maintenance procedures. Such a new paradigm is based on data-driven approaches in determining the most opportune time for maintenance, thereby increasing the useful lives of parts while, at the same time, cutting down on operational downtime and costs.

When discussing condition-based preventive maintenance on generator, Scheduled Oil Sampling (S·O·S) is essential. This S·O·S is a tested technique applied in generator and heavy equipment maintenance to monitor equipment and oil quality. Periodic sampling or S·O·S testing assists in establishing if the oil is in a fit state or if it has to be replaced. Essentially, S·O·S testing monitors oil viscosity, oxidation, and acid levels over a period of time. Additionally, it stretches oil change intervals if the oil remains in a good state, thereby saving money. Consequently, S·O·S prevents unwarranted oil changes, saving on maintenance, and waste and promoting timely oil replacement before oil inflicts damage.

Despite the absence of real-time capability, this classical method has gained widespread recognition by virtue of its proved capability in early warning of issues as well as maintenance scheduling enhancement.

## 1.2. Problem Statement

The primary issue in achieving a true predictive maintenance system of diesel generator oil lies in the development of a non-intrusive, dependable method for continuous monitoring of its condition. Available methodologies tend to be intrusive, both labor and expense, and possibly disrupting generator function. As an example, S·O·S testing delivers periodic, and not continuous, monitoring, so problems are merely identified on the rare occasion of sampling. Furthermore, equipment could develop a critical fault during intervals or gaps in sampling without giving warning. As a direct effect, there exists a possibility of surprising breakdowns despite frequent testing, as well as a higher likelihood of catastrophic failure in the event a problem increases exceedingly rapidly. Conversely, the effectiveness of S·O·S testing significantly depends on adequate sampling techniques; mistakes could originate from dirty sampling gear or cups, improper sampling protocols, and human errors during sample handing or labeling. These could cause misplaced results or inferences related to conditions in the oil. Accordingly, there exists a particular need for a new approach capable of monitoring oil deterioration in real-time without a need for touch or disruption. Conversely, the implementation of real-time monitoring provides predictive maintenance, reduces oil usage, and augments overall system reliability, which make it a cornerstone part for modern-day generator administration and running efficiency. This study thus aims at meeting the need by considering the potentiality in exploiting the use of acoustic signals, which are inevitably produced during the running of a diesel engine, as a predictor of oil condition. The primary issue lies in the development of a robust machine learning model capable of accurately labeling the status of the oil (i.e., healthy, degraded) from characteristics extracted from these acoustic signatures.

## 1.3. Significance of the Study

The value of this work is multifaceted, having both practical and intellectual value. From a practical perspective, the development of a predictive model based on acoustic signals provides a low-cost and non-intrusive way to monitor the health of diesel generators. This study makes condition-based oil changes possible, which could lead to considerable maintenance costs savings, reduced oil consumption and waste, and a considerable reduction in unexpected mechanical breakdowns. The ability to predict maintenance is going to increase system reliability and efficiency in operations, particularly in critical situations whereby system downtime is unacceptable. Also, it would be observed that, because of lack of real-time data, maintenance groups are forced into following reactive or scheduled maintenance procedures. Therefore, oil changes are made too soon, thus incurring wasteful costs, or too late, thus increasing the possibility of equipment breakdown or disastrous failure. Essentially, maintenance decisions are made on assumptions and not on real oil conditions.

This thesis academically enhances the expanding corpus of knowledge pertaining to predictive maintenance and the implementation of machine learning within the realm of industrial automation. It investigates a distinctive and relatively unexplored modality—acoustic data—in relation to a prevalent and significant diagnostic endeavor. The methodology formulated, which employs a Multi-layer Perceptron (MLP) Classifier to analyze the intricate patterns present in acoustic signals, has the potential to function as a fundamental framework for subsequent investigations into acoustic-based condition monitoring applicable to various types of machinery.

## 1.4. Thesis outline

As already stated, fluid analysis programs are critical in the quality maintenance of lubricants in ensuring machinery reliability and averting malfunctioning. The programs comprise performance testing and oil condition monitoring in a bid to diagnose contamination, degradation, and additive depletion, among causes of machinery breakdowns. Enhanced methods, namely FT-IR spectroscopy and in-line sensors, have been created for expert fault detection and measurement in selected oil-related faults. This requires a comprehensive understanding of failure modes relevant to the particular system, how they develop, and measurement methodologies applied in condition monitoring.

Fluid analysis plays a vital role in modern equipment maintenance as the predictive information gained through such services curtails breakdowns and enhances effective operation through reduction in unnecessary costs. This research is intended to investigate Cat® S·O·S℠ Services, a respected global fluid analysis program that has been operating successfully for more than half a century. The study shows, through a review of past data, working practices, and economical effects, how fluid analysis can revamp equipment maintenance strategies by ensuring early detectability of potential breakdowns, effective distribution of resources, and enhancement in machinery lifespan.

Fluid analysis could easily be considered the most influential predictor of the condition of a machine's major components. With Cat S·O·S℠ Services, advance diagnostic technology is put together in a comprehensive and integrated format, aided by skilled interpretation in order to allow customers to make maintenance choices based on facts. This research explores exhaustively the various working, economic, and strategic benefits involved with fluid analysis, highlighting how it is especially crucial in cutting down operational costs, promoting machine reliability, and increasing the lifespan of parts.

The main objective of this research is to develop a method to determine generator oil condition by using acoustic signals as given by the S·O·S testing, “Action Required”, “No Action Required” or “Monitor”. By extracting noise signals associated with oil condition and time and frequency domain spectrum related features, research is in line for oil condition and acoustic pattern correlating. Besides, based on extracted features, a predicting system based on machine learning shall be formulated in order to enable efficient and accurate oil health monitoring in generators. The method is a data-based and non-invasive maintenance predicting method targeted at reliability and performance improvement in generator systems.

To achieve the latter, the study shall strive to meet the following particular aims:

- To investigate the relationship between acoustic signatures and the degradation state of lubricating oil in a diesel generator.
- To develop an effective methodology for capturing, processing, and extracting relevant features from the acoustic signals.
- To train and evaluate a machine learning model, specifically a Multi-layer Perceptron Classifier, to accurately predict oil condition.
- To validate the performance of the proposed model using standard classification metrics such as accuracy, precision, and recall.

The remaining of the thesis is ordered as follows. Chapter 2 presents a detailed description of acoustic signal base condition monitoring and Oil condition monitoring, and comprehensive literature which is used in preventive maintenance. Chapter 3 presents the methodology proposed for this research to predict oil condition. Chapter 4 presents results obtained by proposed method and discussion about obtained results. Conclusion provides the findings highlight the correlation between acoustic emission patterns from the generator and oil condition. Also, about machine learning based predictive model and its effectiveness in monitoring oil condition with regards to S·O·S testing with improved accuracy. Finally, the recommendations for further improve the effectiveness and applicability of acoustic signal-based oil condition, ensuring enhanced reliability in generator maintenance strategies.

# CHAPTER 2

# LITERATURE REVIEW

Condition monitoring plays a major role in preventive maintenance, among condition monitoring Oil Condition Monitoring has a major role. There are real time faults prediction and oil condition prediction methods using advanced sensor mechanisms. A review of proposed acoustic signal – based oil condition prediction, major challenges are discussed in chapter 2.

## 2.1. Fluid Analysis Programme

As a result of evolution of the Fluid Analysis Programme, at present it can be categorized into three major sections, coolant analysis, fuel analysis and lube oil analysis. In each section consist of different kind of testing methods according to the requirements. When it comes to the lube oil analysis, below are the three main testing methods which are used in lube oil analysis,

i. Total Base Number (TBN)
   TBN is the degree of an oil's alkalinity, reflecting the ability to neutralize acidic contaminants. TBN monitoring is, therefore, essential to determine whether the oil is degraded with respect to protection against corrosion in engine components.

ii. Kinematic Viscosity
   This parameter reflects the oil's flow resistance under specified temperatures, directly impacting lubrication efficiency. Changes in viscosity can signal contamination, thermal degradation, or incorrect oil application, affecting equipment performance and longevity.

iii. Elemental Analysis
   Elemental analysis quantifies trace metal concentrations within the lubricant to indicate wear patterns, contamination levels, and additive depletion. The data is used in the diagnosis of early mechanical failure and maintenance optimization.

### 2.1.1 History of Fluid Analysis Programme

Fluid analysis programs have their roots in the early 20$^{th}$ century when industries started recognizing the importance of monitoring lubrication systems in machinery. These programs have transformed over the decades with advances in technology and a heightened emphasis on predictive maintenance into key tools for equipment health management. The Evolution and Impact of Oil Condition Monitoring from Early Developments to Modern Applications as follows,

i. Early Developments:

1900s-1950s: The concept of lubrication oil monitoring originated in industries like aviation and marine, where the reliability of the engine was crucial. Basic methods focus on visual inspection and rudimentary tests for viscosity and contamination.

1960s-1970s: With advancements in analytical instruments like spectrometers, fluid analysis became more precise. Industries such as mining and construction adopted these techniques to monitor wear metals and contaminants in oil.

ii. Institutional Adoption:

Decades of Expertise in Fluid Analysis - In 1968 Established, Cat® S·O·S℠ Services has become the largest OEM - operated fluid analysis program worldwide, offering expertise in oil, coolant, and diesel fuel diagnostics. With ISO 9001 – certified laboratories and a network of over 90 dealer labs, the program delivers fast, reliable results, enabling precise failure prediction and timely interventions. This study investigates how the program's historical advancements in fluid diagnostics contribute to its unparalleled accuracy in identifying and mitigating component risks.

1980s: Major machinery manufacturers such as Caterpillar, Komatsu, and others began to formalize fluid analysis programs. CAT introduced the S·O·S℠ program, which set industry standards for scheduled oil testing and trending.

ASTM and ISO Standards: Development of standards for fluid sampling and analysis during this period helped ensure consistency and reliability in testing methods.

iii. Technological Evolution:

1990s: Portable analytical devices and computerized systems allowed on-site testing, thus enabling faster decision-making. Data from fluid analysis began to integrate with maintenance software, marking the rise of predictive maintenance systems.

2000s-P Present: Modern fluid analysis makes use of advanced techniques such as infrared spectroscopy, particle count analysis, and wear debris analysis. Programs now include oil, coolant, and fuel testing, offering a holistic view of machinery health.

iv. Current Applications:

Fluid analysis programs, such as Cat® S·O·S℠, now play a critical role in predictive maintenance strategies. They are used across industries to extend equipment life, reduce operating costs, and improve efficiency and reliability.

### 2.1.2. Key Benefits of CAT S·O·S Services

An oil analysis program offers enormous advantages in energy, railway, sea transport, and aviation, among other areas. It basically enables predictive maintenance; hence, less unplanned downtime, fewer maintenance costs, more reliability, and higher productivity. The program offers early detection of impending failures, thus preventing them from causing secondary damage. With data-driven decisions, it optimizes maintenance scheduling and prolongs the service life of equipment. This application of oil analysis by the Spanish National Railway Company exemplifies such benefits, which in turn underscore that improvements in diagnostics and expert systems may have great economic and operational advantages [4]. Key benefits include [2],

i. Targeted Component Replacement
The ability to isolate faults to specific components – such as valves or cylinders – reduces the need for extensive repairs or system replacements. This precision leads to significant cost savings and shorter downtime, enabling operators to maintain productivity while addressing maintenance issues proactively.

ii. Scheduled Maintenance for Downtime Reduction
Unplanned repairs often disrupt operational schedules and incur additional costs. By predicting potential failures, fluid analysis facilitates maintenance planning that aligns with operational demands, allowing for repairs during non-peak periods or when alternative equipment is available.

iii. Extended Oil Drain Intervals
Optimizing oil drain intervals through fluid analysis reduces maintenance frequency, conserving resources and lowering costs. Collaborative efforts with equipment dealerships to establish and monitor new intervals ensure engine reliability and enhance fleet-wide cost efficiency.

iv. Maximized Component Lifespan
Condition-based maintenance enabled by fluid analysis ensures that components are replaced only when necessary, optimizing their usage. This approach minimizes service disruptions and prevents premature disposal of functional parts, maximizing return on investment.

v. Enhanced Resale Value through Service Histories

A detailed service history, backed by regular fluid analysis, provides prospective buyers with insights into equipment health and maintenance practices. This transparency boosts resale value by instilling confidence in the machine's operational reliability.

## 2.2. Caterpillar Oil Analysis Programme (CAT S·O·S Services)

The Caterpillar S·O·S$^{SM}$ Program is a structured approach to monitoring the health of your equipment through oil analysis. It would definitely help to detect wear, contamination, and improper maintenance early on. Here's how this procedure can be implemented step by step:

### 2.2.1. Oil Sample Collection

i. Use Correct Equipment: Use sampling kits as specified by Caterpillar, including sample bottles, tubing, and pumps as shown in Figure 2.1.


**Figure 2.1:** CAT Specified Sampling Kit for Oil Sample Collection

*Note:* 1 – Bottle Group, 300 per box, 2 – Probe, Holder, 3 – Cap and Probe Group, 500 per box, 4 – Mailer Container – reusable, 5 – Tube – 6.35 mm (1/4 in) O.D. * 30.5 m (100 ft) rolls, 5 rolls per box, 6 – Tube – 7.9 mm (5/16 in) O.D. * 30.5 m (100 ft) rolls, 5 rolls per box, ,7 – Clear Bottle, 71g (2 ½ oz), 200 per box, 8 – Clear Bottle, 114g (4 oz), 200 per box, 9 – Vacuum Pump & Seal Kit for Pump, 10 – Probe, purging, 11 – Tube Cutter & Replacement Blades, 12 – Sealed Cap (for temporary sealing of leaking sample valve), 13 – Oil Sampling Valve (7/16 inch – 20 – ext. thread) & O – Ring Seal, 14 – Oil Sampling Valve 91/4 inch – 18 NPTF – ext. thread), 15 – Oil Sampling Valve (M10*1 – ext. thread) & O – Ring Seal, 16 – Dust

ii. Warm Engine/Equipment: The machine should be at operating temperature to take a representative oil condition sample.

iii. Sample Location: Draw oil from the designated sampling port to avoid contamination. Do not take the sample from the oil pan or drain plug unless specified.

Sampling valves - Sampling valves are generally found on stationary engines. O-rings in the valves prevent metal-to-metal contact, that cause contamination. When a sampling valve is utilized, enough oil needs to be drawn into it to flush out the gauge line or oil gallery in which it is mounted. It provides a representative sample of the oil in the compartment. [1]

iv. Mid – Stream Collection: Take the sample midstream to avoid sediments or floating particles.

v. Label Accordingly: Label the sample with the machine details, hours of operation, and sample date.

### 2.2.2. Sample Documentation

i. During the sampling process fill out the associated documentation, as shown in Figure 2.2, with information such as: Machine make, model, serial number, grade and brand of oil applied, hours on the machine since last oil change and sampling, ambient conditions of operation, any observed changes in equipment performance etc.

<table>
<tr><td colspan="2">01(a). Customer -UTE<br>:</td><td colspan="2">01(b). Date : 2023-05-17</td><td colspan="2">01(c). Date of Last Analysis :<br>NA</td></tr>
<tr><td colspan="6">02. Reason for Analysis / Special Details (Defects/ machinery failure etc): To monitor oil condition.</td></tr>
<tr><td colspan="6">03. Machinery Name : Generator</td></tr>
<tr><td colspan="6">04. Machinery Make and Model : CAT C13</td></tr>
<tr><td colspan="6">05. Machinery Sr. No. : NH300786</td></tr>
<tr><td colspan="2">06(a). Sampling Point : Sampling valve</td><td colspan="4">06(b). Sampling date: 2023-05-15</td></tr>
<tr><td colspan="6">07. Lubricant (Type / Make / Grade): CAT 15W 40</td></tr>
<tr><td colspan="6">08. System Capacity : 60 L</td></tr>
<tr><td colspan="6">09. Oil added since last oil change (Ltrs) : NA</td></tr>
<tr><td colspan="2">10 (a). Oil running hrs (Operated duration since last oil change): 216</td><td colspan="4">10 (b). Date last oil changing routine carried out :<br>2022-08-18</td></tr>
<tr><td colspan="6">11. Total R/H of the Machinery : 1311</td></tr>
<tr><td colspan="2">12 (a). After Major repair /<br>After major routine R/H : NA</td><td colspan="4">12 (b). Date of Major Routine/ Repairs carried out :<br>NA</td></tr>
<tr><td colspan="6">13. Average Monthly R/H (for minimum 06 months operation) : -</td></tr>
<tr><td colspan="6">14. Oil Batch No. :</td></tr>
<tr><td colspan="6">15. Name / Contact number of a responsible person :<br>……………………………………</td></tr>
<tr><td rowspan="3">16.Required Tests (*optional):</td><td>Kinematic Viscosity</td><td>*</td><td>Wear Metal Analysis</td><td>*</td><td></td></tr>
<tr><td>Total Base Number</td><td>*</td><td>Water Content</td><td></td><td></td></tr>
<tr><td>Flash Point</td><td></td><td>Insoluble Test</td><td></td><td></td></tr>
<tr><td colspan="2">17. Repairs/Routines attended since last oil analysis report:</td><td colspan="4"></td></tr>
</table>

**Figure 2.2:** Sample Details Report

### 2.2.3. Laboratory Analysis

Samples are sent to an authorized Caterpillar S·O·S$^{SM}$ Services Laboratory or a qualified lab for testing. The lab performs key analyses, including:

i. Total Base Number (TBN): Determines the oil's ability to neutralize acids. Monitors oil degradation and the need for an oil change.

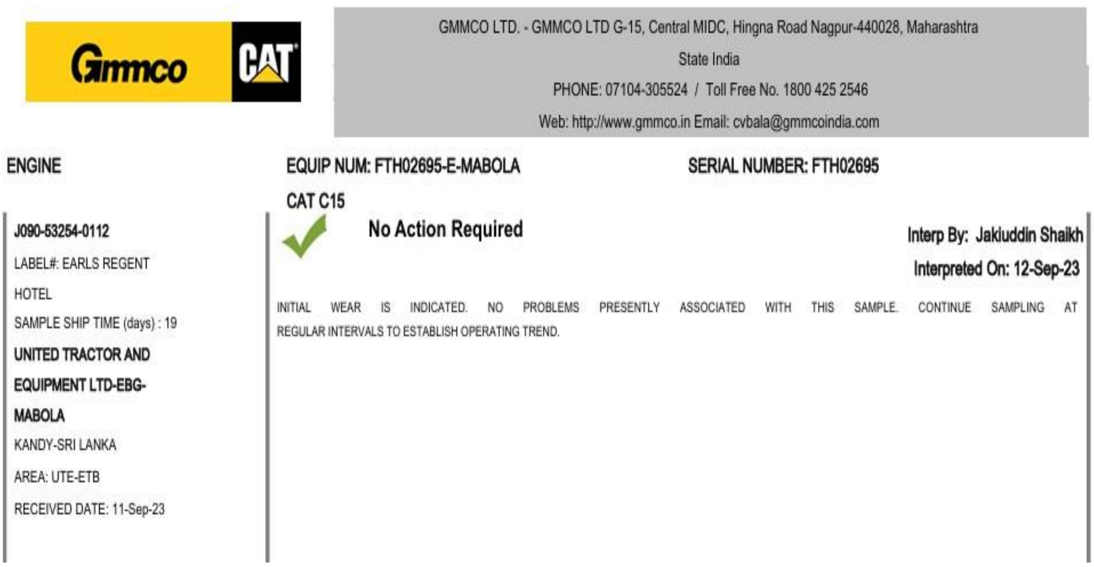
ii. Kinematic Viscosity: Measures the oil's flow resistance at standard temperatures (typically 40°C and 100°C). Identifies oil thinning or thickening due to contamination or degradation.

iii. Elemental Analysis: Provides results for wear metals (iron, copper, aluminum), contaminants (silicon, sodium), and additive elements (zinc, phosphorus). Determine if wear, contamination or additive depletion may be occurring in any component.

Other than the above key tests, the tests below, such as Water Content Testing (Quantifies coolant leakage or condensation), Fuel Dilution (Fuel contamination of oil, which may lower viscosity) and Oxidation/Nitration (Oil degradation due to heat and combustion gases) also conducted according to the requirements.

### 2.2.4. Laboratory Analysis

Flagged values or values out of the normal range should be identified in the lab report. Trend analysis and maintenance recommendations are provided in every CAT S·O·S$^{SM}$ report. Status of the S·O·S$^{SM}$ results are categorized under three areas, Action Required, No Action Required and Monitor as shown in Figure 2.3.


Gmmco CAT

GMMCO LTD. - GMMCO LTD G-15, Central MIDC, Hingna Road Nagpur-440028, Maharashtra
State India
PHONE: 07104-305524 / Toll Free No. 1800 425 2546
Web: http://www.gmmco.in Email: cvbala@gmmcoindia.com

ENGINE

EQUIP NUM: FTH02695-E-MABOLA

CAT C15

SERIAL NUMBER: FTH02695

J090-53254-0112
LABEL#: EARLS REGENT HOTEL
SAMPLE SHIP TIME (days) : 19
UNITED TRACTOR AND EQUIPMENT LTD-EBG-MABOLA
KANDY-SRI LANKA
AREA: UTE-ETB
RECEIVED DATE: 11-Sep-23

No Action Required

Interp By: Jakiuddin Shaikh
Interpreted On: 12-Sep-23

INITIAL WEAR IS INDICATED. NO PROBLEMS PRESENTLY ASSOCIATED WITH THIS SAMPLE. CONTINUE SAMPLING AT REGULAR INTERVALS TO ESTABLISH OPERATING TREND.

**Figure 2.3:** CAT Lab S·O·S$^{SM}$ Report – Summary Information

| SAMPLE INFORMATION | |
|---|---|
| Sampled Date | 23-Aug-23 |
| Sample Id | J090-53254-0112 |
| Lab Date | 11-Sep-23 |
| Meter [Hr] | 775 |
| Comp Meter [Hr] | 775 |
| Meter On Fluid | 636 |
| Fluid Brand | CAT |
| Fluid Weight | 15W-40 |
| Fluid Type | DEO |
| Fluid Change | U |
| Filter Change | U |
| Kidney Loop | U |
| Total Fluid Added | 0 |

PREVIOUS SAMPLE

For additional sample history, go to: my.cat.com

**CONDITION / CONTAMINATION**

23-Aug-23

| VISCOSITY (CENTISTOKES) ASTM D445 | | |
|---|---|---|
| V100 | Viscosity at 100 C | 13.12 |

| INFRARED (UFM) ASTM E2412 | | |
|---|---|---|
| ST | Soot | 13 |
| OXI | Oxidation | 7 |
| SUL | Sulfur Products | 15 |
| NIT | Nitration | 5 |

| WATER | | |
|---|---|---|
| W | Water | N |

| ANTIFREEZE | | |
|---|---|---|
| A | Antifreeze | N |

| FUEL | | |
|---|---|---|
| F | Fuel | N |

**CLEANLINESS**

23-Aug-23

| FERROUS DEBRIS | | |
|---|---|---|
| PQI | PQ Index | 4 |

**WEAR LEVELS / ADDITIVES**

23-Aug-23

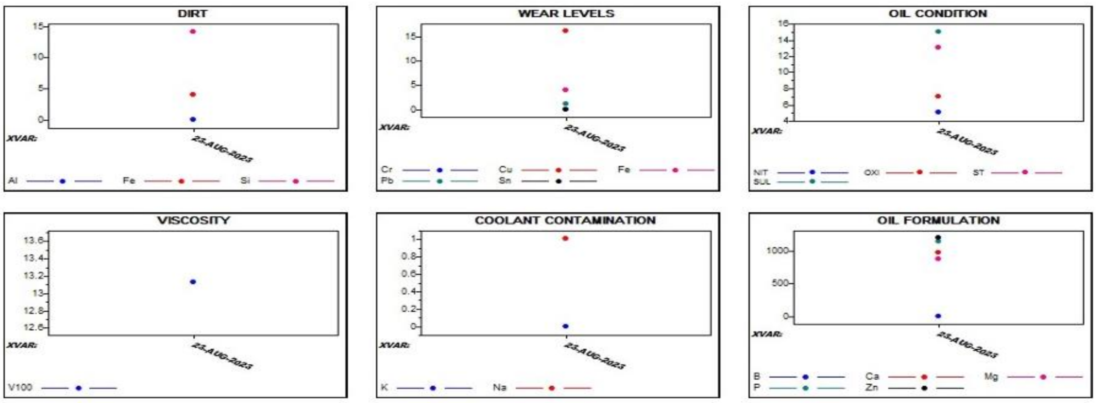
| ELEMENTAL ANALYSIS (PPM) ASTM D5185 [OIL] / ASTM D6130 [COOLANT] | | |
|---|---|---|
| Cu | Copper | 16 |
| Fe | Iron | 4 |
| Cr | Chromium | 0 |
| Al | Aluminum | 0 |
| Pb | Lead | 1 |
| Sn | Tin | 0 |
| Si | Silicon | 14 |
| Na | Sodium | 1 |
| K | Potassium | 0 |
| Mo | Molybdenum | 5 |
| Ni | Nickel | 0 |
| Ag | Silver | 0 |
| Ti | Titanium | 1 |
| V | Vanadium | 0 |
| S | Sulfur | 4401 |
| Ca | Calcium | 967 |
| P | Phosphorus | 1149 |
| Zn | Zinc | 1195 |
| Mg | Magnesium | 878 |
| B | Boron | 1 |

**Figure 2.4:** CAT Lab S·O·S$^{SM}$ Report – Test Information

*Note:* Test Information – Total Base Number, Kinematic Viscosity and Elemental Analysis issued by CAT Lab – Gmmco S·O·S$^{SM}$ Report

ENGINE
SERIAL NUMBER: FTH02695
EQUIP NUM: FTH02695-E-MABOLA
CAT C15

J090-53254-0112
No Action Required




**Report Comment**

Our sample reports has been updated! For more information on the new report, go to - https://www.youtube.com/watch?v=4h8bREJVUrs

**Figure 2.5:** CAT Lab S·O·S$^{SM}$ Report – Trend Analysis using Historical Data

### 2.2.5. Corrective Action Based on the Analysis

Corrective and preventive maintenance strategies are guided by the recommendations provided through Cat® S·O·S$^{SM}$ analysis. For instance, when significant oil degradation is detected, actions such as oil replacement or, in cases of high contamination, simultaneous oil and filter changes may be necessary. The elemental analysis, particularly of metal content (as shown in Figure 2.4), can help identify potential mechanical faults by revealing early signs of wear or failure. Additionally, periodic resampling plays a critical role in identifying abnormal trends in equipment performance, thereby enabling timely maintenance interventions and minimizing operational downtime.

