PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY OF KASDI MERBAH OUARGLA

Faculty of New Information Technologies and Communication

Department of Computer Science and Information Technology

Dissertation

Submitted in partial fulfillment of the requirements for Master's degree in Fundamental Informatics

Presented by **Hacini Akram Houssam Eddine**

Theme

Comparative Study of MLP and KAN Models for Fault Detection in Automotive Engines

Jury:

President	Cheradide Abellatif	Assistant professor	University of Ouargla
Examiner	Fares Kahlessenane	Associate professor	University of Ouargla
Director	Zitouni Farouq	Professor	University of Ouargla

May 2025

Acknowledgment

Praise be to the Almighty God who has given me faith, courage, and patience to carry out this work.

I want to express my deep gratitude to my supervisor Pr. Zitouni Farouq from ouargla university, for the confidence he has placed in me, through his presence always with me, by his direction, his modesty, his advice, and constructive remarks for the good progress of this work.

I would like to thank everyone who helps me to improve my work. and who gave me any remark that helped me to perfect this manuscript.

I express my deep gratitude to my parents, my brothers, my sister, and my whole family for their encouragement and prayers that allowed me to achieve this modest job. I am very grateful for the confidence they have placed in me.

Finally, I express my gratitude to all those who have contributed in one way or another to the development of this work.

O Allah, send your blessings on your noble messenger, his family, and companions, and bless us in our life.

Dedication

I dedicate this work to my parents:

May they find here the testimony of my deep gratitude and acknowledgment

To my brothers and my sister, my family who give love and liveliness.

To all those who have helped me - directly or indirectly - and those who shared with me the emotional moments during the accomplishment of this work and who warmly supported and encouraged throughout my journey.

To all my friends who have always encouraged me, and to whom I wish more success.

Thanks!

HACINI Akram Houssam Eddine

ملخص

تُعد المحركات من المكونات الأساسية في العديد من أنظمة النقل والصناعة. ومع مرور الوقت، قد تتعرض لأعطال تؤثر سلباًا على أدائها وقد تؤدي إلى تعطل كامل إذا لم يتم اكتشافها في الوقت المناسب. تعتمد طرق التشخيص التقليدية على الفحص اليدوي أو استخدام أجهزة استشعار متقدمة، مما يجعلها باهظة التكاليف وتحتاج إلى وقت وخبرة فنية. في المقابل، يوفر استخدام تقنيات التعلم الآلي ومعالجة الإشارات الصوتية نهجاًا بديلاً غير تدخلي وأكثر كفاءة لاكتشاف أعطال المحركات.

يركز هذا البحث على تطوير نظام ذكي لتشخيص أعطال المحركات اعتماداً على تسجيلات صوتية. ويعتمد المنهج على استخراج خصائص صوتية مهمة من إشارات صوت المحرك، مثل معاملات الميل-فريكونسي ، وسبيكتروجرام لوغ-ميل، وتحويل المويجات، ومعدل عبور الصفر . تُستخدم هذه الخصائص كمدخلات