### 2.2.6. Record Keeping

Keeping the history of oil analysis for a particular machine helps to predict trends of wear and schedule predictive maintenance. In the Caterpillar S·O·S$^{SM}$ programme they provide the historical data pattern as shown in Figure 2.5. Also, Caterpillar provides S·O·S$^{SM}$ results on a digital platform (as shown in Figure 2.4, under Previous Sample data) which includes equipment S·O·S$^{SM}$ history and its relevant details for easy monitoring and analysis.

## 2.3. Challenges of Traditional S·O·S Testing

While traditional S·O·S testing offers great insight into equipment health, it is not without its own set of limitations and challenges. If these are not identified and worked upon, they may affect the overall effectiveness of the program in the following ways:

i. Limited Real – Time Insights

   Lag Between Sampling and Results: Traditional S·O·S testing depends on periodic sampling, which can delay the detection of sudden failures or rapid degradation. Also, depending on the Lab facilities and their capacities sometimes results might take 2 – 3 weeks for results or it can be even more.

   Dynamic Equipment Changes: Fast evolving problems may not be detected until the next scheduled test, and the potential for equipment damage is increased.

ii. Dependence on Sampling Accuracy

   Human Error: Incorrect sampling practices, such as contamination at collection or incorrect labeling, can render the results invalid.

   Inconsistent Conditions of Sampling: Oil temperature, location, or timing may lead to unreliable or unrepresentative data. For instance, the machine should be at operating temperature to take a representative oil condition sample.

iii. Testing Coverage and Limitations

Focus on Known Parameters: While it identifies key issues like wear metals and viscosity Changes, traditional testing may miss other critical factors such as micro – contaminants or certain modes of failure.

Unable to Monitor Complex Interactions: Interactions between different contaminants or their combined effects on lubrication may not be fully captured.

iv. Time and Cost Factors

Delays in Report Turnaround: The need to send samples to laboratories and wait for results may introduce delays in decision – making. In the absence of certified facilities within the country, samples have to be sent to laboratories in other countries, which may lead to delays of 2 to 4 weeks depending on the prevailing circumstances.

Recurring Expenses: The routine samplings, shipping, and laboratory costs add up over time and may affect budgets.

v. Equipment Downtime

Operational Disruptions: Sampling often requires that equipment be taken offline, which affects productivity.

Logistical Issues: Poor accessibility of the sampling points on some equipment, mainly in isolated areas, is prevalent.

vi. Interpretation Challenges

Complex Data Interpretation: Understanding and implementing actions based on test results mostly requires specialized knowledge. Misinterpretations can result in incorrect maintenance actions.

Gaps in Trend Analysis: Meaningful trend analysis requires a historical dataset, which may not be available for new equipment or systems.

vii. Reactive Maintenance Approach

Delayed Action: Traditional S·O·S testing is inherently reactive; it addresses issues after they arise rather than providing predictive insights in real time.

Missed Predictive Opportunities: Without integration into predictive maintenance programs, potential failures might escalate before being addressed.

viii. Environmental and Safety Concerns

Hazardous Materials: Used oil samples are considered hazardous and require special handling and disposal methods.

Exposure Risks: There is a risk of exposure to hot or contaminated oil in the process of sample collection.

While traditional S·O·S testing remains a key tool for maintenance programs, its limitations must be addressed through the incorporation of newer technologies such as real – time sensors, advanced diagnostics, and predictive analytics. By overcoming these challenges, organizations can achieve more proactive and effective maintenance strategies. [3]

Based on the limitations on S·O·S testing as discussed above, mostly advanced sensor-based techniques are used in oil condition prediction. Although they possess benefits, sophisticated sensor – based oil condition monitoring methods are accompanied by colossal challenges such as cost, data intricacy, integration, and environmental sensitivity. For these challenges to be overcome, sensor calibration, better AI models, strong data management methods, and industrial standardization need to be introduced to ensure that oil degradation forecasts are accurate and reliable. This research proposed a method to predict oil condition using acoustic signals as a solution for the above-mentioned limitations.

## 2.4. Challenges of collecting acoustic signals from generators

When collecting Acoustic signals of equipment, there are key factors to consider.

i. Measurement Distance (Microphone Position)

Acoustic signal is a pressure wave that will propagate longitudinally through the air, suffering attenuation – that is, a loss in energy as it moves from source to observer. For the case of a point source radiating sound uniformly in all directions, this attenuation is governed by the inverse square law, with the intensity decreasing by a factor of four, or by 6 dB, as the distance from the source doubles. This demonstrates how measurement distance can be an important factor in the detection of anomalous machine sounds.

In acoustics, the near field, close to the source, does not have a well-defined relation between intensity and distance, whereas the far field, at least one wavelength from the source, obeys the inverse square law as shown in Figure 2.6. ISO 3745 provides requirements for microphone positions in the far field.

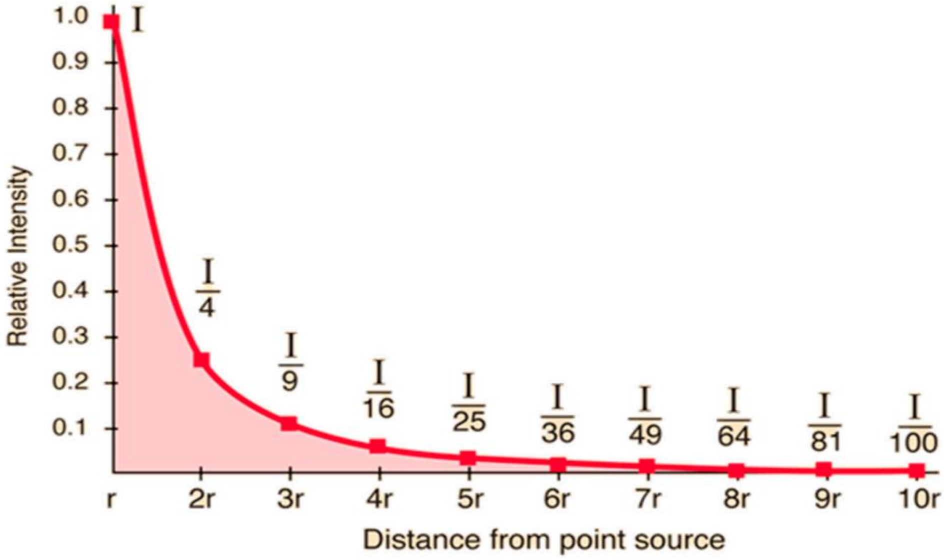



**Figure 2.6:** Distance effect on sound intensity propagation and attenuation [5]

For small, low-noise sources, it recommends distances of at least one meter with special requirements for measurements at shorter distances. Further research and detailed guidelines are needed for an accurate anomalous sound measurement, especially in terms of the effects of measurement distance. [5]

ii. Single Microphone Measurement Device

The selection on "Single Micrograph Measurement Device and Sampling Parameters" points out the problem of mismatching acoustic measuring devices concerning anomaly machine operating sound detection and classification because each microphone has a different transfer function, frequency response, and perception of the sound; hence, there could be differences in data acquisition during both development and testing. Even though considering its relevance, it received scant attention, first in the wider community of acoustic scene classification and thereafter within machine sound analysis.

The TUT Urban Acoustic Scenes dataset, recorded with various microphones in different acoustic environments in European cities, underlines the importance of device variation in developing robust pattern learning algorithms. This dataset shows how variations in recording devices can affect the performance of classification systems.

This establishes that although limited research is conducted on how mismatches of recording devices affect the detection of anomalous sounds, valuable insight were derived through existing open-source projects such as DCAMS based on the choice of microphones, sampling frequency, and sample duration. Clearly, handling these mismatches is a further step in making the sound classification systems more accurate and reliable. [5]

iii. Addressing Measurement Artifacts / Sound Mixtures with background noise (i. Background Noise and Distance Effect)

Industrial environments offer mixtures of signals, including the target acoustic signal indicative of the machine anomalies and background noises from other operating machinery, together with general factory sounds. In effective monitoring, separation of the acoustic signal of interest from the background noise becomes necessary. Traditional methods for spectral subtraction assume constant background noise and short-time stationarity of the target signal, which may inadvertently remove critical fault frequencies and reduce their applicability.

It is for this reason that the method of Blind Signal Separation (BSS), which requires no prior knowledge of the individual signals or their mixing process, offers a promising alternative. Further research can be done to optimize BSS for acoustic-based machine condition monitoring.
Besides this, the distance between the sound source and the microphone introduces attenuation and gives effects on repeatability between laboratory and industrial settings, hence it affects the accuracy of data-driven models. For this issue, a normalization scheme has been proposed which alters the frequency domain spectrum of the measured sound intensity such that the distance effect is minimized. This involves the use of mean of the rectified time-domain acoustic signal intensity for normalizing the spectrum.

While the above d-normalization scheme shows some promises about spectral representations, how well it will generalize to other acoustic image representations such as cochleagrams, and Mel-spectrograms remains an open question. In this regard, alternative normalization methods must be investigated to further research about their effects on the accuracy of data-driven models for acoustic-based condition monitoring.[5]

## 2.5. Preprocessing Techniques in Acoustic signal Processing

The acoustic signals developed in the present work were experimentally tested during no-load conditions. The acoustic frequency contents are substantially reliant on alternator working frequency and engine speed (RPM). Although there is low combustion noise generated by the generator during no-load, the frequencies in specific bands are still detectable.

- Low frequency band (20 – 200 Hz): Produced by engine firing, alternator rotation hum, and associated mechanical causes.

- Mid-band frequency (200 – 1000 Hz): Resulting from mechanical harmonics related to valve, piston, and shaft motions.
- The high frequency band (above 1 kHz, up to 5 – 10 kHz): Produced by injector noise, turbocharger whining, and mechanical shocks.

This research concentrated on the high-frequency band, specifically up to 10 kHz, since it encapsulates the most pertinent information necessary for identifying alterations in oil condition. All preprocessing methods were designed and implemented in accordance with the generator's operational frequency characteristics under no-load conditions, particularly at an engine speed of 1545 RPM.

Acoustic emission means propagation of structure-borne and fluid-borne waves, a mode of energy release in a material during some physical activity. These may include plastic deformation, microfracture, wear, bubble collapse, friction, and impacts acting on or within the material. These emissions, detectable by piezoelectric sensors, possess a widely varying amplitude in the range from microvolts to millivolts and frequencies of up to several kHz to some MHz. Due to the strong attenuation of the signals in the air, a couplant has to be applied to ensure efficient transmission between sensor and material. [14][21][23]

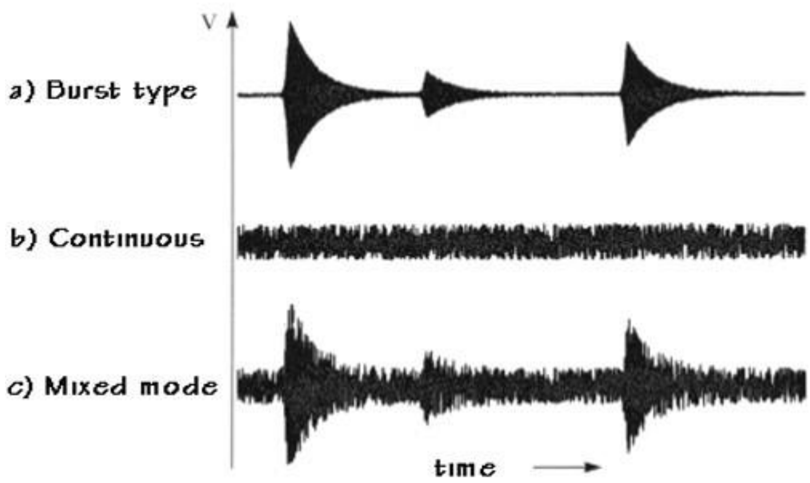



**Figure 2.7:** Types of Acoustic signals [7]

Acoustic signal pre-processing is an essential step in the development of the signal bandwidth to avoid aliasing and consists of amplification and filtering stages. The modes of acoustic signals are continuous, burst, and mixed as shown in Figure 2.7. Burst emissions consist of short, discrete transients, whereas continuous emissions are closely spaced bursts that raise the level of the background signal without individual features being distinct. Large bursts over a continuous background comprise mixed mode acoustic signals. Since the nature of acoustic signals is broadband, time-domain processing is mainly used [7]. Some of the acoustic signals preprocessing methods are as follows,

i. A Frequency Domain Blind Signal Separation (BSS) Method

Blind Signal Separation (BSS) is a method that aims at restoring independent sound sources from their mixtures without any prior knowledge of their original features. BSS in acoustic environments needs to perform convolutive separation, which uses multichannel finite impulse response (MC-FIR) filtering. While BSS algorithms have managed to separate artificially mixed signals, real environments have been more problematic. Moreover, acoustic echo cancellation requires advanced BSS algorithm that integrates acoustic echo cancellation which requires minimal parameter tuning and it is very important for real-world audio applications.

The algorithm computational complexity relates to the number of microphones involved. Also, it is possible to run this algorithm in real time on a standard personal computer. More advanced version of the algorithm could be further developed to perform both echo cancellation and blind signal separation with minimal computational complexity. The effectiveness of the algorithm can be validate using recordings captured in real world environments. Main benefit of the extended algorithm has an advantage over standard conventional echo cancellation because it can be applied Problem-free under double talk conditions. This makes it suitable for applications such as teleconferencing and hands-free telephony [6].

ii. Low pass filter method

A low pass filter (LPF) allows low frequency components to pass while attenuating high frequency components above specified cutoff frequency. In general, LPF is widely used in various applications in audio signal processing. LPF is essential in numerous areas of acoustic signal processing, improving audio quality and guaranteeing correct digital representation. It is extensively applied to noise reduction, removing high-frequency interference while leaving vital sound elements intact, improving speech clarity and music definition. In anti-aliasing, LPFs guard against unwanted artifacts prior to analog-to-digital conversion, providing accurate sampling. They also assist in smoothing audio signals, diminishing abrupt waveform transitions for more natural playback. They also assist in bass enhancement by accentuating low-frequency sounds, favoring subwoofers and equalizers. In speech processing, they remove high-frequency noise, improving vocal clarity. In audio compression and encoding, LPFs eliminate inaudible frequencies, maximizing file sizes and bandwidth utilization. They are also essential in musical effects and sound design, forming tones in synthesizers and filter sweeps. LPFs are invaluable in both technical and creative uses of audio. [9][13],

LPFs are typically used to attenuate the high frequency noise but allow lower frequency signals such as speech or music. Some of the common realizations are

Infinite Impulse Response (IIR) filters, Cascaded Integrator-Comb filters, and Butterworth filters that offer various amounts of attenuation along with computational complexity. LPFs are used analog and digitally within an auditory system to enhance speech intelligibility, attenuate quantization noise, and clarify the signal. More recent uses include the use of adaptive filtering along with discrete-time methodologies to deliver maximum performance during real-time operations [14].

iii. Kalman Filter

The Kalman filter has widespread use to the processing of auditory signals to undertake operations such as speech enhancement, noise cancellation, and echo cancellation. The Kalman filter behaves like an ideal recursive estimator that continually updates based on noisy measurements. The Kalman filter has the ability to model time-varying systems very effectively and is therefore particularly well-adapted to suppressing transient acoustic interference and tracking.

In echo cancellation, the Kalman filter learns to model the unmeasured acoustic echo path to suppress backscattered sound interference employed in telecommunication and hand-free speech communications. More complex implementations such as the General Kalman Filter (GKF) optimize convergence speed and computational complexity through the manipulation of time blocks of samples rather than an isolated time sample. Frequency-domain Kalman filtering has also been contemplated to optimize convergence speed and performance under poor audio environments .

Overall, the Kalman filter's ability to dynamically adapt to the progression of acoustic conditions makes the Kalman filter an effective tool within modern devices handling audio signals. [12]

## 2.6. Windowing

In a typical machine learning system operating in the time domain, windowing is essential for processing real-time audio signals, which are inherently non-stationary over time. To effectively analyze these signals, the windowing technique is employed, segmenting long non-stationary signals into shorter quasi-stationary chunks.

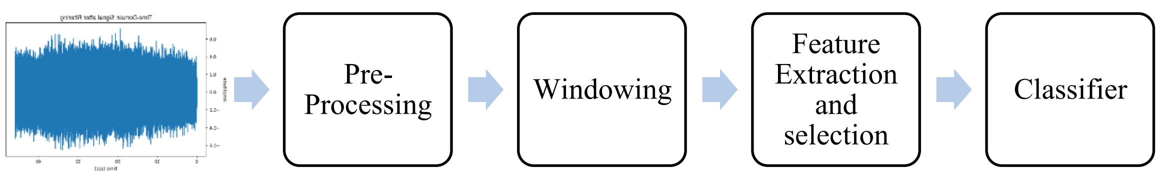



**Figure 2.8:** Typical Machine Learning System in time-domain [8]

Figure 2.8. illustrates a typical machine learning system utilizing audio signals in the time domain, where windowing is implemented by multiplying the signal with a window function, which remains zero outside the designated region of interest [8].

Figure 2.9 illustrates windowing of a signal using a rectangular window as a function. For analyzing the whole non-stationary signal, the rectangular window has to be shifted over time from the left of the signal plot towards the right side of the plot. Window size depends on the type of the original acoustic signal, and it is rendered adaptive. Figure 2.10 illustrates the process of adaptive rectangular window sliding over time.

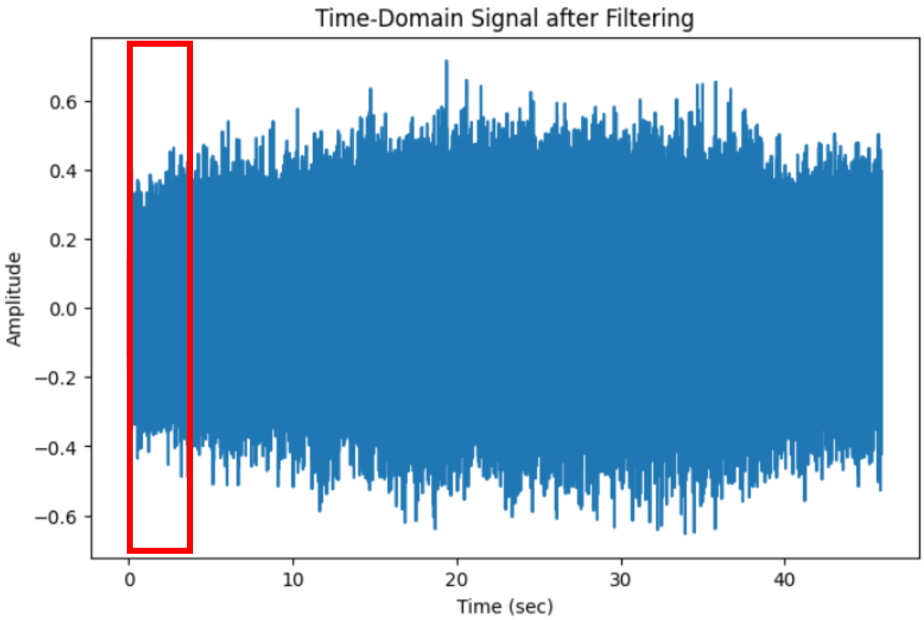



**Figure 2.9:** Windowing in Time Domain

One of the major limitations of the rectangular window is the abruptness of the transitions at the edges, which may introduce some distortions in the analysis. These are called the Gibbs phenomenon and result from the sharp changes in the shape of the window. To avoid this problem, window functions with smoother edges are used, such as the Hamming and Hanning windows. As shown in Figure 2.11 these functions taper smoothly to zero at the edges, rise gradually to a peak in the middle, and in that manner, smoothing reduces edge effects and minimizes distortions due to the Gibbs phenomenon greatly.

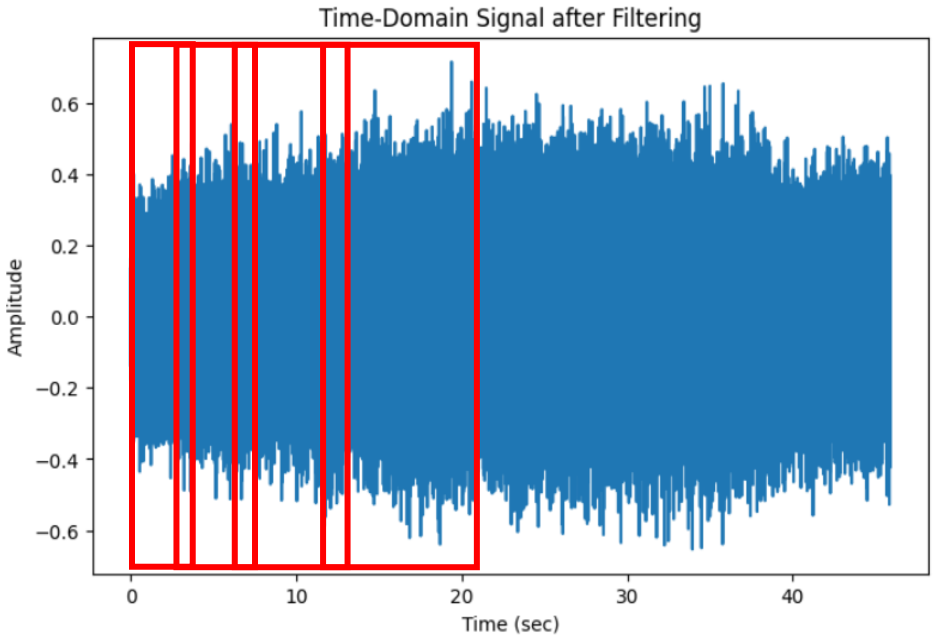



**Figure 2.10:** Adaptive Windowing & Window shifting in Time Domain

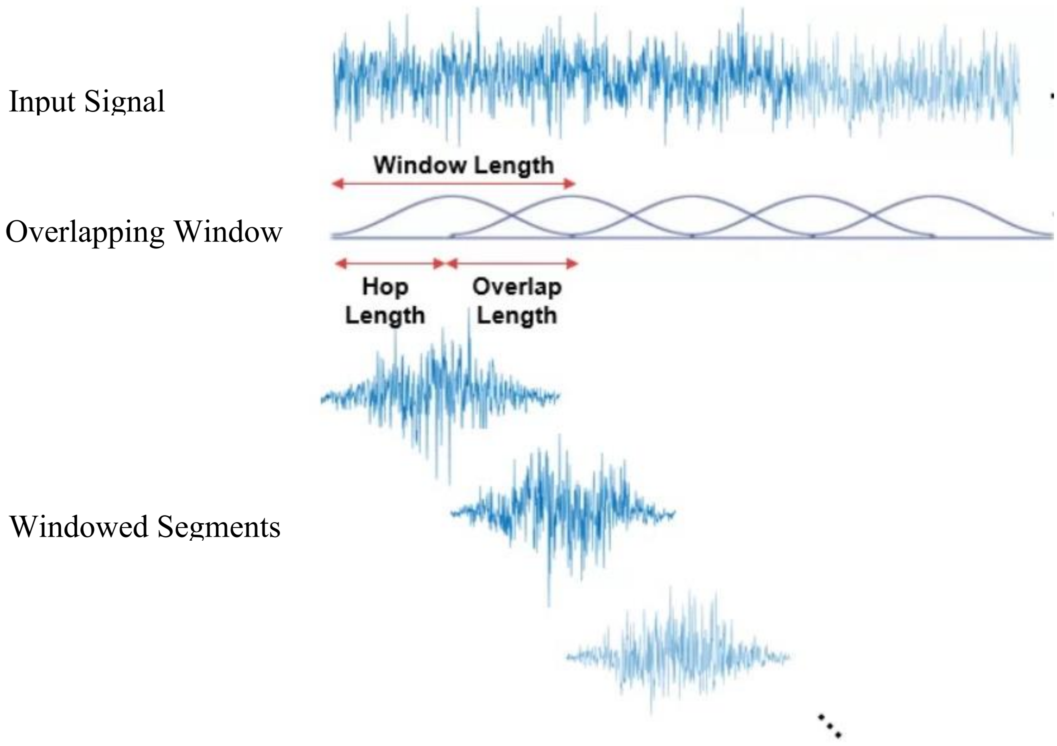



**Figure 2.11:** Windowing using Hanning windows

Thus, windowing is a very important technique in time domain analysis, allowing for good processing and feature extraction in non-stationary audio signals. [17][18]

## 2.7. Feature Extraction methods

Audio feature extraction can be understood to be such an analysis that emphasizes the most salient and distinctive properties of a signal. A good feature is one that captures the salient aspects of an audio signal in a more compact form. In general, the development of features for audio signals, as depicted in Figure 2.12, has fallen into the following categories: time domain, frequency domain, joint time-frequency domain, and deep features. [24]

The earliest and most straightforward features had their origin from the time domain and evolved up until the late 1950s; they still are very important for audio analysis and classification. During the 1950s and the 1960s, features such as pitch and formants emerged from the frequency domain and have seen applications in various contexts. The first algorithms of joint time-frequency feature extraction appeared in the late 1960s. Since then, they have been the focus of numerous efforts in audio signal processing. Nowadays, deep features are widely used in many applications. Since 2010, they started to spread to various areas such as acoustic scene classification, speaker recognition, and audio-video analysis. [8][24]

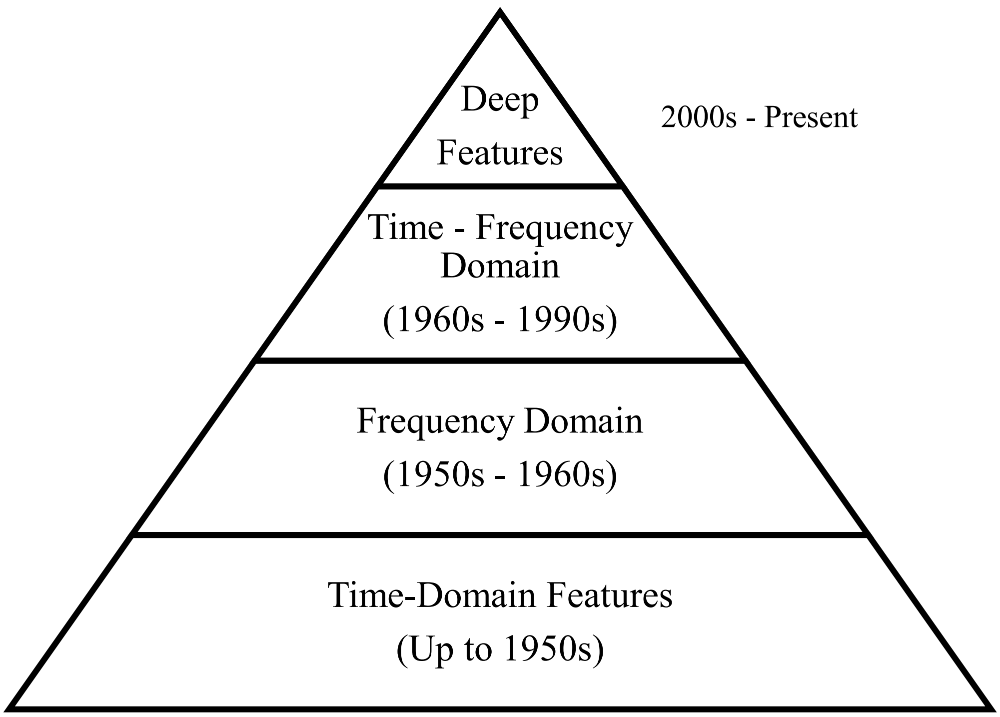



**Figure 2.12:** Evaluation of Audio Feature Extraction [8]

i. Time Domain features

Time domain feature extraction is a simple technique for analyzing sound signals with a focus on direct features from the original signal and not transforming to frequency space. This method records prominent features such as amplitude, energy, rate crossing zeros, and entropy to define behavior in a signal in terms of time. These features find applications in real-time audio processing, speech, and fault detection since they provide computationally effective information regarding behavior in a signal. With time-domain analysis, machine learning models can effectively class and interpret acoustics patterns for a variety of applications.

As discussed in section 2.6, time domain analysis is quite simple if the signal is stationary or short in time. Real-time audio signals are not stationary. For analyzing this kind of signal, the windowing technique is used. In this technique, a long non-stationary signal is divided into small quasi-stationary segments with the help of a window.

a. Amplitude Envelope (AE)

   The amplitude envelope is a smooth curve outlining the peaks of the waveform of a time-domain signal, thereby tracing out the general shape of the amplitude as a function of time as shown in Figure 4.5. It is one of the key features of the sounds produced by musical instruments and speech for understanding their amplitude modulations. A correct estimation of the amplitude envelope is necessary to prevent overfitting, and at the same time be able to model sudden changes in amplitude while maintaining smoothness in stable regions. [9]

   The AE is a valuable feature in machinery condition monitoring, particularly for detecting faults and assessing oil condition. Variations in the amplitude envelope can indicate issues such as increased friction due to oil degradation, leading to wear or damage in machine components. By analyzing the characteristics of the amplitude envelope, including periodicity and modulation, deeper insights into the health of machinery and its lubrication state can be obtained. [10][23]

b. Root Mean Square Energy (RMSE)

   Root Mean Square Energy reflects the overall energy of the signal. RMSE is one of the most straightforward yet important spectral feature. This feature can be computed directly as a sum of the squared amplitude components in the band. Figure 4.6 shows the RMSE of audio sample 7 in red. It is also common to compute the log energy in a band, and it can be used to identify abnormal sounds associated with oil breakdown.[9] [15] [19].

c. Signal Entropy

   Signal entropy is used as a quantitative measurement to obtain fault information features from sensor signals for enabling fault detection, diagnosis, and prognostics. Entropy measures such as Shannon entropy, energy entropy, permutation entropy, Rényi entropy, sample entropy, and fuzzy entropy are explained in detail for use in vibration, acoustic emission, and other condition monitoring signals. These entropy measures are used in optimizing machine learning algorithms and signal processing techniques for accurate and reliable condition monitoring of industrial rotating equipment. In oil condition monitoring signal entropy measures the randomness of the signal and it can be sensitive to the presence of contaminants or bubbles in the oil. In this research Shannon Entropy variant, energy entropy is used to predict oil condition monitoring [16]. Energy entropy can be used to quantify the regularity of the data series. For analyzing condition monitoring data of the machinery.

d. Impact Index

The Impact Index is a quantitative measure used in acoustic signal processing for the detection and analysis of impulsive phenomena in a signal. It is used in a very broad field of fault detection, structural health monitoring, and non-destructive testing for abnormal detection such as defects. A vast array of methods like kurtosis analysis, energy variation, and entropy-based methods are used to compute impact index in order to highlight transient impact phenomena. The impact index is particularly effective in analyzing non-stationary signals and can be used to enhance machine learning-based fault diagnosis systems by providing a reliable indicator of impact-based abnormalities. This feature captures the intensity of short-duration, high-amplitude events in the signal, potentially related to metal-to-metal contact due to severe wear [16].

e. Zero Crossing Rate (ZCR)

Zero Crossing Rate counts the number of times the signal crosses the zero-amplitude line and can be indicative of changes in friction within the generator. Figure 4.7 shows the ZCR of audio sample 7 and ZCR is a measure that is commonly utilized as it can directly discriminate periodic signals (low ZCR values) from random to a certain degree contaminated signals (high ZCR values) [15].

ii. Frequency Domain features

a. Band Energy Ratio (BER)

Equipment are operating under two frequency bands, low frequency band and high frequency band. Low frequency band represents the fundamental operating frequency of the generator and its associated vibrations while high frequency band capture noise related to internal friction and component wear. In a healthy generator with good lubrication, the energy in the low-frequency band might be dominant compared to the high-frequency band. This would result in a relatively low BER value for the high-frequency band. However, as oil degrades and friction increases, the energy in the high-frequency band might become more prominent compared to the low-frequency band. This shift in energy distribution would lead to a higher BER value for the high-frequency band [17]. Figure 4.8 shows the BER of sample 7.

b. Spectral Centroid

Spectral Centroid (SC) is the magnitude spectrum's center of gravity, which the interval of frequencies that encompasses the most energy. SC is commonly used as a marker of the brightness of a sound. SC is, however, highly vulnerable to low-pass filtering, as it gives more weight to the higher frequencies. Such a vulnerability is a problem to down-sampling a sound signal (e.g., 44,100 Hz to 11,025 Hz), as higher frequencies tend to diminish or disappear as a result of the application of the Nyquist–Shannon theorem, hence altering the SC values. This center of mass of the frequency spectrum can be indicative of changes in lubrication properties [17]. Figure 4.9 shows the SC of audio sample 7.

c. Spectral Bandwidth

Spectral Bandwidth (BW) is also known as Spectral Spread (SS), it is derived from the Spectral centroid, thus also a feature of the frequency domain. Spectral bandwidth offers a measure of the width of the interesting regions of the signal around the centroid. Also, BW can be represent as a variance of the mean frequency of the signal. The impact of oil degradation on spectral bandwidth can be complex and depends on the specific type of degradation and the generator's operating conditions [17]. Figure 4.10 shows the BW of audio sample 7.

d. Spectral Roll-off Frequency

Spectral roll-off is the frequency below which there is a designated percentage of 85% to 99% of the overall spectral energy as indicated in Figure 4.11. It can be prone to wear particles in the oil [15].

e. Spectral Flux

Spectral flux is a term used to describe the dynamic evolution of spectral information. It is computed either as a derivative of the amplitude spectrum or as the normalized correlation of successive amplitude spectra as shown in Figure 4.12. This measures the rate of change in the frequency spectrum over time and can be sensitive to sudden changes in oil viscosity [15].

## 2.8. Feature Selection methods

The accuracy of a Machine Learning (ML) model is a function of integrity in its input data. High datasets with irrelevant attributes generate unnecessary complexity, with overfitting becoming an issue and model performance compromised. Removing irrelevant attributes through feature selection reduces datasets, with high accuracy, less computational cost, and less demand for memory requirements. In contrast to feature extraction, feature selection entails no transformation of data. Methods for feature selection entail three types of approaches:

i. Wrapper Methods – Optimize feature sets according to model performance.

   Wrapper methods evaluate feature utility through performance impact analysis for a learning algorithm. Wrapper methods contrast with filter approaches in that wrapper approaches utilize a model and select feature sets through iterative testing.

   Wrapper approaches, such as Genetic Algorithm, produce optimized feature sets for an ML model with high accuracy and computational efficiency but with a computational cost in contrast with filter approaches.

ii. Filter Methods – Use statistics for feature ranking and filtering out irrelevant features. It rank features model-independently, comparing them in terms of redundancy and relevance with use of statistical values.

   a. Minimum Redundancy Maximum Relevance (MRMR) Method

      This method centers on condition monitoring through feature selection optimization. It minimizes the feature space by choosing the most pertinent features and penalizing redundancy. The method guarantees maximum relevance by picking features that best describe the target class and minimum redundancy by making sure the chosen features are as diverse as possible.

   b. Fisher's Discriminant Ratio (FDR) Method

      This method examines feature selection and information extraction by measuring the discriminatory power of a single feature. The solution involved selecting the most stable features, significantly improving classification performance. Filter-based methods such as MRMR and FDR enhance machine learning models by selecting optimal feature sets with low redundancy and high relevance, leading to improved performance with reduced computational expense.

iii. Embedded Methods

Embedded feature selection methods, where feature selection is done while training models. Embedded methods leverage the efficiency of filter methods and the accuracy of wrapper methods. Random Forest, for example, orders the feature importance by contributions to decision splits, which decides the most useful features. Unlike wrapper methods, embedded methods reduce computational cost by selecting features at run-time. Embedded methods, i.e., Random Forest, improve the model's performance as they enable effective and efficient feature selection during training.

The use of correct feature selection methodologies can make ML models efficient, understandable, and predictive with high accuracy. [11]

## 2.9. Machine Learning Techniques for Condition Monitoring

Machine Learning is a subset of artificial intelligence specialized in building algorithms based on data and continuous improvements. ML models for condition monitoring can be categorized into four subsections, classification, clustering, regression and ensemble. [11][22]

i. Classification Analysis

Classification analysis describes an ML technique for class label data prediction from historical observation. Classification can be conducted in a supervised way or unsupervised way. To compare classification model performance, a broad range of standard performance indicators can be used, including accuracy, precision, recall, specificity, and the F1-score. [19]

a. Multi-Layer Perceptron (MLP)

MLP represents an extremely easy feed forward neural network technique for supervised learning, carrying the signal one direction only (forward) from the input, through the hidden, up to the output. The neurons also only have one activation function. The function represents the mathematical "gateway" from the input to the output to the next one as an ongoing function. In other words, there is learning of higher representations by carrying activation functions. The most used one remains the ReLU (Rectified Linear Unit), but industrial usage applies to others like Sigmoid, Softmax, Leaky ReLU, PReLU predominantly. It's also the last procedure of MLP learns the data pattern by applying the backpropagation learning algorithm. The output of the network minus target output is propagated back to adjust network's bias and weights. The learning algorithm therefore enhances its prowess to discern the pattern.

b. Radial Basis Function Neural Network (RBFNN)

RBFNN is also a supervised classifier of the entire region of interest as well as of the feedforward type of neural network. RBFNN is the same in structure as MLP, with only one hidden layer. An activation function of the radial basis function type is implemented in order to transform the high-dimensional space of the hidden space. RBFNN algorithms can easily implement classification or identification of faults.

c. Convolutional Neural Network (CNN)

Convolutional Neural Network or Convolutional Network is a supervised deep learning technique to process topological data sets and was originally inspired by visual cortex. CNN is a generalization of ANN (Artificial Neural Network) and consists of sparse connectivity, and they are used with a smaller kernel relative to input. A standard CNN consists of fully connected, convolution, and pooling layers. The convolutional layer is the very first and it extracts the features of input data by convolving input data with smaller-sized kernels for producing an activation map. Therefore, CNN can directly point towards trends within a locality of data, unlike MLP, which can learn global trends. [20]

d. Autoencoder

Autoencoder is an ANN used for encoding the pattern of the given data and subsequently reconstructing it with less discrepancy. The form of the encoder is mostly three form: the encoder stage consists of a sequence of linear feed-forward filters, the activation stage is a nonlinear mapping of encoded coefficients to a value between 0 and 1, and the decoding stage assembles the input from mapped values after the activation stage. In most situations, an error term is established so that the error generated by the middle layer can be computed. Some of them, such as stacked sparse autoencoder, denoising autoencoder, adversarial autoencoder, and even more are implementing autoencoders in their frameworks.

If the input signals are noisy in the actual system, then it is inconvenient to reconstruct the input data vector. As compensation for this flaw, one such solution is adding noise to input vectors but measuring the reconstruction loss against the denoised input. This is referred to as denoising autoencoders.

e. Bayesian Network (BN)

Bayesian Network or Bayesian Network Classifier is an unsupervised or supervised model that senses and represents multi-varied distributions by computing the Bayesian inference probability. It is constructed in three steps:

detect the dependencies among the variables, forecast the prior probability distribution, and compute the conditional probability distribution. They are used in fault detection to forecast the response values (fault labels) of a set of observations.

f. Native Bayes

Native Bayes (NB) classifier's (NBC) derivation was developed to bypass the BN's intractable complexity. Similar to BN, the occurrence of an event is predicted by learning from experience, but unlike BN, every feature is conditional in the class known as "parent".

g. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a statistical Machine Learning method. SVM algorithm provides a mapping from inputs to outputs using training data as supervised machine learning methods. Additionally, SVMs can handle big data and address multidomain classification industrial problems.

h. Fuzzy Logic (FL)

Fuzzy Logic (FL) or Fuzzy Set Theory is a robust technique to transform imprecise inputs into precise outputs based on linguistic rules. Real-world information is often imprecise and partially true, which creates a fuzzy situation where classical decision theories may fall short in terms of strength. FL is most relevant in decision-making problems under this classical decision-making situations.

FL is based on four fundamental building blocks: Membership Functions (MFs), which associate every element of the data with a membership grade in the interval 0 to 1; fuzzy sets, which classify elements with a membership grade continuum; fuzzy logical operators, which form new fuzzy sets from ones provided; and fuzzy rules, which describe relationships between inputs and outputs as per human reasoning. To get a final result, a defuzzification process is required to convert fuzzy outputs into a single crisp value. A system utilizing FL for decision-making, mimicking human expert reasoning, is referred to as a Fuzzy Inference System (FIS). Of various FL models, Mamdani and Sugeno fuzzy rule-based systems are widely applied in Predictive Maintenance (PdM). This approach is extremely useful in industrial applications, e.g. microcontrollers, for system optimization and decision-making purposes.

ii. Cluster Analysis

Cluster analysis shares the objectives of classification analysis but with other methods. While classification relies on prior labels, clustering attempts to partition data points into groups such that similar data points are grouped together. Cluster models are typically evaluated using measures of classification performance.

Clustering-based models can be used in both supervised and unsupervised learning. In condition monitoring, unsupervised clustering techniques have been the focus of much research interest in recent times. Among the most commonly used clustering techniques, as recognized in recent studies, are K-means, Fuzzy C – Means (FCM), and Agglomerative Hierarchical Clustering, which detect patterns and outliers in data very effectively.

a. K-means clustering

K-means clustering is an iterative unsupervised algorithm for partitioning n observations into k clusters. The algorithm assigns new points to the nearest cluster center and resets the cluster centers at the end of each cycle in an effort to minimize intra-cluster variance, thereby enhancing the cluster membership.

b. Fuzzy C – Means (FCM)

FCM is an unsupervised dynamic clustering algorithm similar to K-means but incorporates fuzzy membership functions to determine the degree of membership of each point in each cluster. Memberships quantify data point-cluster centroid relationship strength to achieve stronger and more accurate clustering. The FCM clustering method for fault detection in grounding distribution systems, demonstrating its applicability in condition monitoring applications.

c. Hierarchical Clustering

Unlike K-means and Fuzzy C – Means (FCM), agglomerative hierarchical clustering is a bottom-up approach, initially assigning each data point to a separate cluster. As the process continues, similar clusters are progressively combined using a defined measure of similarity method. The procedure continues until all data points are assigned to a single cluster. The optimal number of clusters is then determined by a dendrogram, which represents the hierarchy of the clusters in terms of the cluster distances.

d. Support Vector Machine (SVM)

SVM principle can even be applied in clustering analysis. One-Class SVM (OC

SVM) is an unsupervised learning strategy that constructs a spherical decision boundary to address problems of one-class classification. OCSVM differentiates normal instances from potential outliers using a special optimization equation and is particularly valuable in anomaly detection and fault detection in most applications.

iii. Regression Analysis

Regression is a machine learning technique used to predict continuous output variables from observed data. In fault detection, fault models identify faults by tracking fault-free patterns and considering deviations from such patterns to be faults. The second main application is in Remaining Useful Life (RUL) estimation, where regression techniques predict the estimated life of machinery based on operational parameters, enabling proactive maintenance and avoiding failures.

a. Recurrent Neural Network (RNN)

RNN is an ANN that is viable to detect the pattern in the sequential data. The RNN network has a number of feed-forward neural networks that transmit information to the next node, which is called the hidden state. In other words, the output from the previous node is the compounding of the knowledge it has accumulated so far (hidden state) and its experience and input to the next node. RNNs thereby utilize past output to obtain the next. The algorithm, however, is computationally expensive since the network requires two inputs and operations are not in parallel. Moreover, RNNs suffer from two well-documented problems: "gradient vanishing" and "exploding gradients".

b. Support Vector Regression (SVR)

Besides classification issues, SVM is also applied to regression issues with Support Vector Regression (SVR). The transition from SVM to SVR is realized through the introduction of an ε-insensitive (It defines a region of tolerance in which errors are not penalized) area, or the ε-tube (boundary or region surrounding the regression function in which errors are tolerated and not penalized), over the function. In supervised learning using this method, the optimization problem is rotated within the tube to get the best approximation of the continuous-valued function. The flattest tube containing most of the training points is the best solution with least deviation and yet generalization power.

iv. Ensemble Learning

Ensemble learning is a collection of several supervised or unsupervised machine learning algorithms to minimize overall variance by combining their outputs. The main purpose of ensemble learning is to overcome the individual limitations of each model for better accuracy. To achieve this, several ensemble methods, such as Bagging, Stacking, and AdaBoost, have been developed.

a. Stacking

Stacking method involves blending heterogeneous weak learners, which are parallel to one another, with the output of these weak learners being utilized to train a meta-learner.

b. Bagging

The bagging (Bootstrap Aggregating) method trains learners in parallel and combines their outputs deterministically by a simple average.

c. Boosting

In contrast to Bagging, the Boosting method addresses the weaknesses of the ensemble in a sequential manner, rather than in parallel.

These ML techniques applied to predictive maintenance, emphasizing the importance of proper data preprocessing and feature selection to maximize ML model effectiveness. The above literature review suggest that model selection should consider data characteristics, system constraints and cost effectiveness. In some cases, hybrid approaches combining simple physics-based models with signal processing may offer optimal results. A balanced, comparative assessment is crucial to selecting the most suitable ML strategy for condition monitoring. [11]

## 2.10. Librosa Audio Processing Library using Python

Librosa is a Python audio analysis library that is widely used in the field of acoustic signal processing due to its flexibility, ease of use, and full range of features. Librosa, in acoustic use, facilitates researchers and developers to process, analyze, and extract meaningful patterns from raw audio signals, which are crucial in applications such as classification, monitoring, and fault detection.

Librosa provides a range of functionalities particularly beneficial in acoustic signal analysis,

i. Feature Extraction:

Librosa provides an easy way to extract audio features such as Mel-Frequency Cepstral Coefficients (MFCCs), chroma features, spectral centroid, zero crossing rate, and mel-spectrograms etc. These are utilized in such tasks as sound classification, emotion recognition, and anomaly detection etc.

ii. Audio Preprocessing:

It offers standardized processing of audio inputs, such as resampling, normalization, noise reduction, and trimming of duration, that is required for model performance consistency.

iii. Time-Frequency Analysis:

Librosa facilitates the conversion of time-domain audio signals to frequency-domain representations using functions like STFT (Short-Time Fourier Transform) and Mel-scale transforms.

iv. Visualization:

The library also features a display of waveforms, spectrograms, and other acoustic features to aid in understanding and debugging of audio data.

v. Efficient Integration with Machine Learning Pipelines:

Librosa outputs are in NumPy format, and it supports integration with libraries like scikit-learn, TensorFlow, and PyTorch for downstream classification or prediction tasks.

Considering the benefits and capabilities in acoustic signal processing, the Librosa library was utilized in the proposed oil condition prediction framework for data preprocessing, feature extraction, feature selection, and integration with machine learning algorithms.

## 2.11. Literature Review Summary

In summary, the literature review attests to the viability and potentiality of acoustic-based oil condition monitoring using machine learning but also poses some issues open for discussion. These include the need for robust preprocessing in variable industrial conditions, standard feature selection practices, and hybrid models combining domain expertise with data-driven methods. Table 2.1 tabulates the reviewed literature by highlighting the challenges in acoustic signal acquisition, the techniques used for preprocessing, the extraction of the features, selection of the features, and machine learning techniques.

**Table 2.1:** Literature Review Summary

| Focus Area | Key Concepts / Techniques | Challenges / Insights |
|---|---|---|
| Acoustic Signal Acquisition Challenges [5] | - Microphone placement<br>- Device mismatch<br>- Background noise interference<br>- Distance attenuation and normalization | - Variability in signal capture<br>- Inconsistent data due to device differences<br>- Signal loss due to attenuation<br>- BSS and normalization methods needed |
| Preprocessing Techniques [6][7][9][12][13][14][21][23] | - Blind Signal Separation (BSS)<br>- Low Pass Filters (LPF)<br>- Kalman Filters | - Complexity in real-world application<br>- Requires balance between filtering and feature preservation<br>- Adaptive filtering needed for real-time processing |
| Windowing [8][17][18] | - Rectangular, Hamming, Hanning windows<br>- Adaptive sliding windows | - Edge effects (Gibbs phenomenon)<br>- Trade-off between time and frequency resolution |
| Feature Extraction [8][9][10][15][16][17][19][23][24] | - Time-domain features (AE, RMSE, ZCR, Entropy, Impact Index)<br>- Frequency-domain features (BER, SC, BW, Roll-off, Flux) | - Selection must match application context<br>- Feature variability with oil degradation and machine wear |
| Feature Selection [11] | - Wrapper (GA), Filter (MRMR, FDR), Embedded (Random Forest) | - Reduces complexity and overfitting<br>- Balancing relevance and redundancy |

| | | |
|---|---|---|
| Machine Learning Techniques [11][19][20][22] | - Classification (MLP, CNN, RBFNN, SVM)<br>- Clustering (K-means, FCM, Hierarchical)<br>- Regression (SVR, RNN)<br>- Ensemble (Bagging, Boosting, Stacking) | - Accuracy depends on preprocessing and feature set<br>- Model choice must consider data type and resource limits |
| Librosa Library (Python) | - Audio loading, trimming<br>- Feature extraction (MFCC, SC, ZCR)<br>- Visualization and ML integration | - Enables efficient and consistent signal analysis<br>- High compatibility with ML workflows |

## 2.12. Research Gap

Current techniques utilized in monitoring generator oil condition are mainly based on traditional Scheduled Oil Sampling (S·O·S) test methods. The techniques involve the acquisition of oil samples at predetermined time intervals, which are then sent out to laboratories for analysis. Even though the techniques offer a few advantages, they have a range of shortcomings, such as the delayed detection of oil degradation, ineptness in monitoring real-time changes, and a greater chance of unforeseen generator failure. Additionally, fault detection systems implemented in generators are mainly designed to detect mechanical issues yet are not directly designed by specific real-time oil condition predictions.

This produces a sizeable research void within generator maintenance and reliability improvement. A particular deficiency exists within the area of modern, real-time, non-invasive methods of monitoring generator oil condition. Current methodology does not adequately address dynamic, real-time changes in oil quality, important in support of anticipative maintenance and avoidance of expensive breakdowns.

As a response, Figure 2.13 presents work targets establishing a technique of real-time estimation of oil conditions by acoustic signals. The technique is based on exploiting the interconnectivity of acoustic patterns emitted by the engine and changes in oil quality, thereby permitting uninterrupted monitoring without interfering with generator operating processes.

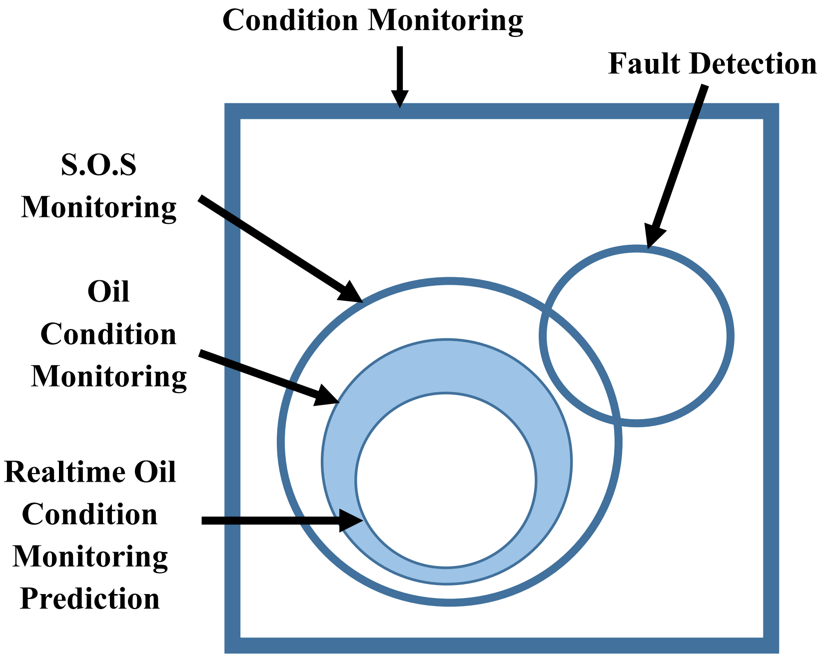



**Figure 2.13:** Research Contribution

The research is guided by the following objectives:

- Signal and noise recognition: Identify and separate oil degradation-related acoustic signals from background noise.
- Extracting features: Extract important features from the time domain and frequency domain through the application of spectrum analysis techniques.
- Correlation and feature selection: Find and choose features that are strongly correlated with oil deterioration patterns.
- Machine Learning Model Development: Develop a machine learning model applicable on extracted acoustic features to forecast generator oil condition.

This research enhances the larger domain of condition monitoring by advancing from conventional sampling techniques to real-time predictive maintenance, thereby diminishing downtime, enhancing the reliability of generators, and optimizing the scheduling of maintenance activities.

# CHAPTER 3

# METHODOLOGY FOR OIL CONDITION PREDICTION

This chapter discuss about the methodology of proposed oil condition prediction method. Research sample space was based on CAT diesel generators. Normally S·O·S testing is done for the following reasons, it can be due to check contamination of dirt, water, fuel, glycol, wrong oil, metal particles, soot, oil degradation and additive depletion as discussed in Chapter 2, which may be helpful for troubleshooting purposes and to be aware of equipment condition and status.

## 3.1. Samples collecting

In this research complete sample consist of two parts, getting the audio sample by running the generator with collected oil sample and physical Oil sampling by using standard procedures as shown in section 2.2.

### 3.1.1. Sampling of Audio Signals

i. Microphone Details

As mentioned above, a complete sample consist of S·O·S results and the audio sample. When taking the corresponding audio sample, audio signal characteristics depends on the microphone properties and its characteristics. In this research "Blue Yetti" microphone was used for extracting audio samples as presented in Figure 3.1. The Blue Yetti microphone specifications and characteristics are as below,


**Figure 3.1:** Blue Yetti Microphone which is used to collect Audio Samples

As per the Owner's manual Blue Yeti Nano Microphone Technical Specifications are mentioned below,

Microphone and performance

- Power required/consumption: 5V 150mA
- Sample rate: 48kHz
- Bit rate: 24bit
- Capsules: 3 Blue-proprietary 14mm condenser capsules
- Polar patterns: Cardioid, Bidirectional, Omnidirectional, Stereo
- Frequency response: 20Hz-20kHz
- Sensitivity: 4.5mV/Pa (1kHz)
- Max Sound Pressure Level (SPL): 120dB (THD: 0.5% 1kHz)

Headphone Amplifier

- Impedance: >16 ohms
- Power Output (RMS): 130 mW
- THD: 0.009%
- Frequency Response: 15Hz-22kHz
- Signal to Noise: 100dB

Specifications

- Dimensions (extended in stand): 4.72" (12cm) × 4.92" (12.5cm) × 11.61" (29.5cm)
- Weight (microphone): 1.2lbs (0.55kg)
- Weight (Stand): 2.2lbs (1kg)

System requirements:

- PC: Windows 7, Windows Vista, XP Home Edition or XP Professional
- USB 1.1/ 2.0; 64 MB RAM (minimum)

Above, mentioned Blue Yetti microphone equipped with four polar patterns. Polar patterns are like the ears of microphone; each polar pattern hear the world in a distinct way. These unique modes of polar patterns dictate which directions the microphone will pick up or ignore the sound. The four polar patterns details are as follows:

a. Cardioid Pattern

The cardioid pattern in the Blue Yeti microphone is such a type of unidirectional pickup, which mainly picks up sound from the front of the microphone and reduces pickup from the sides and rear as shown in Figure 3.2.

This type of pattern is suited for single-source recordings, including podcasting, streaming, voice-overs, and vocals, as it isolates the operated sound source and provides minimal background noise. The cardioid pattern ensures that the sound is clear and focused from the front, catching only the front sound, and hence preferred during solo recording sessions.

When using the Cardioid Pattern gain setting should set in between  Medium to Low because it do not need to amplify any distracting background noise.

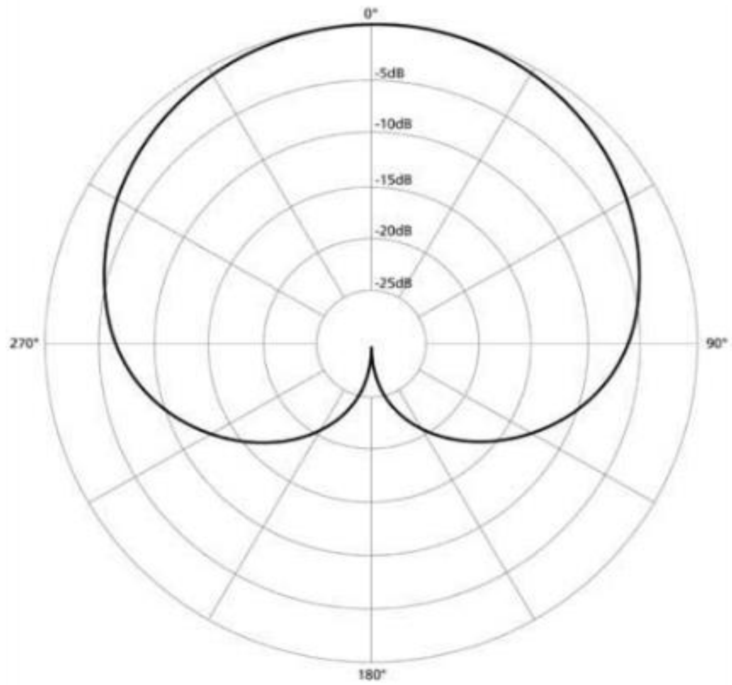



**Figure 3.2:** Blue Yetti microphone Cardioid Polar Pattern

b. Bidirectional Pattern

The Bidirectional pattern in the Blue Yeti microphone is set to capture sound equally from both the front and rear of the microphone, while picking up very little from the sides. This pattern would be perfect in a situation when recording a two-person conversation or an interview where participants sit across each other, for instance, across the table.

Bidirectional polar pattern will capture the sound as shown in Figure 3.3, therefore gain needs to set to medium when using bidirectional polar pattern.

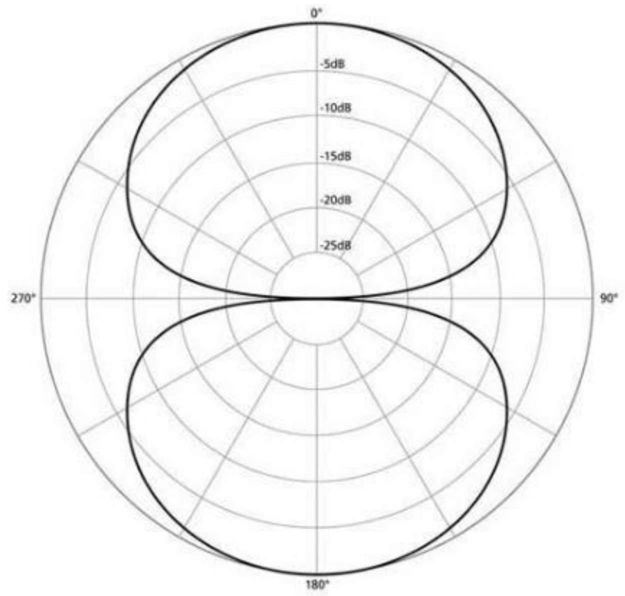



**Figure 3.3:** Blue Yetti microphone Bidirectional Polar Pattern

c. Omnidirectional Pattern

The omnidirectional pattern on the Blue Yeti microphone is set to pick up sound from every direction: front, back, and sides as shown in Figure 3.4. This pattern would be ideal in situations where you want to capture ambient sounds or multiple people speaking from any position around the microphone. This ensures a very natural and airy recording, thus perfect for capturing the room atmosphere, conference calls, or group interviews. The omnidirectional mode picks up audio levels and tone consistently, irrespective of the position of the sound source from the microphone.

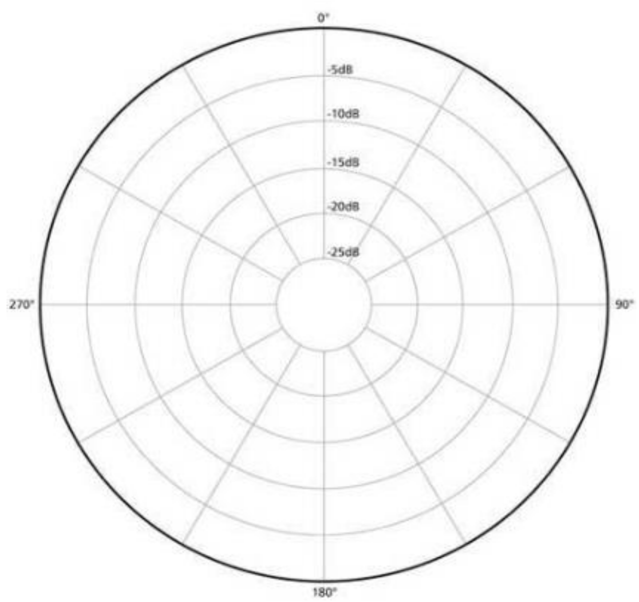



**Figure 3.4:** Blue Yetti microphone Omnidirectional Polar Pattern

d. Stereo Pattern

The stereo pattern in a Blue Yeti microphone captures sound from left and right channels as shown in Figure 3.5, making it ideal for creating a realistic and immersive audio experience. This pattern is commonly used for recording music, capturing the ambiance of a room, or when a distinct separation of sound sources is desired, such as in ASMR recordings or live music performances. It allows for a natural and wide stereo image, enhancing the spatial quality of the audio.

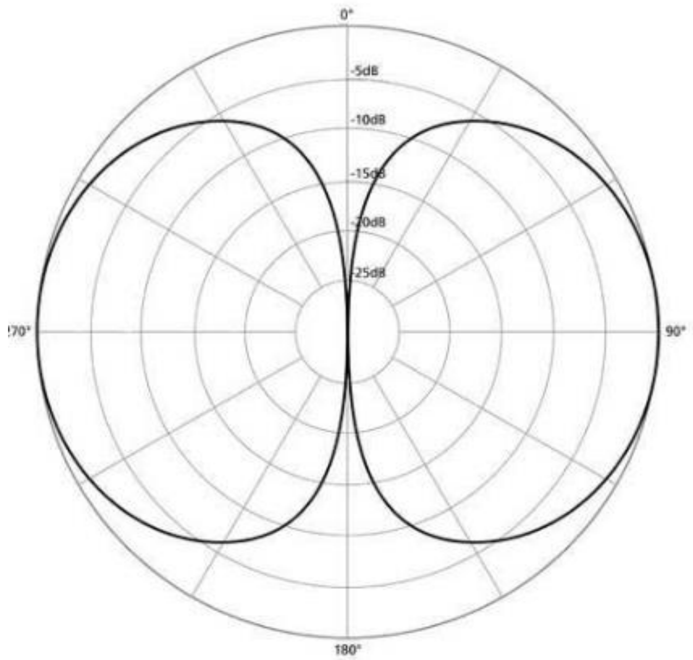



**Figure 3.5:** Blue Yetti microphone Stereo Polar Pattern

In this research, to collect the audio signal of the generator Cardioid pattern was used because there’s no requirement of amplifying any distracting background noise. To achieve this gain setting was set in between medium to low.

ii. Sampling process

To get the Audio samples of the generator acoustic signals, microphone and microphone stand was set as shown in Figure 3.6. Microphone distance was maintained as per the literature review of the measurement distance of audio sampling as mentioned in Chapter 2. To address these measurement artifacts, mainly background noise and distance effect measurement distance, r was maintained, $1 \text{ m} < r < 1.5 \text{ m}$ . Likewise, by minimizing the measurement artifacts audio samples were taken as shown in Figure 3.7 and Figure 3.8.

In the meantime, oil samples were taken as per the standard procedure for S·O·S testing, which means all relevant sample details of the test were taken as shown in Figure 1.2. After that samples were sent to the respective laboratories to have the results.


**Figure 3.6:** Apparatus for collecting audio signals from the generator


**Figure 3.7:** During acoustic sample collecting process - 1

**Figure 3.8:** During acoustic sample collecting process - 2

### 3.1.2. Sample summary

Table 3.1 illustrates in detail information on collected oil samples from various models of diesel generators. In every sample, information indicated in the table includes the model type of the generator, the corresponding engine oil capacity in liters, reading at the service meter unit (SMU) when the sample was taken, and the hours run by the engine oil at that time since it was last changed. Generator types all share a range of Caterpillar engines from C13, C15, C18, 3412, 3406, 3306, C32, 3512, to 3516, and oil capacities ranging from 10 L to 420 L. The oil running hours are extremely variable across samples and depend on engine operating patterns and maintenance schedules.

**Table 3.1:** Sample Summary

| Sample No. | Gen. Model | Gen. Oil Capacity (L) | SMU | Oil Running Hours |
|---|---|---|---|---|
| 1 | C13 | 45 | 1518 | 318 |
| 2 | C18 | 90 | 1189 | 167 |
| 3 | C13 | 45 | 1311 | 290 |
| 4 | 3406 | 65 | 2567 | 107 |
| 5 | 3412 | 90 | 3038 | 211 |
| 6 | C15 | 65 | 1645 | 189 |
| 7 | 3306 | 45 | 3340 | 255 |
| 8 | C18 | 90 | 1342 | 178 |
| 9 | 3412C | 150 | 2090 | 56 |
| 10 | 3306 | 45 | 2789 | 75 |

| | | | | |
|---|---|---|---|---|
| 11 | 3412C | 150 | 1908 | 78 |
| 12 | 3406 | 65 | 736 | 67 |
| 13 | 3412C | 150 | 2155 | 65 |
| 14 | C13 | 45 | 1756 | 375 |
| 15 | C15 | 65 | 2174 | 150 |
| 16 | 3406 | 65 | 1674 | 137 |
| 17 | 3412C | 150 | 2106 | 90 |
| 18 | C15 | 65 | 1489 | 112 |
| 19 | 3412C | 150 | 2004 | 96 |
| 20 | C13 | 45 | 1198 | 390 |
| 21 | 3412C | 150 | 2237 | 110 |
| 22 | C3.3 | 10 | 2678 | 356 |
| 23 | C18 | 90 | 1043 | 137 |
| 24 | C32 | 100 | 865 | 140 |
| 25 | C18 | 90 | 1265 | 158 |
| 26 | C13 | 45 | 986 | 410 |
| 27 | 3412 | 90 | 4448 | 451 |
| 28 | C18 | 90 | 1456 | 198 |
| 29 | C4.4 | 25 | 378 | 378 |
| 30 | 3412 | 90 | 2478 | 356 |
| 31 | C32 | 100 | 1003 | 75 |
| 32 | C13 | 45 | 748 | 124 |
| 33 | 3412 | 150 | 3675 | 245 |
| 34 | C7.1 | 30 | 1487 | 45 |
| 35 | C15 | 60 | 965 | 78 |
| 36 | 3412 | 90 | 2598 | 157 |
| 37 | C32 | 100 | 983 | 67 |
| 38 | 3406 | 65 | 3498 | 139 |
| 39 | 3412 | 150 | 4598 | 167 |
| 40 | C18 | 90 | 966 | 76 |
| 41 | C15 | 60 | 1983 | 245 |
| 42 | 3412 | 90 | 4576 | 426 |
| 43 | C32 | 100 | 1786 | 125 |
| 44 | C4.4 | 15 | 1897 | 78 |
| 45 | 3412 | 150 | 8765 | 657 |
| 46 | C13 | 45 | 1576 | 376 |
| 47 | 3412 | 150 | 6578 | 156 |
| 48 | 3306 | 45 | 2567 | 98 |

| | | | | |
|---|---|---|---|---|
| 49 | C15 | 60 | 2767 | 254 |
| 50 | 3406 | 65 | 4589 | 345 |
| 51 | C15 | 60 | 2398 | 210 |
| 52 | C18 | 90 | 1576 | 250 |
| 53 | C15 | 60 | 1785 | 298 |
| 54 | 3412 | 90 | 3476 | 534 |
| 55 | C32 | 100 | 3786 | 398 |
| 56 | C18 | 90 | 1235 | 78 |
| 57 | 3516 | 420 | 978 | 56 |
| 58 | 3516 | 420 | 1487 | 101 |
| 59 | 3306 | 45 | 1876 | 47 |
| 60 | 3508 | 320 | 657 | 64 |
| 61 | 3412C | 150 | 2768 | 289 |
| 62 | C13 | 45 | 1876 | 421 |
| 63 | C15 | 65 | 2987 | 234 |
| 64 | 3512 | 320 | 1987 | 257 |
| 65 | C18 | 90 | 1675 | 376 |
| 66 | C15 | 60 | 2865 | 265 |
| 67 | 3412 | 90 | 6574 | 256 |
| 68 | 3412C | 150 | 3478 | 356 |
| 69 | 3516 | 420 | 2354 | 178 |
| 70 | C15 | 65 | 2567 | 139 |
| 71 | C13 | 45 | 1398 | 238 |
| 72 | C18 | 90 | 1345 | 56 |
| 73 | 3412 | 150 | 9765 | 178 |
| 74 | 3412C | 150 | 923 | 89 |
| 75 | 3412 | 90 | 8321 | 134 |

A total of 75 samples were collected; in each sample both audio sample and S·O·S test sample were collected and sent to the laboratory.

## 3.2. Functional Block Diagram

Figure 3.9 presents the functional block diagram of the proposed method, illustrating the key components and their interrelations within the system. It provides a clear and structured overview of the method, describing data flow and procedures in the implementation.

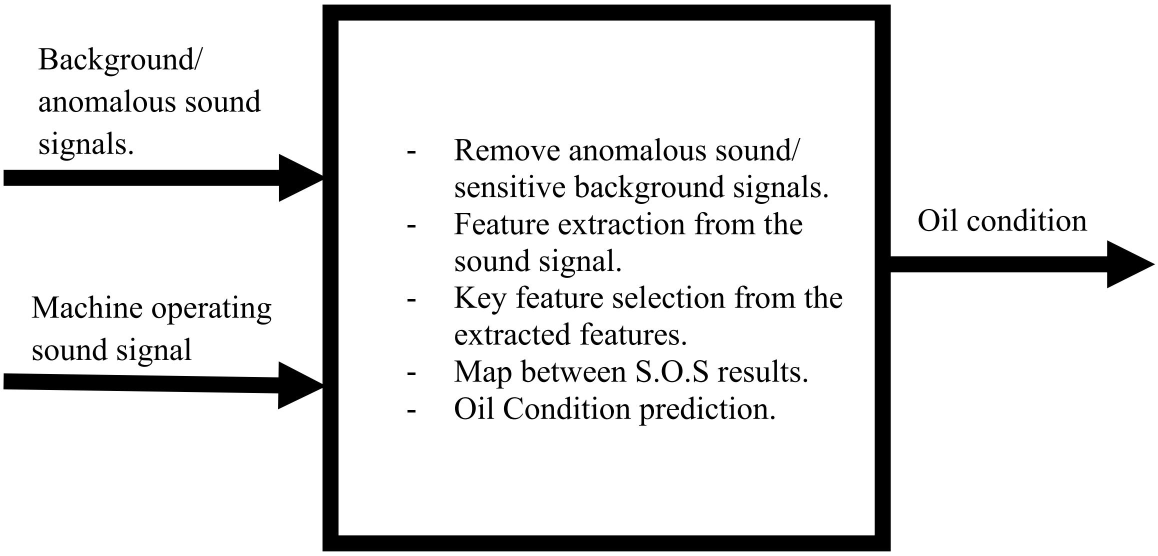



**Figure 3.9:** Functional block diagram of the proposed method

## 3.3. System Overview

As illustrated in Figure 3.10, the system comprises several key modules of the proposed system. The input data, which is extracted from acoustic signals and the S·O·S testing data, passes through functional modules and interactions between various subsystems within the architecture to predict oil condition of the generator.

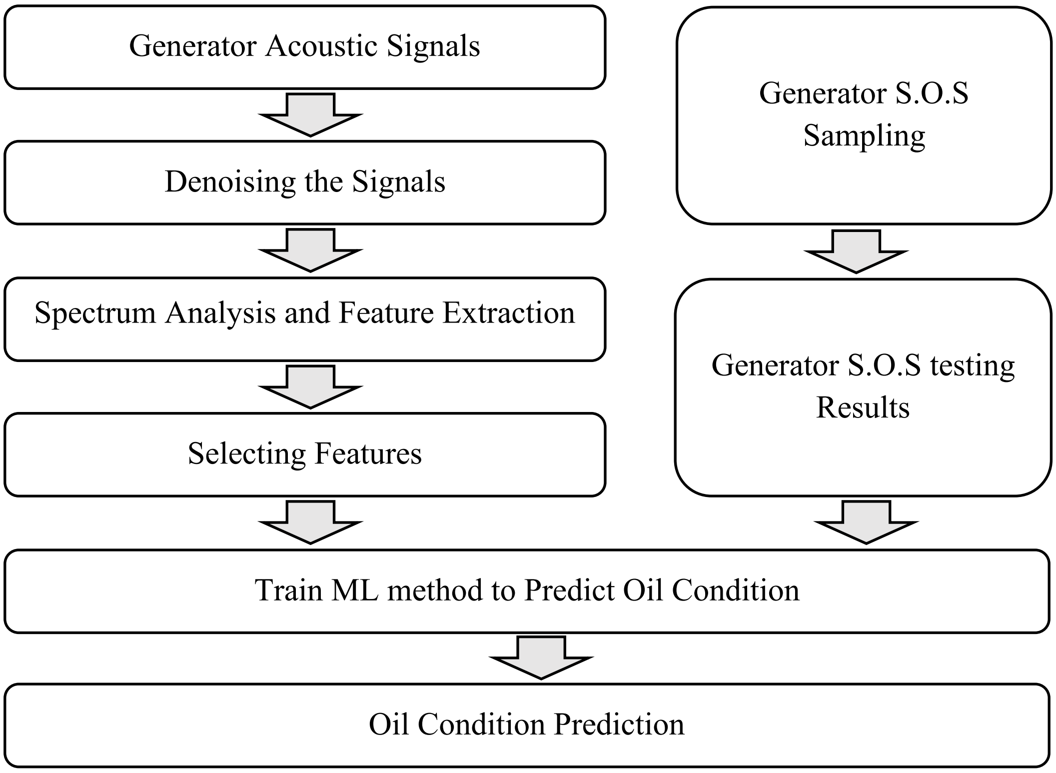



**Figure 3.10:** System overview of the proposed method

## 3.4. Neural Network Architecture for Oil condition prediction

As discussed in section 3.1 generator S·O·S testing sample and the generator acoustic signals were obtained. After collecting S·O·S samples, samples were sent to the laboratory for testing purposes according to the standards. Meantime acoustic features are extracted from the acoustic signal. After receiving the S·O·S testing results and extracted features one complete set of sample created. Those completed sample sets are used to train the neural network.

The architecture of the neural network consists of an input layer of 10 neurons, and there are two hidden layers. The first one of the two hidden layers contains 10 neurons with ReLU activation and the second hidden contains 5 neurons with ReLU activation. In multi-class classification, the network is finished using the output layer containing 3 neurons with ReLU activation, as shown in Figure 3.11.

Parameters in Neural network

i. Activation Function – Compared to other activation functions "identity", "logistic", "tanh". ReLU is very fast to compute because it's a simple operation which means computationally efficient. Unlike sigmoid and tanh, ReLU doesn't squash values into small ranges, which helps gradients remain significant and improves learning in deep networks. Also, ReLU introduces the non-linearity required to learn complex patterns. In the meantime, it outputs zero for all negative values, it naturally creates sparse representation, which can improve generalization and reduce overfitting. In practice ReLU work better and train faster than sigmoid or tanh activation function.

ii. Solver – The weight optimization solver "lbfgs" was employed. The "adam" solver does go pretty fast on pretty big datasets, thousands of examples, as far as training and validation of the neural network are concerned. But on small datasets, "lbfgs" could be faster and also more performing. In addition, "lbfgs" belongs to the quasi-newton optimizer family.

iii. Alpha – strength of the L2 regularization, also called Ridge Regularization or weight decay. This technique is used to prevent overfitting by discouraging large weights in your model. Default value is 0.0001.

iv. batch_size – Batch size parameter with integer 'auto' default. Minibatch size for stochastic optimizers. The classifier won't utilize minibatch if the solver isn't 'lbfgs'. The batch_size=min(200, n_samples) if 'auto'.

v. Learning rate – weight update learning rate schedule. Learning rate "Constant", at "invscaling", "adaptive". Default learning rate = "constant". Constant constant learning rate defined by learning_rate_init, Invscaling decreases learning rate gradually at each time step, adaptive does not change learning rate, remains constant as ‘learning_rate_init’ as long as loss on training decreases. Adaptive is generally used only if solver = “sgd”.

vi. Learning_rate_init – default value = 0.001, to initialize the step-size used to update the weights. It is only applicable when solver = ‘sgd’ or ‘adam’.

vii. Max_iter – int, default =200. Maximum number of iterations. The solver iterates to convergence or up to this number of iterations.

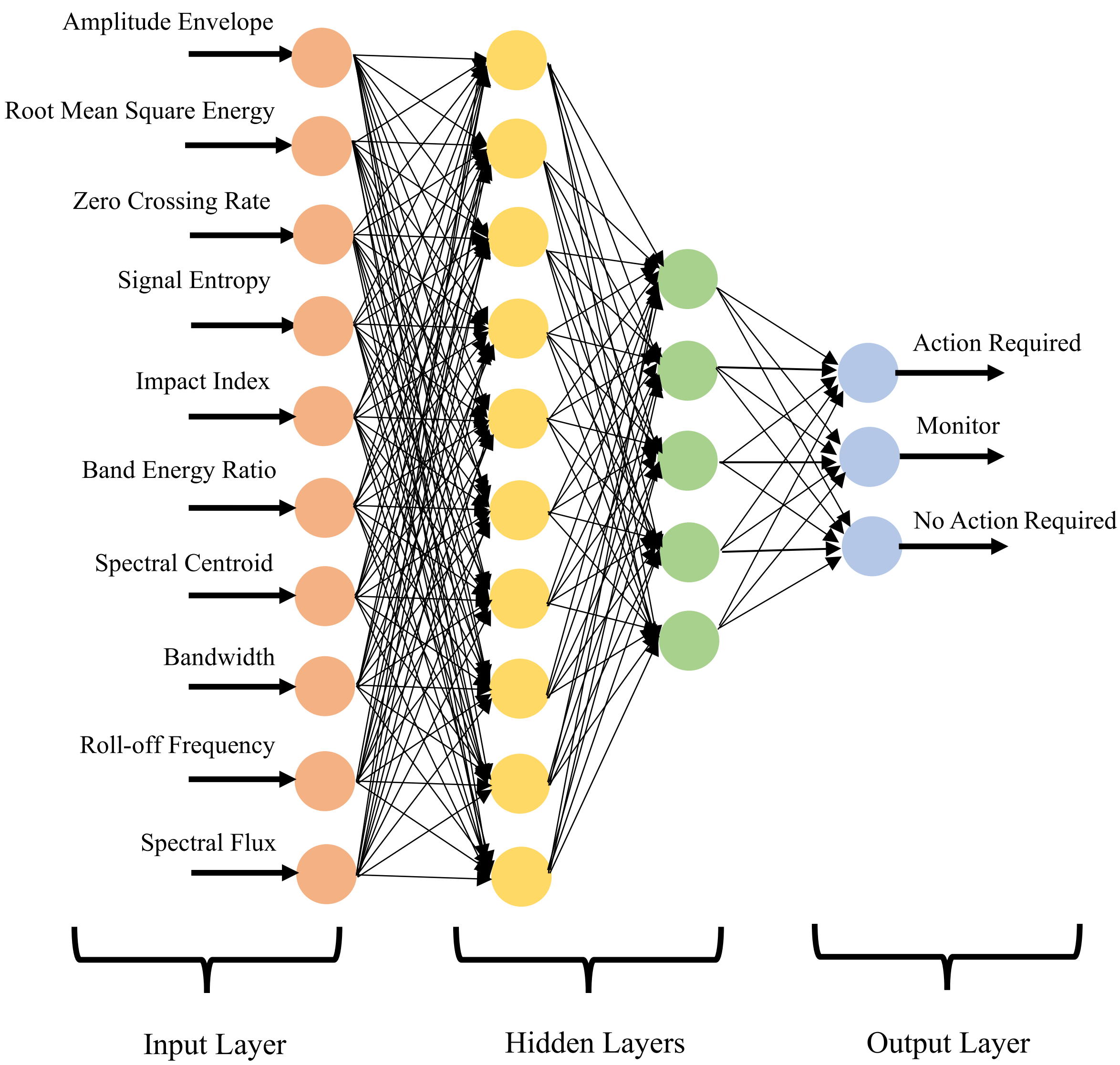



**Figure 3.11:** Neural Network Architecture

## CHAPTER 4

## RESULTS AND DISCUSSION

This chapter represents the results obtained from the proposed method. After completing the full sample data sets, by using the extracted key audio features values and respective S·O·S test results, as mentioned in section 3.5, sample space was optimized for the training of specified neural network to achieve desired outcomes.

### 4.1. Preprocessing and Visualization of Audio signal

For better understanding, visualization of audio sample 7, shown in Figure 4.1, which shows Time Domain signal visualization. The waveform is tightly packed and even, showing a continuous and high-energy sound signal. There are no significant gaps or sudden peaks, which might imply a steady-state signal rather than an impulsive or transient sound.

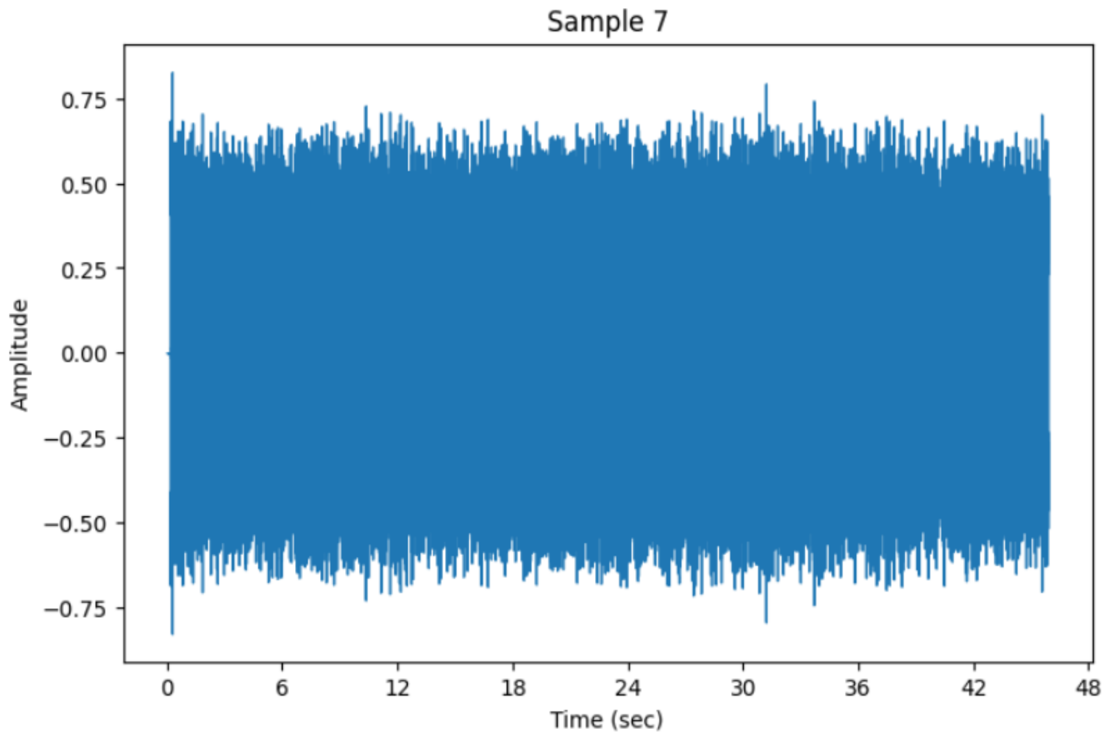



**Figure 4.1:** Acoustic Signal Sample 7 Time Domain signal visualization

*Note:* Time Domain signal visualization using Librosa library.

As discussed in Chapter 2, Time domain reflects how a signal varies with time, but Fast Fourier Transformation (FFT) transforms the signal into the frequency domain such that its frequency components are apparent. It is useful in identifying periodicities, base frequencies, and hidden patterns that are not easily visible in the time domain. The frequency domain visualization of audio sample 7, shown in Figure 4.2, by using FFT.

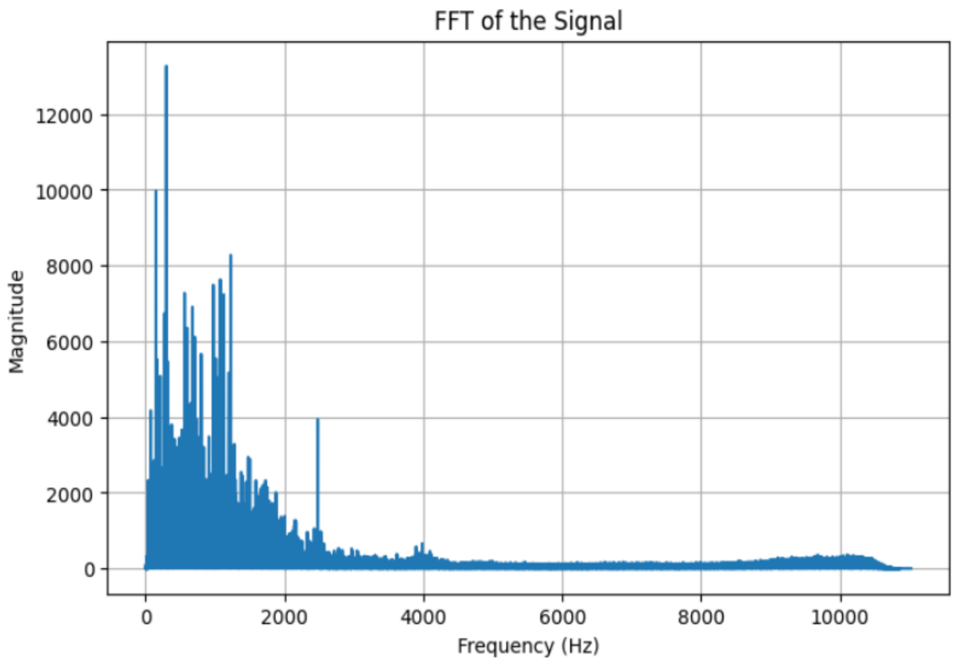



**Figure 4.2:** Acoustic Signal Sample 7 Frequency Domain signal visualization

*Note:* Frequency Domain signal visualization using Librosa library.

In a generator system, there are more than one rotating component, for instance engine and generator rotating parts, turn at different frequencies. Therefore, desired and undesired frequencies both exist in the acoustic sample and are part of the overall spectral composition of the acoustic sample.

As discussed in Chapter 2, a low-pass filter was employed to mitigate high-frequency noise components present in the generator audio signals, including electrical interference and background noise. This preprocessing step was essential for enhancing signal clarity and improving feature extraction accuracy. Figure 4.4 presents the comparison between the original (noisy) and filtered signals of audio sample 7 in the time domain, while Figure 4.5 illustrates the same comparison in the frequency domain.

Following the preprocessing of all samples, key features in both the time and frequency domains were extracted from the audio signals corresponding to each generator instance, as detailed in Chapter 2.

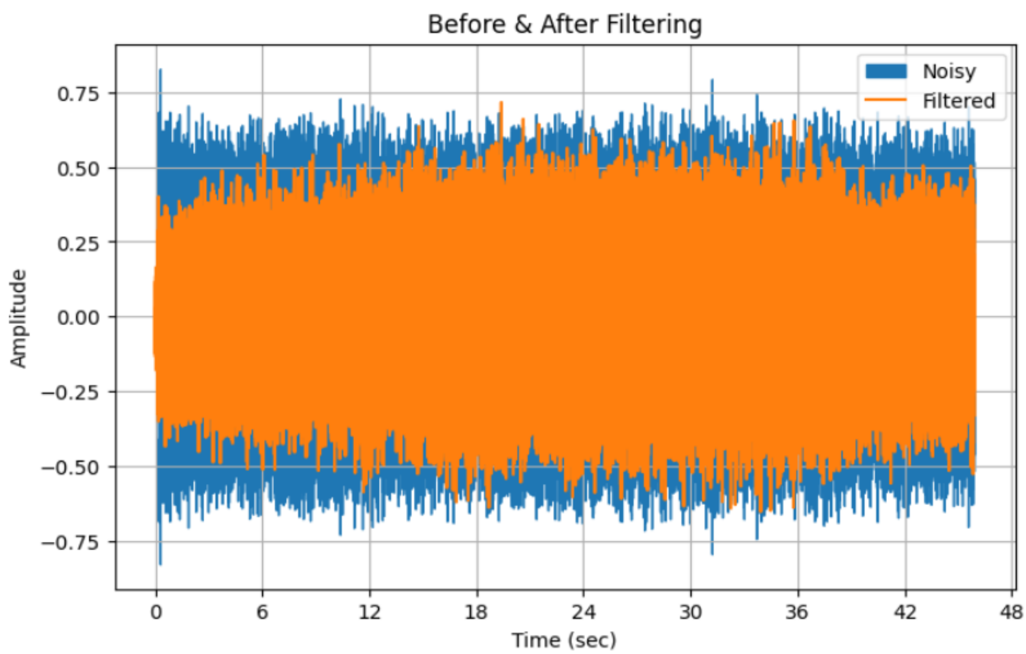



**Figure 4.3:** Filtered and Noisy Signal of sample 7 in Time Domain after preprocessing

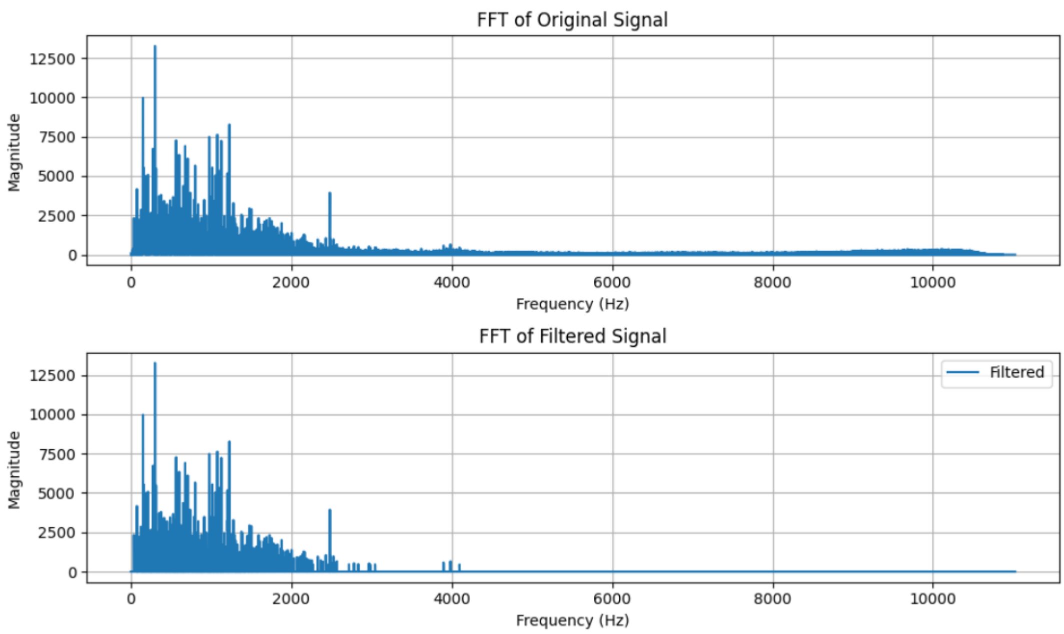



**Figure 4.4:** Filtered and Noisy Signal of sample 7 in Frequency Domain after preprocessing

These features were selected for training and validating the machine learning system due to their complementary characteristics: time domain features capture the instantaneous variations in the signal, while frequency domain features reflect its repetitive patterns and harmonic content. The integration of both domains enhances the robustness of acoustic signal-based condition monitoring by improving diagnostic accuracy and resilience to noise.

## 4.2. Visualization of Features

Specifically, the selected time domain features included Amplitude Envelope, Root Mean Square Energy (RMSE), Zero Crossing Rate (ZCR), Signal Entropy, and Impact Index. From the frequency domain, Band Energy Ratio (BER), Spectral Centroid (SC), Bandwidth, Spectral Roll-off Frequency, and Spectral Flux were utilized.

### 4.2.1. Time Domain Features

a. Amplitude Envelope

Figure 4.5 shows Amplitude envelope of audio sample 7 by using Librosa library. Time is on the x-axis and amplitude is on the y-axis. The blue line is for pre-processed audio signal, and the red line is for extracted amplitude envelope, representing energy changes in the signal with respect to time.

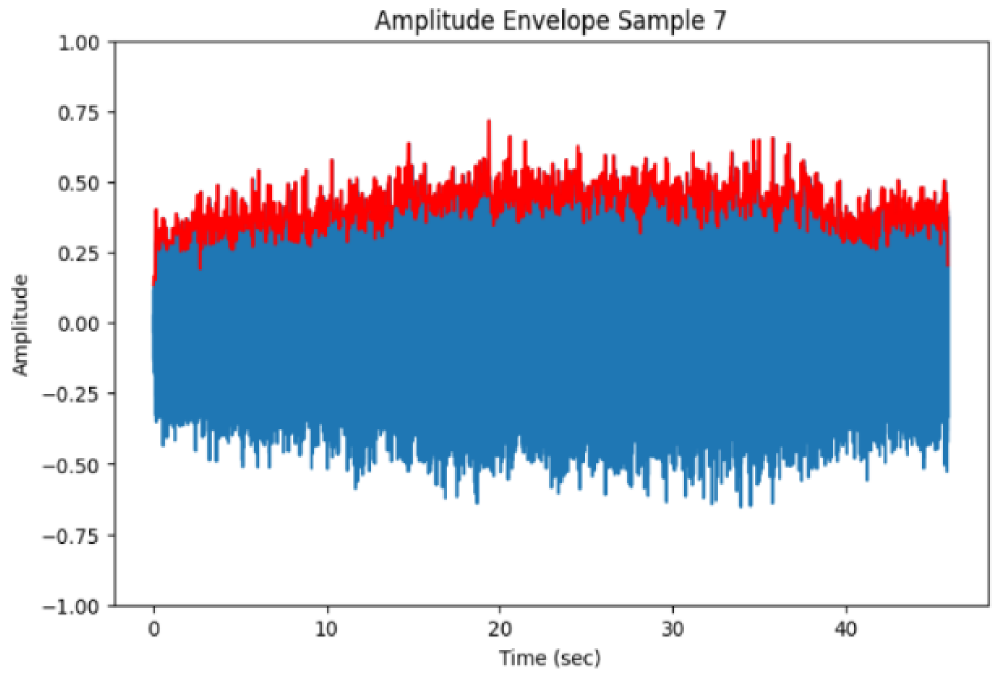



**Figure 4.5:** Amplitude Envelope of Sample 7

b. Root Mean Square Energy (RMSE)

Figure 4.6 shows the Root Mean Square Energy (RMSE) of audio Sample 7, computed using Librosa library. Original audio signal is in blue and RMS energy is in red, indicating energy fluctuation in the signal over time. Time is on the x-axis and amplitude is on the y-axis.

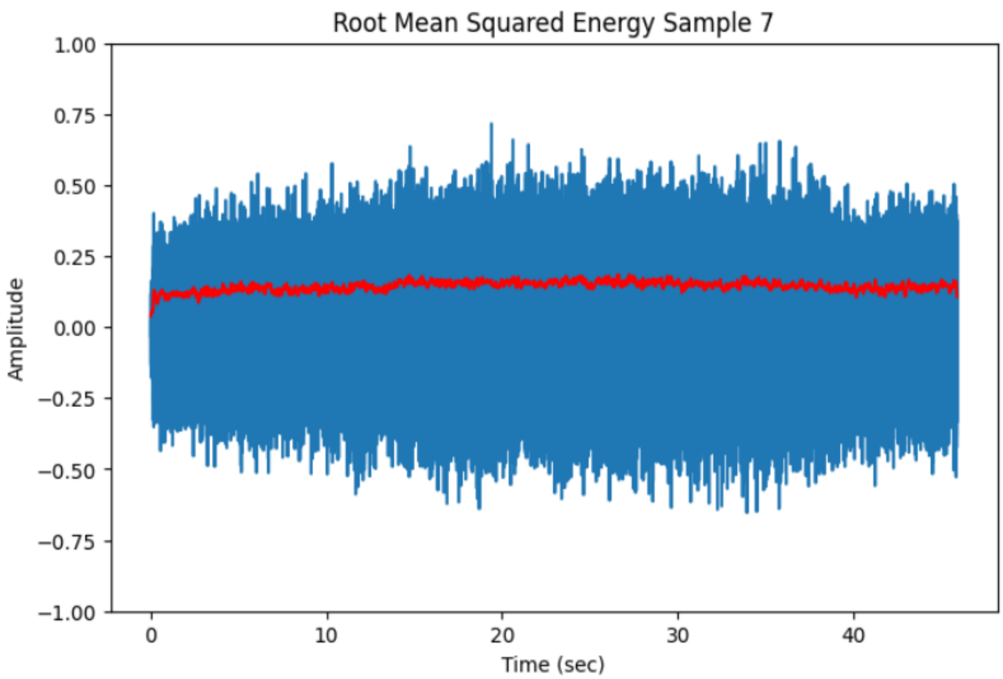



**Figure 4.6:** Root Mean Square Energy of Sample 7

c. Zero Crossing Rate (ZCR)

Figure 4.7 shows the Zero Crossing Rate (ZCR) of audio Sample 7 computed using the Librosa library. The x-axis denotes time in seconds, while the y-axis represents the ZCR amplitude. The analysis was performed to examine variations in signal activity over time.

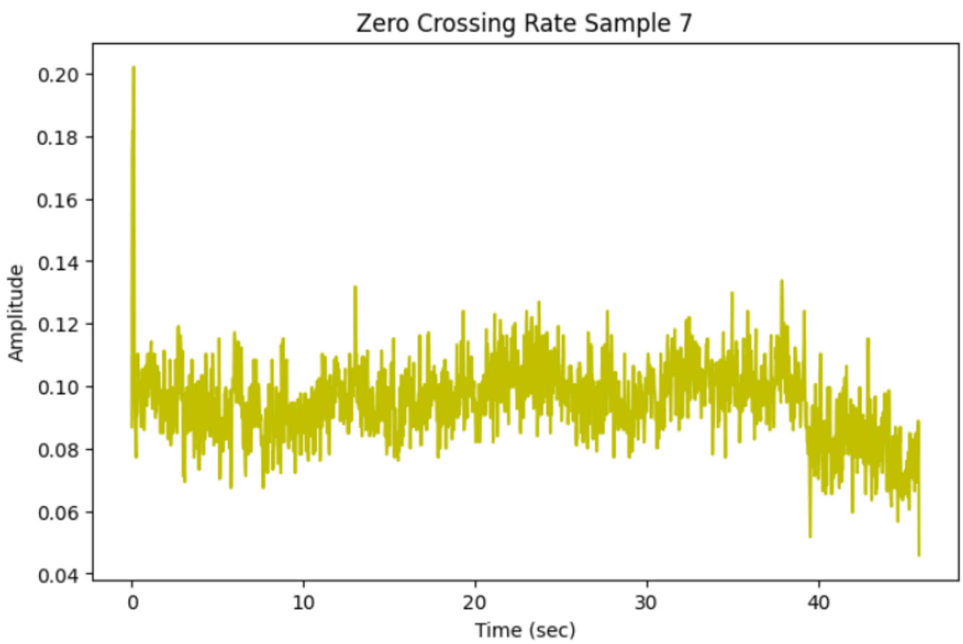



**Figure 4.7:** Zero Crossing Rate of Sample 7

### 4.2.2. Frequency Domain Features

a. Band Energy

Figures 4.8 shows Band Energy Ratio (BER) of audio sample 7, computed using the Librosa library. The x-axis denotes time in seconds, while the y-axis represents the BER amplitude. The analysis was performed to examine variations in signal activity over time.

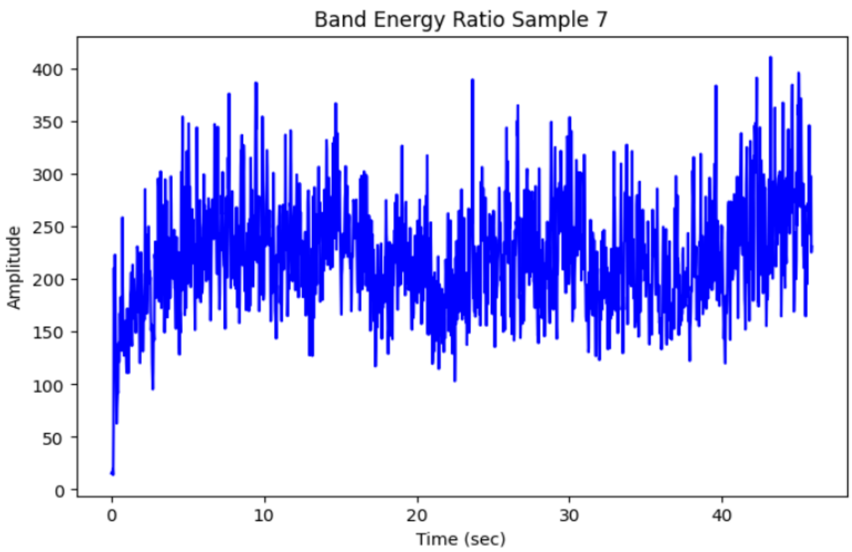



**Figure 4.8:** Band Energy Ratio of Sample 7

b. Spectral Centroid

Figure 4.9 shows the Spectral centroid analysis of audio sample 7 over time. The spectral centroid was computed using Librosa library, which provides a measure of the center of mass of the spectrum.

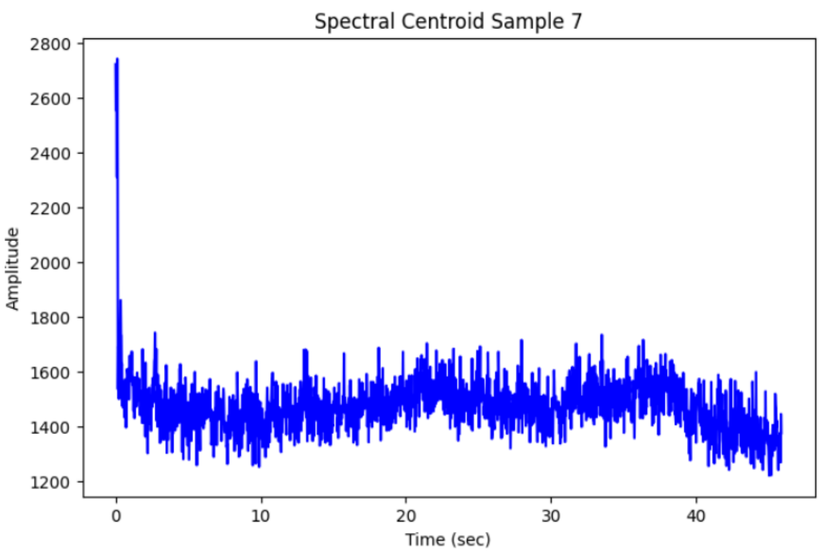



**Figure 4.9:** Spectral Centroid of Sample 7

The plot illustrates how the centroid varies over time. The x-axis represents time in seconds, while the y-axis represents the amplitude of the spectral centroid.

c. Spectral bandwidth

Figure 4.10 shows the Spectral bandwidth of audio sample 7 over time. The spectral bandwidth was computed using Librosa library, which measures the spread of the spectrum around its centroid. The plot illustrates variations in spectral bandwidth. The x-axis represents time in seconds, while the y-axis represents amplitude.

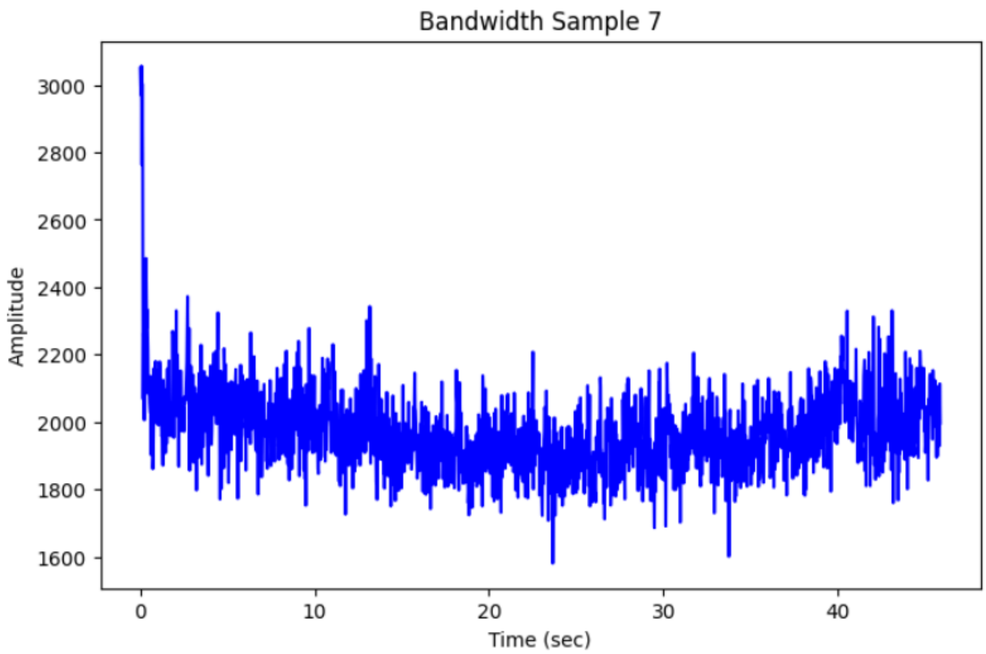



**Figure 4.10:** Spectral bandwidth of Sample 7

d. Spectral Roll-off Frequency

Figure 4.11 shows the Spectral Roll-off Frequency of audio sample 7 over time. The spectral bandwidth was computed using Librosa library. The x-axis represents time in seconds, while the y-axis represents amplitude.

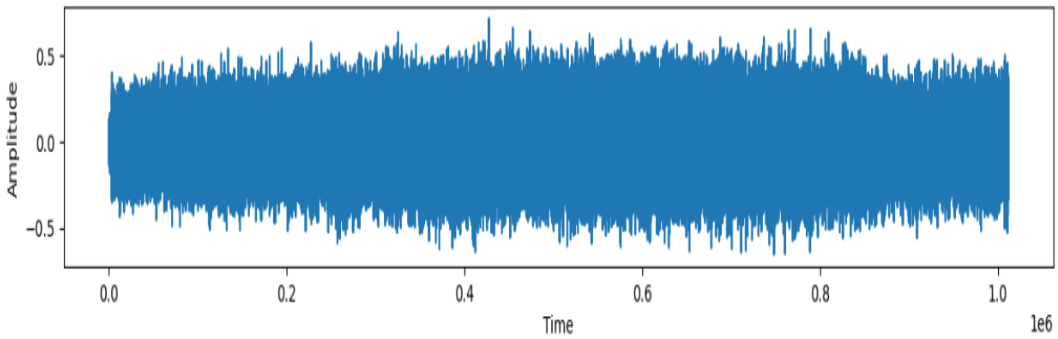



**Figure 4.11:** Spectral Roll-off Frequency of sample 7

e. Spectral Flux

Figure 4.12 illustrates the spectral flux of audio sample 7, computed using Librosa library. Spectral flux measures the rate of change in the power spectrum of an audio signal, providing insights into its dynamic characteristics. The x-axis represents time in seconds, while the y-axis indicates spectral flux values. This analysis helps in understanding variations in signal energy over time.

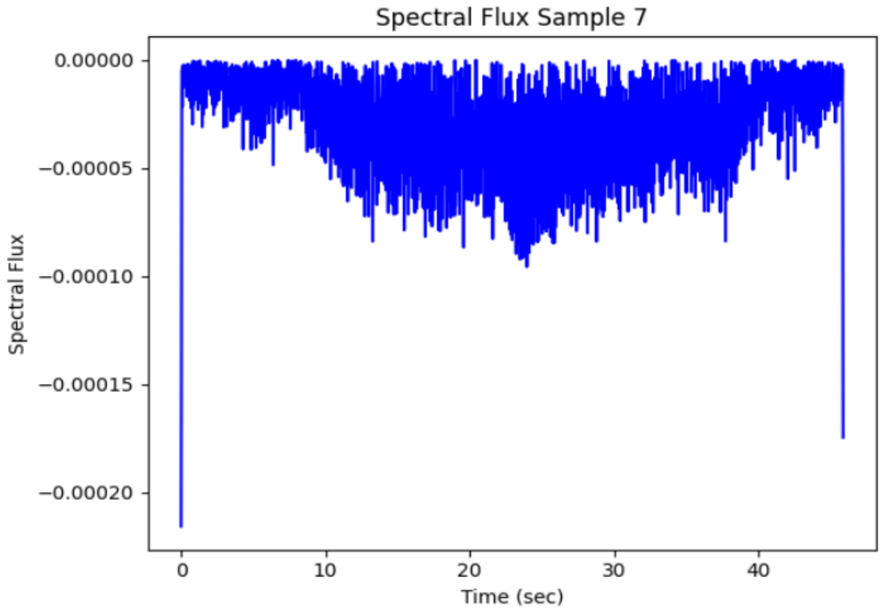



**Figure 4.12:** Spectral Flux of Sample 7

## 4.3. Feature Extraction Summary

As outlined in Chapter 3, a complete sample was defined as one that includes both the S·O·S test results, categorized as Monitor, Action Required, or No Action Required and the corresponding key feature values. In the initial trial, a total of 60 such datasets were compiled, as summarized in Table 4.1.

**Table 4.1:** Sample summary in first trial

| S·O·S Sample Result | Total Samples | No. of samples for Training | No. of samples for Testing |
|---|---|---|---|
| Monitor | 22 | 17 | 5 |
| Action Required | 17 | 12 | 5 |
| No Action Required | 21 | 16 | 5 |
| Total | 60 | 45 | 15 |

Due to the suboptimal performance of the initially trained model, the dataset was expanded and rebalanced to enhance the model's predictive accuracy. Specifically, the number of samples within each S·O·S result category (Monitor, Action Required, and No Action Required) was adjusted to ensure a more uniform distribution, addressing class imbalance which can negatively impact machine learning performance. As a result, the total sample count was increased from 60 to 75. This refinement aimed to provide the model with a more representative and diverse training set, ultimately supporting more robust and generalizable learning. The revised sample distribution is illustrated in Table 4.2.

**Table 4.2:** Sample summary in second trial

| S·O·S Sample Result | Total Samples | No. of samples for Training | No. of samples for Testing |
|---|---|---|---|
| Monitor | 25 | 20 | 5 |
| Action Required | 25 | 17 | 8 |
| No Action Required | 25 | 19 | 6 |
| Total | 75 | 56 | 19 |

## 4.4. Complete Sample Details with Feature Values

**Table 4.3:** Sample details, Acoustic Signal Feature Values and S·O·S Results

| | Gen. Model | Gen. Oil Capacity | SMU | Oil Running Hours | Amplitude Envelope (AE) | Root Mean Square (RMS) Energy | Zero Crossing Rate (ZCR) | Signal Entropy | Impact Index | Band Energy Ratio (BER) | Spectral Centroid (SC) | Bandwidth (BW) | Spectral Roll-off Frequency | Spectral Flux | S.O.S Result |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | C13 | 45 | 1518 | 318 | 0.422044 | 0.158824 | 0.04846 | 7.01994 | 7111 | 0.0236 | 752.011 | 642.9894 | 1465.4913 | -8.73E-06 | M |
| 2 | C18 | 90 | 1189 | 167 | 0.558388 | 0.201254 | 0.08149 | 7.09542 | 10945 | 0.02085 | 1102.86 | 869.3089 | 2134.2122 | -1.70E-05 | M |
| 3 | C13 | 45 | 1311 | 290 | 0.410993 | 0.157793 | 0.05832 | 7.04162 | 7888 | 0.10366 | 824.04 | 713.8501 | 1806.4793 | -3.11E-05 | M |
| 4 | 3406 | 65 | 2567 | 107 | 0.555019 | 0.198825 | 0.10202 | 7.06926 | 12225 | 0.07464 | 1626.47 | 1701.359 | 2507.4906 | -3.04E-05 | M |
| 5 | 3412 | 90 | 3038 | 211 | 0.308114 | 0.10912 | 0.08268 | 6.59648 | 17403 | 1.0472 | 975.908 | 668.8219 | 1704.5113 | -4.05E-05 | M |
| 6 | C15 | 65 | 1645 | 189 | 0.408451 | 0.140114 | 0.09116 | 6.7864 | 26950 | 0.83106 | 1142.77 | 890.3782 | 1995.3497 | -4.36E-05 | M |
| 7 | 3306 | 45 | 3340 | 255 | 0.32964 | 0.121331 | 0.04826 | 6.88498 | 20139 | 5.54E+00 | 679.15 | 563.9284 | 1254.2945 | -2.78E-05 | M |
| 8 | C18 | 90 | 1342 | 178 | 0.44516 | 0.158078 | 0.06553 | 6.87896 | 33614 | 2.22E+00 | 846.436 | 672.5622 | 1571.6539 | -2.88E-05 | M |
| 9 | 3412C | 150 | 2090 | 56 | 0.314287 | 0.109588 | 0.07651 | 6.75208 | 6447 | 0.53243 | 867.699 | 626.0134 | 1461.4291 | -6.52E-06 | N |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 3306 | 45 | 2789 | 75 | 0.414493 | 0.143447 | 0.08746 | 6.98662 | 10542 | 0.35893 | 1155.92 | 1340.704 | 1821.5511 | -2.18E-05 | N |
| 11 | 3412C | 150 | 1908 | 78 | 0.314628 | 0.113473 | 0.05912 | 6.71836 | 8925 | 0.43227 | 754.487 | 515.821 | 1326.0979 | -4.43E-05 | M |
| 12 | 3406 | 65 | 736 | 67 | 0.427152 | 0.150493 | 0.06943 | 6.89251 | 14743 | 0.3223 | 894.01 | 683.5546 | 1595.1593 | -6.27E-05 | M |
| 13 | 3412C | 150 | 2155 | 65 | 0.312185 | 0.105715 | 0.07332 | 6.649 | 12178 | 4.60E+14 | 886.546 | 606.588 | 1528.3128 | -2.51E-05 | M |
| 14 | C13 | 45 | 1756 | 375 | 0.42564 | 0.144119 | 0.08138 | 6.80171 | 21314 | 0.63214 | 1037.86 | 965.3674 | 1665.9805 | -2.83E-05 | M |
| 15 | C15 | 65 | 2174 | 150 | 0.321349 | 0.125428 | 0.03265 | 7.0083 | 17798 | 9.03607 | 501.871 | 401.1335 | 883.54232 | -5.11E-05 | M |
| 16 | 3406 | 65 | 1674 | 137 | 0.414936 | 0.153737 | 0.05019 | 7.00156 | 26528 | 4.13686 | 1005.09 | 1523.031 | 1383.9408 | -5.63E-05 | M |
| 17 | 3412C | 150 | 2106 | 90 | 0.367779 | 0.131151 | 0.16479 | 6.97764 | 12427 | 0.4695 | 1128.8 | 834.4789 | 2257.0734 | -2.50E-05 | M |
| 18 | C15 | 65 | 1489 | 112 | 0.458049 | 0.159275 | 0.15194 | 6.91713 | 17591 | 0.40942 | 1431.1 | 1237.301 | 2477.9276 | -3.92E-05 | M |
| 19 | 3412C | 150 | 2004 | 96 | 0.356789 | 0.122806 | 0.14233 | 6.95651 | 13689 | 0.19027 | 1177.94 | 809.6574 | 1991.0726 | -1.77E-05 | A |
| 20 | C13 | 45 | 1198 | 390 | 0.428223 | 0.143954 | 0.13754 | 6.87877 | 18644 | 0.26155 | 1454.42 | 1340.383 | 2452.471 | -2.44E-05 | A |
| 21 | 3412C | 150 | 2237 | 110 | 0.355812 | 0.128021 | 0.14035 | 6.90552 | 14680 | 0.21841 | 1348.21 | 881.2454 | 2316.7966 | -3.15E-05 | A |
| 22 | C3.3 | 10 | 2678 | 356 | 0.407133 | 0.143734 | 0.13721 | 6.92358 | 18462 | 0.31811 | 1552.38 | 1369.126 | 2518.6596 | -3.73E-05 | A |
| 23 | C18 | 90 | 1043 | 137 | 0.428551 | 0.196673 | 0.0414 | 7.35094 | 41706 | 7.69457 | 740.999 | 553.5329 | 1414.1793 | -1.06E-05 | M |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 24 | C32 | 100 | 865 | 140 | 0.500765 | 0.212451 | 0.04905 | 7.27901 | 47101 | 4.96919 | 894.848 | 722.5807 | 1796.2501 | -1.24E-05 | M |
| 25 | C18 | 90 | 1265 | 158 | 0.491685 | 0.18128 | 0.07542 | 7.10156 | 38099 | 1.51633 | 1034.26 | 698.0384 | 1928.7611 | -1.81E-05 | A |
| 26 | C13 | 45 | 986 | 410 | 0.582737 | 0.207069 | 0.0861 | 7.04371 | 49527 | 1.14397 | 1170.67 | 824.2309 | 2109.5873 | -2.06E-05 | A |
| 27 | 3412 | 90 | 4448 | 451 | 0.413104 | 0.148677 | 0.0654 | 6.81531 | 19658 | 1.47156 | 900.53 | 751.4 | 1752.4436 | -1.27E-05 | M |
| 28 | C18 | 90 | 1456 | 198 | 0.511453 | 0.181104 | 0.07995 | 6.97815 | 28844 | 0.96061 | 1106.92 | 907.3011 | 2144.0069 | -1.47E-05 | M |
| 29 | C4.4 | 25 | 378 | 378 | 0.362181 | 0.145075 | 0.03091 | 7.18688 | 13943 | 6.81681 | 536.467 | 541.7578 | 1272.5939 | -0.00021 | M |
| 30 | 3412 | 90 | 2478 | 356 | 0.414221 | 0.158969 | 0.04848 | 7.17116 | 16705 | 2.90681 | 704.551 | 602.1659 | 1500.8777 | -0.00021 | M |
| 31 | C32 | 100 | 1003 | 75 | 0.322448 | 0.118664 | 0.05464 | 6.33086 | 1E+05 | 2.22E+15 | 737.777 | 540.8698 | 1359.3565 | -1.28E-05 | N |
| 32 | C13 | 45 | 748 | 124 | 0.221718 | 0.087986 | 0.0401 | 6.67893 | 68301 | 3.69E+15 | 578.158 | 422.4362 | 1005.6335 | -8.04E-06 | N |
| 33 | 3412 | 150 | 3675 | 245 | 0.345423 | 0.130051 | 0.04006 | 6.62085 | 1E+05 | 430887 | 569.12 | 459.3412 | 979.43044 | -1.31E-05 | N |
| 34 | C7.1 | 30 | 1487 | 45 | 0.256513 | 0.100482 | 0.03053 | 6.6281 | 60605 | 3.49E+13 | 424.451 | 334.5692 | 710.98185 | -1.02E-05 | N |
| 35 | C15 | 60 | 965 | 78 | 0.341547 | 0.122266 | 0.05423 | 6.65563 | 60069 | 307.345 | 744.612 | 556.6132 | 1244.519 | -1.75E-05 | N |
| 36 | 3412 | 90 | 2598 | 157 | 0.261074 | 0.095365 | 0.04907 | 6.7254 | 36975 | 787.389 | 682.454 | 471.1853 | 1132.0856 | -1.50E-05 | N |
| 37 | C32 | 100 | 983 | 67 | 0.368667 | 0.128541 | 0.04623 | 6.74064 | 28923 | 1.09E+01 | 631.296 | 468.6129 | 1043.4519 | -8.85E-06 | N |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 38 | 3406 | 65 | 3498 | 139 | 0.295372 | 0.103352 | 0.04106 | 6.76794 | 19101 | 2.05E+01 | 509.811 | 312.6727 | 848.20212 | -6.57E-06 | N |
| 39 | 3412 | 150 | 4598 | 167 | 0.332947 | 0.122097 | 0.05068 | 6.55052 | 28576 | 1.15E+01 | 668.358 | 522.8276 | 1109.8979 | -9.30E-06 | N |
| 40 | C18 | 90 | 966 | 76 | 0.213495 | 0.082062 | 0.0367 | 6.76793 | 12883 | 1.53E+02 | 449.977 | 372.6308 | 659.73912 | -4.53E-06 | N |
| 41 | C15 | 60 | 1983 | 245 | 0.234483 | 0.08931 | 0.03621 | 7.05305 | 1555 | 0.50873 | 585.865 | 548.5201 | 1179.59 | -2.73E-05 | A |
| 42 | 3412 | 90 | 4576 | 426 | 0.192031 | 0.076664 | 0.02321 | 7.11292 | 1201 | 0.65272 | 365.667 | 343.5431 | 666.47029 | -6.93E-06 | A |
| 43 | C32 | 100 | 1786 | 125 | 0.318072 | 0.121038 | 0.04452 | 6.68826 | 94505 | 8.85E+13 | 647.451 | 487.328 | 1133.7387 | -1.41E-05 | N |
| 44 | C4.4 | 15 | 1897 | 78 | 0.244456 | 0.098458 | 0.04027 | 6.71188 | 63621 | 5.50E+14 | 569.864 | 375.38 | 985.70586 | -7.41E-06 | N |
| 45 | 3412 | 150 | 8765 | 657 | 0.325521 | 0.121411 | 0.04777 | 6.71849 | 40821 | 4.81E+01 | 663.088 | 504.7906 | 1203.3963 | -1.37E-05 | A |
| 46 | C13 | 45 | 1576 | 376 | 0.254432 | 0.098148 | 0.04118 | 6.8585 | 26819 | 4.34E+02 | 534.558 | 373.8491 | 939.77167 | -9.17E-06 | A |
| 47 | 3412 | 150 | 6578 | 156 | 0.518895 | 0.176492 | 0.08648 | 6.76622 | 74207 | 5.34E+00 | 1160.29 | 1101.844 | 2002.2488 | -0.00183 | N |
| 48 | 3306 | 45 | 2567 | 98 | 0.43546 | 0.151521 | 0.07066 | 6.88037 | 55069 | 9.26E+00 | 912.753 | 860.3396 | 1490.3862 | -0.0012 | N |
| 49 | C15 | 60 | 2767 | 254 | 1.303277 | 0.924686 | 0.08461 | 6.92873 | 8E+06 | 31.4948 | 2365.4 | 2320.412 | 4914.923 | -0.03005 | A |
| 50 | 3406 | 65 | 4589 | 345 | 1.485037 | 0.892886 | 0.07207 | 7.30778 | 7E+06 | 263.82 | 1519.73 | 1376.463 | 2965.7445 | -0.02967 | A |
| 51 | C15 | 60 | 2398 | 210 | 1.603263 | 0.679431 | 0.04208 | 7.14257 | 4E+06 | 5.73E+15 | 625.773 | 473.0766 | 1117.7579 | -0.01315 | A |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 52 | C18 | 90 | 1576 | 250 | 1.313244 | 0.875984 | 0.11701 | 7.24944 | 6E+06 | 12.5034 | 2682.79 | 2482.856 | 5626.3162 | -0.0527 | A |
| 53 | C15 | 60 | 1785 | 298 | 1.512486 | 0.837289 | 0.09346 | 7.36638 | 5E+06 | 46.2398 | 1778.04 | 1587.335 | 3312.5699 | -0.05267 | A |
| 54 | 3412 | 90 | 3476 | 534 | 1.659476 | 0.771277 | 0.07731 | 7.32853 | 4E+06 | 994.336 | 1223.85 | 1001.258 | 2423.5491 | -0.05208 | A |
| 55 | C32 | 100 | 3786 | 398 | 1.662102 | 0.716425 | 0.06783 | 7.19574 | 4E+06 | 20227.3 | 1053.45 | 868.0379 | 2166.0489 | -0.05092 | A |
| 56 | C18 | 90 | 1235 | 78 | 1.310912 | 0.910207 | 0.0853 | 7.00977 | 7E+06 | 20.673 | 2473.87 | 2497.749 | 5418.6236 | -0.03863 | N |
| 57 | 3516 | 420 | 978 | 56 | 1.465911 | 0.87492 | 0.06382 | 7.33514 | 6E+06 | 139.411 | 1481.85 | 1472.075 | 2925.3992 | -0.03842 | N |
| 58 | 3516 | 420 | 1487 | 101 | 1.568485 | 0.826132 | 0.05039 | 7.35497 | 5E+06 | 4057.95 | 952.913 | 864.2432 | 1853.7905 | -0.03717 | N |
| 59 | 3306 | 45 | 1876 | 47 | 1.3249 | 0.902839 | 0.11296 | 7.11007 | 6E+06 | 11.2175 | 2823.32 | 2641.373 | 5994.1888 | -0.02364 | N |
| 60 | 3508 | 320 | 657 | 64 | 1.511996 | 0.86224 | 0.08904 | 7.31441 | 5E+06 | 37.814 | 1852.84 | 1722.132 | 3552.1905 | -0.02292 | N |
| 61 | 3412C | 150 | 2768 | 289 | 0.356789 | 0.122806 | 0.14233 | 6.95651 | 13689 | 0.19027 | 1177.94 | 809.6574 | 1991.0726 | -1.8E-05 | A |
| 62 | C13 | 45 | 1876 | 421 | 0.428223 | 0.143954 | 0.13754 | 6.87877 | 18644 | 0.26155 | 1454.42 | 1340.383 | 2452.471 | -2.4E-05 | A |
| 63 | C15 | 65 | 2987 | 234 | 0.355812 | 0.128021 | 0.14035 | 6.90552 | 14680 | 0.21841 | 1348.21 | 881.2454 | 2316.7966 | -3.2E-05 | A |
| 64 | 3512 | 320 | 1987 | 257 | 0.407133 | 0.143734 | 0.13721 | 6.92358 | 18462 | 0.31811 | 1552.38 | 1369.126 | 2518.6596 | -3.7E-05 | A |
| 65 | C18 | 90 | 1675 | 376 | 0.491685 | 0.18128 | 0.07542 | 7.10156 | 38099 | 1.51633 | 1034.26 | 698.0384 | 1928.7611 | -1.8E-05 | A |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 66 | C15 | 60 | 2865 | 265 | 0.582737 | 0.207069 | 0.0861 | 7.04371 | 49527 | 1.14397 | 1170.67 | 824.2309 | 2109.5873 | -2.1E-05 | A |
| 67 | 3412 | 90 | 6574 | 256 | 0.234483 | 0.08931 | 0.03621 | 7.05305 | 1555 | 0.50873 | 585.865 | 548.5201 | 1179.59 | -2.7E-05 | A |
| 68 | 3412C | 150 | 3478 | 356 | 0.192031 | 0.076664 | 0.02321 | 7.11292 | 1201 | 0.65272 | 365.667 | 343.5431 | 666.47029 | -6.9E-06 | A |
| 69 | 3516 | 420 | 2354 | 178 | 0.312185 | 0.105715 | 0.07332 | 6.649 | 12178 | 4.60E+14 | 886.546 | 606.588 | 1528.3128 | -2.51E-05 | M |
| 70 | C15 | 65 | 2567 | 139 | 0.42564 | 0.144119 | 0.08138 | 6.80171 | 21314 | 0.63214 | 1037.86 | 965.3674 | 1665.9805 | -2.83E-05 | M |
| 71 | C13 | 45 | 1398 | 238 | 0.321349 | 0.125428 | 0.03265 | 7.0083 | 17798 | 9.03607 | 501.871 | 401.1335 | 883.54232 | -5.11E-05 | M |
| 72 | C18 | 90 | 1345 | 56 | 0.345423 | 0.130051 | 0.04006 | 6.62085 | 1E+05 | 430887 | 569.12 | 459.3412 | 979.43044 | -1.31E-05 | N |
| 73 | 3412 | 150 | 9765 | 178 | 0.256513 | 0.100482 | 0.03053 | 6.6281 | 60605 | 3.49E+13 | 424.451 | 334.5692 | 710.98185 | -1.02E-05 | N |
| 74 | 3412C | 150 | 923 | 89 | 0.341547 | 0.122266 | 0.05423 | 6.65563 | 60069 | 307.345 | 744.612 | 556.6132 | 1244.519 | -1.75E-05 | N |
| 75 | 3412 | 90 | 8321 | 134 | 0.261074 | 0.095365 | 0.04907 | 6.7254 | 36975 | 787.389 | 682.454 | 471.1853 | 1132.0856 | -1.50E-05 | N |

*Note*: A – Action Required, M – Monitor, N – No Action Required

## 4.5. Results

By using table 4.3, extracted feature values and S·O·S results, neural network model was trained to predict the oil condition. The performance of the neural network was evaluated by using both confusion matrix and key classification metrics, precision, recall and F1 – score. The confusion matrix presents the classification performance of the neural network across predicting three categories, “Action required”, “Monitor”, and “No Action Required”. In key classification metrics precision represents the proportion of the number of cases correctly predicted for each category, recall equivalents to the ability of the model to correctly identify real instances in each class and f1-score represents the harmonic mean of precision and recall. Considering these key metrics, overall model performance is measured by overall accuracy.

i. First trial using 60 sample data

During the first trial of training the Neural Network, table 4.1 shows the sample data distribution. Out of 60 sample data, 45 sample data were used to train the neural network, and 15 were used for testing purposes. Table 4.4 shows the key classification metric values of trained neural network using 60 samples.

**Table 4.4:** Key classification metric values of trained NN using 60 samples

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Action Required | 0.80 | 0.80 | 0.80 | 5 |
| Monitor | 1.00 | 0.40 | 0.57 | 5 |
| No Action Required | 0.50 | 0.80 | 0.62 | 5 |
| | | | | |
| Accuracy | | | 0.67 | 15 |
| Macro avg | 0.77 | 0.67 | 0.66 | 15 |
| Weighted avg | 0.77 | 0.67 | 0.66 | 15 |

*Note*: Key classification matric using librosa library

With regards to the classification metric values obtained, as shown in table 4.4, highest precision, 1.00 is obtained in the “Monitor” category and there’s no false positives. Even though, “No Action Required” class has the lowest precision of 0.5, which means half of the category were incorrect. The “Action Required" and “No Action Required” both categories have high recall value of 0.80, while category “Monitor” has a recall value of 0.40, which indicates that a significant number of “Monitor” cases were false or misclassified. The harmonic mean of precision and recall, F1-score shows highest value 0.80 in “Action Required” and lowest of 0.57 in “Monitor” category, which indicates a need of improvement in “Monitor” category compared to other categories.

Overall model accuracy is 0.67, when using 60 samples for training and validation, this result emphasizes that two-thirds of the total samples were classified into correct categories. Overall results suggest that above trained model using 60 samples performs reasonably well for “Action Required” category and “Monitor” category needs an enhancement. Therefore, further refinement was required to enhance the predictive reliability of Oil condition.

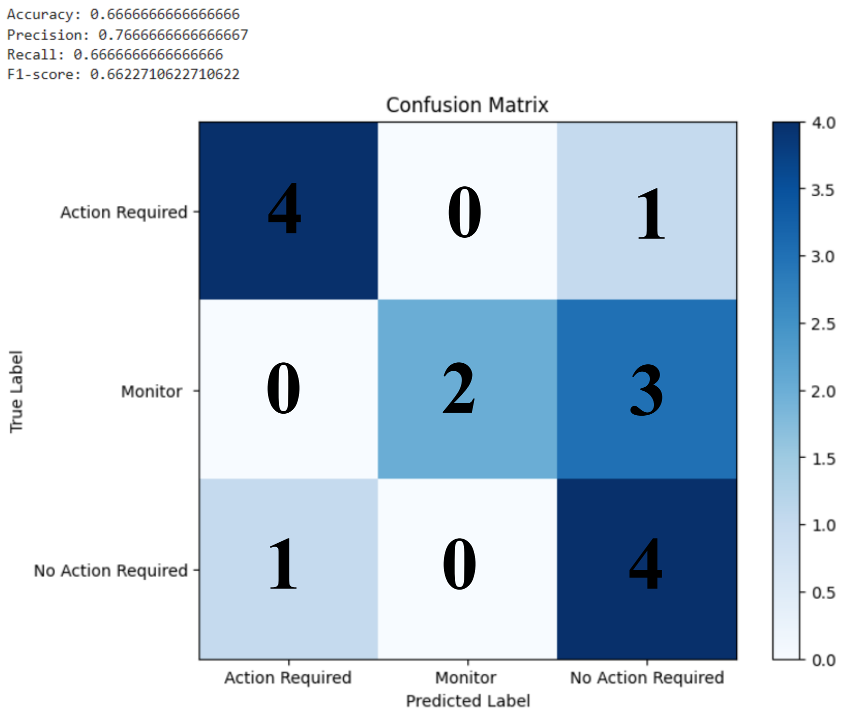



**Figure 4.13:** First attempt Confusion Matrix

*Note*: Confusion metric using Librosa Library


**Figure 4.14:** Training and Validation loss of First Trial

*Note*: Loss curves using Librosa Library

The confusion metric provides the classification performance of the machine learning model. The colour intensity in every cell is a measure of the number of predictions, where darker cells represent higher values, and lighter cells represent less values. The matrix contains the most important classification performance metrics also. The value on each cell shows the number of correct categories predicted from the model. In this attempt there were five samples for testing in each category. For instance, out of 5 samples in "Action Required" category four were predicted correctly from the trained model, and one was predicted as "No Action Required" as a false prediction.

As per Figure 4.13, confusion matrix shows the model has moderate classification accuracy of 0.67 and it presents misclassified instances, which suggests the need for further optimization, tunning of the model and expansion of the training data set.

When considering Figure 4.14, inspection of the loss curves shows the model demonstrates stability and consistency within the training process. The observed trend, however, shows the model may not be successfully gaining sufficient features to enhance validation performance. The apparent discrepancy in the relationship between training and validation loss suggests a possible underfitting issue or limited generalization capability. The fact that the validation loss trajectory is quite flat also indicates that the model's capability has leveled off. This provides a rationale for additional optimization since there is significant potential in enhancing the ability of the model to well represent and generalize the essential patterns within the data.

The results indicate that, although a satisfactory baseline model was established, there remains a need for further enhancement, particularly in reducing misclassification rates and improving overall predictive performance.

ii. Second trial using 75 sample data

As per the results from the model, which trained by 60 samples, it suggests that to improve the overall accuracy minimizing misclassification along with the prediction of oil condition. Therefore, sample size increased to 75 samples for training and validation of the model. As shown in table 4.1 in the first attempt of training the neural network model there were 45 samples to train the model and 15 for validation, in the second attempt as shown in table 4.2 number of samples for training increased to 56 samples and number of samples for testing increase to 19 samples.

**Table 4.5:** Key classification metric values of trained NN using 75 samples

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Action Required | 1.00 | 0.80 | 0.89 | 5 |
| Monitor | 0.83 | 0.62 | 0.71 | 8 |
| No Action Required | 0.56 | 0.83 | 0.67 | 6 |
| | | | | |
| Accuracy | | | 0.74 | 19 |
| Macro avg | 0.80 | 0.75 | 0.76 | 19 |
| Weighted avg | 0.79 | 0.74 | 0.75 | 19 |

As shown in table 4.4 and table 4.5, the classification performance or the quality of the two models can be evaluated by using key metrics, encompassed with precision, recall and F1-score along with the overall accuracy of both models. When it comes to precision analysis, in the first model (60 samples model) 'Monitor" category recorded a highest precision, 1.00 followed by lowest precision, 0.50 in case of "No Action Required" while second model (75 samples model) highest with "Action Required", 1.00 and lowest with that with "No Action Required" of 0.56. By comparing both model values second model has a more balanced distribution of precision values in all the categories as opposed to the first model.

When considering classification matric of recall analysis, in the first model recall values ranged from 0.40 ("Monitor") to 0.80 ("No Action Required", "Action Required"). In the second model the recall values enhanced for both "Monitor" (0.62) and "No Action Required" (0.83), with that of "Action Required" staying at 0.80. So, as values presents in the second model recall improved in terms of categories "Monitor" and "No Action Required".

The harmonic mean of precision and recall, f1-score demonstrated an improvement across all the categories, indicating better overall performance between precision and recall. F1-score of first model ranged from 0.57 ("Monitor") to 0.80 ("Action Required") while in the second model ranged from 0.67 ("No Action Required") to 0.89 ("Action Required").

Overall, the second model showed an overall accuracy improvement of 0.07, from 0.67 to 0.74

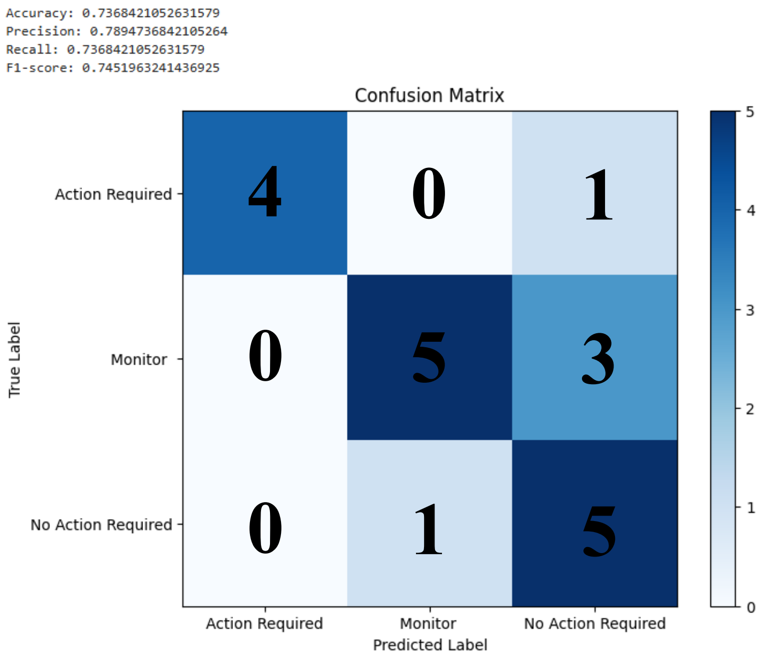



**Figure 4.15** Second attempt Confusion Matrix

*Note*: Confusion metrics using Librosa Library

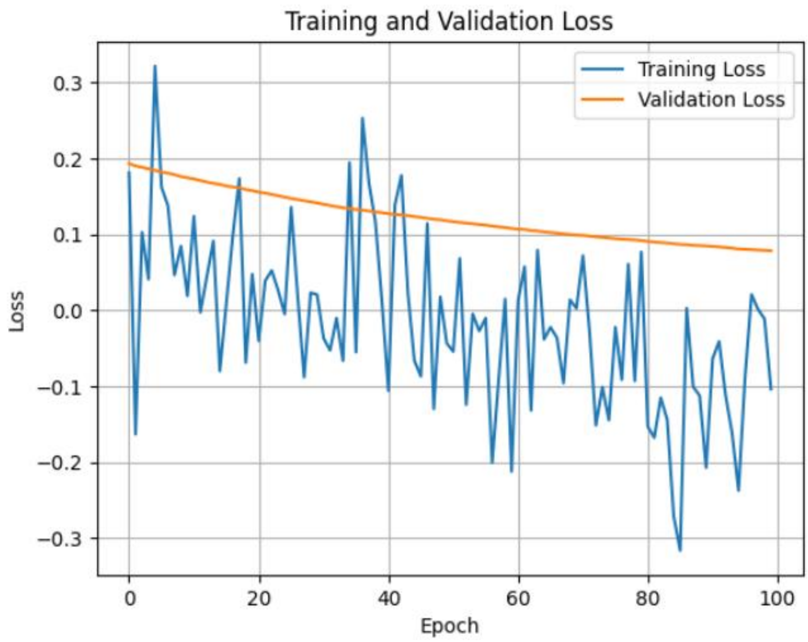



**Figure 4.16** Training and Validation loss of Second Trial

*Note*: Loss curves using Librosa Library

Figure 4.15 and figure 4.16 presents a staggering improvement in all performances, precision, recall and F1-score. The improvement in classification in the category of "Monitor" reflects enhanced discrimination between this category and the other two categories. The model is still committing small classification errors, in discrimination between "No Action Required" and "Monitor", though overall classification error is reduced.

The analysis of validation curves and training provides essential information regarding performance and generalizability. Figure 4.17 illustrates training and validation over 100 epochs to allow observation regarding pattern in learning by the model and stability of the second model.

The training loss is clearly reducing in training, which is an indication that indeed the model is improving during training over the dataset. The training loss also oscillates with some large fluctuations, particularly in early epochs, which is a warning signal towards non-stability in the optimization process. On the validation side, validation loss has smoother reducing trend is an indication towards better stable generalization to unseen samples. No steep rise in validation loss is an indication that certainly the model is not overfitting considerably because optimally working over unseen samples.

The substantial difference in training loss is owing to factors including steep learning rates or small batch sizes. Difference in this area means that sudden changes in weights in the model can interfere with the process of convergence. As training goes on, these fluctuations get decreased with smooth regularization in the process of learning. However, the amount of difference suggests additional refinement to obtain improved stability in the model.

The validation loss is gradually reducing in an indication of better overall ability to generalize by the model. Unlike training loss, validation loss does not experience abrupt changes, which is yet another indicator that overfitting is not occurring. As an indicator where stable validation loss is an indicator in establishing capacity to perform in unseen data, this is an additional indicator in establishing usefulness in practice.

In determining the relationship between the number of hidden layers in a neural network and its predictive ability, the training process was initially started with one hidden layer. Given the dataset included 75 instances, with 10 input features and 3 output variables, the optimal number of neurons in this configuration was determined as being in the range of 8 to 12. Experiments were then conducted using two hidden layers based on the idea that using just one layer may cause the network to underfit the information, as indicated by the validation loss settling at a fairly high level. Two relatively small hidden layers were then considered in order to resolve this problem, such as configurations of the type [10, 5] and [8, 4]. Making this change produced an increase in classification accuracy. However, extending the number of hidden layers further produced overfitting that reduced the generalizability of the model. Consequently, the structure of the hidden layers was deliberately kept at the small scale in order to find an appropriate balance in terms of accuracy and overfitting.

# CONCLUSIONS AND RECOMMENDATIONS

This research proposed an innovative method for oil condition prediction in power generators using acoustic signal analysis, as an alternative to traditional Scheduled Oil Sampling (S.O.S) testing. The approach focused on the classification of oil health into three categories—"Action Required," "Monitor," and "No Action Required"—using features extracted from generator acoustic emissions.

Two experimental trials were conducted to evaluate the performance of a machine learning-based classification system. The second trial, using a larger and more balanced dataset, demonstrated improved accuracy (from 0.67 to 0.74) and overall better generalization capabilities. The model effectively reduced misclassifications, especially in ambiguous categories such as "Monitor" and "No Action Required". Training and validation curves confirmed that while the model experienced some fluctuations during early training phases, validation loss showed a steady decline, suggesting a reduced risk of overfitting and reliable performance on unseen data.

The principal aim of this investigation was to establish a non-invasive technique for predicting the condition of generator oil through the utilization of acoustic signals. This aim was effectively realized, as a machine learning-oriented framework was constructed and executed to establish correlations between acoustic signatures and the states of oil degradation. The system that was developed categorized oil condition into three standard classifications as defined by Caterpillar S.O.S. testing: Action Required, Monitor, and No Action Required. Experimental validation substantiated that acoustic signals serve as reliable indicators of oil health, thereby illustrating the practicality of this method for predictive maintenance.

Specific Objectives:

1. Identification of Relevant Acoustic Signals of Deterioration in Oil
    - The achievement was established through systematic data measurements performed under no-load generator conditions using a Blue Yeti microphone.
    - Distinct frequency bands were analyzed, and the relevant high-frequency band up to 10 kHz was selected for oil condition prediction.
    - The acoustic characteristics associated with oil erosion were distinctly differentiated and identified from backgrounds.
2. Time and Frequency Domain Feature Extraction
    - Processed using the Librosa library in Python.
    - Time-domain features such as the Amplitude Envelope, Root Mean Square Energy (RMSE), Entropy, and Zero Crossing Rate (ZCR) were extracted effectively.
    - The frequency-domain features, such as Spectral Centroid, Spectral Bandwidth, Band Energy Ratio, and Spectral Roll-off, were accurately calculated.
    - Such traits comprised the building block of the predictive model.

3. Correlation of Acoustic Emission Signatures with Feature Selection
   - A clear correlation was established between oil condition and acoustic signal characteristics.
   - Techniques for feature selection were employed to determine the most crucial indicators, thereby enhancing model accuracy and decreasing computational complexity.
4. Development of a Predictive Model of an Oil's State using Machine Learning
   - A Multi-Layer Perceptron (MLP) NN was trained with 75 sample data.
   - The model obtained encouraging classification accuracy in the separation of oil conditions.
   - The next version of the model was much better, hence reflecting its potential use in real applications.

In conclusion, all the objectives were attained. Though further verification with a more comprehensive dataset and more various generator sets is recommended, the results provide a solid foundation for application in the industry.

To enhance the effectiveness and real-world applicability of the proposed acoustic-based oil condition prediction system, several improvements are recommended. From a technical standpoint, fine-tuning learning rates can help stabilize model training, while employing larger batch sizes may contribute to smoother gradient updates and improved convergence. The incorporation of regularization techniques, such as L2 regularization or dropout layers, is advised to minimize the risk of overfitting. Additionally, implementing early stopping mechanisms—by monitoring validation loss—can prevent unnecessary training and save computational resources.

In terms of model development, exploring more sophisticated deep learning architectures like CNNs or RNNs could enhance the classification accuracy for complex acoustic patterns. Ensemble methods should also be considered, as they combine the strengths of multiple models to improve robustness and reduce classification errors.

Regarding data improvements, expanding the dataset is crucial for enabling the model to better capture subtle variations in generator acoustics across oil condition categories. Moreover, enhanced feature engineering, incorporating domain-specific signal processing techniques, may further improve the model's predictive power.

For industrial deployment, integrating the system into an edge device or Internet of Things (IoT) platform would facilitate real-time oil condition prediction and monitoring. To ensure reliability across operational environments, it is essential to validate the model using data from various generator models and conditions.

Finally, future research should investigate alternative normalization techniques to address variability stemming from sensor placement or ambient conditions. Additionally, the use of transfer learning leveraging pre-trained models from similar acoustic tasks could significantly boost performance, especially when dealing with limited or noisy datasets.

# REFERENCES


1. R. L. Klug, "Scheduled oil sampling as a maintenance tool," *SAE Transactions*, vol. 81, pp. 1311–1316, 1972. [Online]. Available: http://www.jstor.org/stable/44717360

2. A. Toms and L. Toms, "Oil analysis and condition monitoring," in *Chemistry and Technology of Lubricants*, R. Mortier, M. Fox, and S. Orszulik, Eds. Dordrecht, Netherlands: Springer, 2010. [Online]. Available: https://doi.org/10.1023/b105569_16

3. S. Rizal and K. Sekarsari, "Condition monitoring scheduled oil sample on crane machine using the fuzzy logic method," *Formosa Journal of Sustainable Research*, vol. 2, no. 7, pp. 1627–1636, 2023. [Online]. Available: https://doi.org/10.55927/fjsr.v2i7.5289

4. V. M. Martínez, B. T. Martínez, P. O. González, L. M. Moreno, and E. A. Lucia, "Results and benefits of an oil analysis programme for railway locomotive diesel engines," *Insight - Non-Destructive Testing and Condition Monitoring*, vol. 45, no. 6, pp. 402–406, 2003. doi:10.1784/insi.45.6.402.52898

5. G. Jombo and Y. Zhang, "Acoustic-based machine condition monitoring—Methods and challenges," *Eng*, vol. 4, pp. 47–79, 2023. [Online]. Available: https://doi.org/10.3390/eng4010004

6. D. W. E. Schobben and P. C. W. Sommen, "A frequency domain blind signal separation method based on decorrelation," *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 1855–1865, Aug. 2002. doi:10.1109/TSP.2002.800417

7. J. Z. Sikorska and D. Mba, "AE condition monitoring: Challenges and opportunities," in *Engineering Asset Management*, J. Mathew, J. Kennedy, L. Ma, A. Tan, and D. Anderson, Eds. London, U.K.: Springer, 2006. [Online]. Available: https://doi.org/10.1007/978-1-84628-814-2_14

8. G. Sharma, K. Umapathy, and S. Krishnan, "Trends in audio signal feature extraction methods," *Applied Acoustics*, vol. 158, p. 107020, 2020. [Online]. Available: https://doi.org/10.1016/j.apacoust.2019.107020

9. M. Caetano and X. Rodet, "Improved estimation of the amplitude envelope of time-domain signals using true envelope cepstral smoothing," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2011. doi:10.1109/ICASSP.2011.5947290

10. T. Barszcz and M. Zabaryłło, "Fault detection method based on an automated operating envelope during transient states for the large turbomachinery," *Journal of Vibroengineering*, vol. 24, no. 1, pp. 75–90, Dec. 2021. [Online]. Available: https://doi.org/10.21595/jve.2021.22165

11. O. Surucu, S. A. Gadsden, and J. Yawney, "Condition monitoring using machine learning: A review of theory, applications, and recent advances," *Expert Systems with Applications*, vol. 221, p. 119738, 2023. [Online]. Available: https://doi.org/10.1016/j.eswa.2023.119738

12. C. Paleologu, J. Benesty, and S. Ciochină, "Study of the general Kalman filter for echo cancellation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1539–1549, Aug. 2013. doi: 10.1109/TASL.2013.2245654.

13. A. O. M. Salih, "Audio noise reduction using low pass filters," *Open Access Library Journal*, vol. 4, p. e3709, 2017. [Online]. Available: https://doi.org/10.4236/oalib.1103709

14. R. R. Porle, N. S. Ruslan, N. M. Ghani, N. A. Arif, S. R. Ismail, N. Parimon, and M. Mamat, "A survey of filter design for audio noise reduction," *Journal of Advanced Review on Scientific Research*, vol. 12, no. 1, pp. 26–44, 2015.

15. R. Serizel, V. Bisot, S. Essid, and G. Richard, "Acoustic features for environmental sound analysis," in *Computational Analysis of Sound Scenes and Events*, T. Virtanen, M. D. Plumbley, and D. Ellis, Eds. Cham, Switzerland: Springer International Publishing, 2017, pp. 71–101.

16. L. Liu, Z. Zhi, H. Zhang, Q. Guo, Y. Peng, and D. Liu, "Related entropy theories application in condition monitoring of rotating machineries," *Entropy*, vol. 21, no. 11, p. 1061, 2019. [Online]. Available: https://doi.org/10.3390/e21111061

17. P. Knees and M. Schedl, "Basic methods of audio signal processing," in *Music Similarity and Retrieval*, The Information Retrieval Series, vol. 36, Berlin, Heidelberg: Springer, 2016. [Online]. Available: https://doi.org/10.1007/978-3-662-49722-7_2

18. T. Backstrom, "Comparison of windowing in speech and audio coding," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013. doi:10.1109/waspaa.2013.6701853

19. Y. H. Ali, R. Abd Rahman, and R. I. Hamzah, "Artificial neural network model for monitoring oil film regime in spur gear based on acoustic emission data,"

*Shock and Vibration*, vol. 2015, Article ID 106945, 12 pages, 2015. [Online]. Available: https://doi.org/10.1155/2015/106945

20. E. Brusa, C. Delprete, and L. G. Di Maggio, "Deep transfer learning for machine diagnosis: From sound and music recognition to bearing fault detection," *Applied Sciences*, vol. 11, no. 24, p. 11663, 2021. [Online]. Available: https://doi.org/10.3390/app112411663

21. W. Wu, T. R. Lin, and A. C. C. Tan, "Normalization and source separation of acoustic emission signals for condition monitoring and fault detection of multi-cylinder diesel engines," *Mechanical Systems and Signal Processing*, vol. 64–65, pp. 479–497, 2015. [Online]. Available: https://doi.org/10.1016/j.ymssp.2015.03.016

22. H.-K. Jo, S.-H. Kim, and C.-L. Kim, "Proposal of a new method for learning of diesel generator sounds and detecting abnormal sounds using an unsupervised deep learning algorithm," *Nuclear Engineering and Technology*, vol. 55, no. 2, pp. 506–515, 2023. [Online]. Available: https://doi.org/10.1016/j.net.2022.10.019

23. G.-J. Feng, J. Gu, D. Zhen, M. Aliwan, F.-S. Gu, and A. D. Ball, "Implementation of envelope analysis on a wireless condition monitoring system for bearing fault diagnosis," *International Journal of Automation and Computing*, vol. 12, no. 1, pp. 14–24, 2015. doi:10.1007/s11633-014-0862-x

24. R. Catanghal, T. Palaoag, and C. Dayagdag, "Environmental acoustic transformation and feature extraction for machine hearing," *IOP Conference Series: Materials Science and Engineering*, vol. 482, p. 012007, 2019. doi:10.1088/1757-899x/482/1/012007

# APPENDIX - A

**Python Code using Librosa Library**

```
import numpy as np
import matplotlib.pyplot as plt
import librosa
import librosa.display
from librosa.display import waveshow
import IPython.display as ipd
import math

# Importing Sample File
Sample_file = "WaveAudio/29.wav"
ipd.Audio(Sample_file)

# sr - means Sample Rate
Sample, sr = librosa.load(Sample_file)

# duration in seconds of 1 sample
sample_duration = 1 / sr
print(f"One sample lasts for {sample_duration:6f} seconds")

# total number of samples in audio file
tot_samples = len(Sample)
tot_samples

# duration of Sample1 audio in seconds
duration = 1 / sr * tot_samples
print(f"The audio lasts for {duration} seconds")

#Visualization of Audio Signal (Plotting Sample - Time vs Amplitude)

#Compute the Fast Fourier Transformation (FFT)
n = len(Sample)
fft_result = np.fft.fft(Sample)
f = np.fft.fftfreq(len(fft_result), d=1/sr)

# Plot only the positive frequencies
plt.plot(f[:len(f)//2], np.abs(fft_result[:len(fft_result)//2]))
plt.title('FFT of the Signal')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude')
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
# -------------- Noise Cancellation (Using LPF) --------------------------------

# Calculate power spectrum for analysis
PSD = np.abs(fft_result) ** 2 / n

# Identify noise components (adjust threshold as needed)
threshold = 1.5  # Example threshold for noise identification Initial 0.05
noise_indices = PSD <= threshold

# Apply filter in the frequency domain
fft_filtered = fft_result.copy()
fft_filtered[noise_indices] = 0  # Zero out noise components

# Reconstruct filtered signal
filtered_signal = np.fft.ifft(fft_filtered)

# --------------------------- Plot Results ---------------------------------------

# Visualization of Original and Filtered Signal in Frequency Domain

plt.figure(figsize=(10, 6))

# Original signal's FFT (Plotting FFT of Original Signal)

# Filtered signal's FFT (Plotting FFT of Filtered Signal)

plt.tight_layout()
plt.show()

# Assuming you have the sampling rate (sr) of the original signal
time_axis = np.arange(len(filtered_signal)) / sr  # Time in seconds

plt.plot(time_axis, filtered_signal)  # Plot time-domain signal
plt.xlabel('Time (sec)')
plt.ylabel('Amplitude')
plt.title('Time-Domain Signal after Filtering')
plt.tight_layout()
plt.show()

#-----------Visualization Original and Filtered Signal in Time Domain----------

#Both Time Domain Plots
librosa.display.waveshow(Sample, sr=sr, label='Noisy')
plt.title('Sample 7')
plt.xlabel('Time (sec)')
plt.ylabel('Amplitude')
```

```
# Plot time-domain signal (Plotting before & after filtering)

plt.legend()
plt.show

#------------------------------Time Domain Features--------------------------------

# Amplitude Envelope
FRAME_SIZE = 1024
HOP_LENGTH = 512

# number of frames in amplitude envelope
ae_Sample = amplitude_envelope(filtered_signal, FRAME_SIZE,
HOP_LENGTH)
#print(ae_Sample)
value1 = np.mean(ae_Sample)
value1 = np.abs(value1)
print(f"Amplitude Envelope value is {value1}")
#len(ae_Sample)

# Visualizing Amplitude Envelope
frames1 = range(len(ae_Sample))
t1 = librosa.frames_to_time(frames1, hop_length=HOP_LENGTH)

# Plot time-domain signal
ax = plt.subplot(1, 1, 1)
plt.plot(time_axis, filtered_signal)
plt.plot(t1, ae_Sample, color="r")
#plt.ylim((-1, 1))

plt.xlabel('Time (sec)')
plt.ylabel('Amplitude')
# Rename Sample Title Accordingly
plt.title("Amplitude Envelope Sample 7")
plt.tight_layout()
plt.show()

# Root Mean Square Energy (RMSE)
FRAME_SIZE = 1024
HOP_LENGTH = 512

rms_Sample = librosa.feature.rms(y=filtered_signal,
frame_length=FRAME_SIZE, hop_length=HOP_LENGTH)[0]
len(rms_Sample)

frame2 = range(len(rms_Sample))
t2 = librosa.frames_to_time(frame2, hop_length=HOP_LENGTH)
```

```
#librosa.display.waveshow(time_axis, filtered_signal)
plt.plot(time_axis, filtered_signal)
plt.plot(t2, rms_Sample, color="r")
#plt.ylim((-1, 1))
# Rename Sample Title Accordingly
plt.title("Root Mean Squared Energy Sample 7")

plt.xlabel('Time (sec)')
plt.ylabel('Amplitude')
plt.tight_layout()
plt.show()

value2 = np.mean(rms_Sample)
value2 = np.abs(value2)
print(f"Root Mean Squared Energy value is {value2}")

# Zero Crossing Rate (ZCR)
FRAME_SIZE = 1024
HOP_LENGTH = 512

# Ensure floating-point type
filtered_signal_ = filtered_signal.astype(np.float64)

zcr_Sample = librosa.feature.zero_crossing_rate(filtered_signal_,
frame_length=FRAME_SIZE, hop_length=HOP_LENGTH)[0]

zcr_Sample.size

#Visualization of ZCR
frame3 = range(len(zcr_Sample))
t3 = librosa.frames_to_time(frame3, hop_length=HOP_LENGTH)

plt.plot(t3, zcr_Sample, color="y")
 # Rename Sample Title Accordingly
plt.title("Zero Crossing Rate Sample 7")

plt.xlabel('Time (sec)')
plt.ylabel('Amplitude')
plt.tight_layout()
plt.show()

value3 = np.mean(zcr_Sample)
value3 = np.abs(value3)
print(f"Zero Crossing Rate value is {value3}")

#Signal Entropy
```

```
y= filtered_signal
# Calculate probability distribution
# Adjust bin count as needed
p = np.histogram(y, bins=np.linspace(min(y), max(y), num=256))[0] / len(y)

# Calculate entropy (assuming base 2)
 # Add a small value to avoid log(0)
value4 = -np.sum(p * np.log2(p + np.finfo(float).eps))

# Print entropy
print(f"Signal Entropy value is {value4}")

#Impact Index
y = filtered_signal

# Calculate signal energy
energy = np.sum(y**2)

# Normalize energy (optional)
# You can normalize the energy by the signal length or maximum value

# Define impact index based on energy
value5 = impact_index = energy  # Or your custom formula based on energy
value5 = abs(value5)

# Print impact index
#print("Acoustic Impact Index (Energy based):", impact_index)
print(f"Impact Index (Energy based) value is {value5}")

#------------------------Frequency Domain Features--------------------------------

#Band Energy Ratio
# Extract Spectrograms
FRAME_SIZE = 2048
HOP_SIZE = 512

# Calculate Band Energy Ratio

def band_energy_ratio(spectrogram, split_frequency, sample_rate):
   """Calculate band energy ratio with a given split frequency."""

   split_frequency_bin = calculate_split_frequency_bin(split_frequency,
sample_rate, len(spectrogram[0]))
   band_energy_ratio = []

   # calculate power spectrogram
```

```
Sample_spec = librosa.stft(filtered_signal_, n_fft=FRAME_SIZE,
hop_length=HOP_SIZE)
ber_Sample = band_energy_ratio(Sample_spec, 250, sr)

# Visualization of Band Energy Ratio (BER)
frames6 = range(len(ber_Sample))
t6 = librosa.frames_to_time(frames6, hop_length=HOP_SIZE)

plt.plot(t6, ber_Sample, color="b")
 # Rename Sample Title Accordingly
plt.title("Band Energy Ratio Sample 7")

plt.xlabel('Time (sec)')
plt.ylabel('Amplitude')
plt.tight_layout()
plt.show()

value6 = np.mean(ber_Sample)
value6 = np.abs(value6)
print(f"band Energy Ratio value is {value6}")

# Spectral Centroid (SC)
FRAME_SIZE = 1024
HOP_LENGTH = 512

sc_Sample = librosa.feature.spectral_centroid(y=filtered_signal_, sr=sr,
n_fft=FRAME_SIZE, hop_length=HOP_LENGTH)[0]

# Visualization of Spectral Centroid (SC)
frames7 = range(len(sc_Sample))
t7 = librosa.frames_to_time(frames7, hop_length=HOP_LENGTH)

plt.plot(t7, sc_Sample, color="b")
# Rename Sample Title Accordingly
plt.title("Spectral Centroid Sample 7")

plt.xlabel('Time (sec)')
plt.ylabel('Amplitude')
plt.tight_layout()
plt.show()

value7 = np.mean(sc_Sample)
value7 = np.abs(value7)
print(f"Spectral Centroid value is {value7}")

#Bandwidth
```

```
ban_Sample = librosa.feature.spectral_bandwidth(y=filtered_signal_, sr=sr,
n_fft=FRAME_SIZE, hop_length=HOP_LENGTH)[0]

#Visualization of bandwidth
frames8 = range(len(ban_Sample))
t8 = librosa.frames_to_time(frames8, hop_length=HOP_LENGTH)

plt.plot(t8, ban_Sample, color="b")
 # Rename Sample Title Accordingly
plt.title("Bandwidth Sample 7")

plt.xlabel('Time (sec)')
plt.ylabel('Amplitude')
plt.tight_layout()
plt.show()

value8 = np.mean(ban_Sample)
value8 = np.abs(value8)
print(f"Bandwidth value is {value8}")

# Spectral Roll-off Frequency

# Compute spectral roll-off
rolloff_freq = librosa.feature.spectral_rolloff(y = filtered_signal_  + 0.01,
sr=sr)[0]  # Add a small value to avoid division by zero

# Print the spectral roll-off frequency
#print("Spectral Roll-off Frequency:", rolloff_freq, "Hz")

# Calculate mean spectral roll-off across frequency bins (optional)
value9 = mean_rolloff = np.mean(rolloff_freq)
#print(mean_rolloff)
print(f"Spectral Roll-off Frequency value is {value9}")

# Optional: Plot the signal and roll-off
plt.figure(figsize=(12, 4))

# Plot waveform
plt.subplot(2, 1, 1)
plt.plot(y, label="Audio Signal")
plt.xlabel("Time")
plt.ylabel("Amplitude")

# Plot spectral roll-off (requires librosa version >= 0.8)
if librosa.__version__ >= "0.8":
   plt.subplot(2, 1, 2)
```

```
    librosa.display.specshow(librosa.stft(y), sr=sr, x_axis="time",
y_axis="log")
    plt.plot(
        librosa.frames_to_time(np.argmax(rolloff_freq, axis=0)), rolloff_freq,
color="r", label="Spectral Roll-off"
    )
    plt.xlabel("Time")
    plt.ylabel("Frequency (Hz)")
    plt.legend()

plt.tight_layout()
plt.show()

# Spectral Flux

# Compute spectral roll-off

 # Add a small value to avoid division by zero
rolloff_freq = librosa.feature.spectral_rolloff(y = filtered_signal_  + 0.01,
sr=sr)[0]

# Print the spectral roll-off frequency
#print("Spectral Roll-off Frequency:", rolloff_freq, "Hz")

# Calculate mean spectral roll-off across frequency bins (optional)
value9 = mean_rolloff = np.mean(rolloff_freq)
#print(mean_rolloff)
print(f"Spectral Roll-off Frequency value is {value9}")

# Optional: Plot the signal and roll-off
plt.figure(figsize=(12, 4))

# Plot waveform
plt.subplot(2, 1, 1)
plt.plot(y, label="Audio Signal")
plt.xlabel("Time")
plt.ylabel("Amplitude")

# Plot spectral roll-off (requires librosa version >= 0.8)
if librosa.__version__ >= "0.8":
    plt.subplot(2, 1, 2)
    librosa.display.specshow(librosa.stft(y), sr=sr, x_axis="time",
y_axis="log")
    plt.plot(
        librosa.frames_to_time(np.argmax(rolloff_freq, axis=0)), rolloff_freq,
color="r", label="Spectral Roll-off"
    )
```

```
    plt.xlabel("Time")
    plt.ylabel("Frequency (Hz)")
    plt.legend()

plt.tight_layout()
plt.show()

# Visualization of spectral flux
frames10 = range(len(flux))
t10 = librosa.frames_to_time(frames10, hop_length=HOP_LENGTH)

plt.plot(t10, flux, color="b")
# Rename Sample Title Accordingly
plt.title("Spectral Flux Sample 7")

plt.xlabel('Time (sec)')
plt.ylabel('Spectral Flux')
plt.tight_layout()
plt.show()

#----------------------------------------ML-------------------------------------------

import tensorflow as tf
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, confusion_matrix

#Importing sample data
data =
pd.read_csv('Training_Data_Set_2_Samples60_Words_updated_75.csv')

from sklearn import preprocessing

Label = preprocessing.LabelEncoder()
Output = Label.fit_transform(data['Output'])
Output

#Defining Training and Testing data set
train, test = train_test_split(data, random_state=42)
X_train = train[train.columns[:10]]
Y_train = train['Output']
X_test = test[test.columns[:10]]
Y_test = test['Output']
```

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)

X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

from sklearn.neural_network import MLPClassifier

MLP = MLPClassifier(activation = 'relu', solver='lbfgs',
hidden_layer_sizes=(10, 5), max_iter=250)
MLP.fit(X_train, Y_train.values.ravel())

predictions = MLP.predict(X_test)
predictions

# Classification Matircs

from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(Y_test, predictions))
print(classification_report(Y_test, predictions))

# Confusion Matrics

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
import matplotlib.pyplot as plt
# Calculate performance metrics
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score

y_test = Y_test
y_pred = predictions

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

# Print performance metrics
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1-score:", f1)
```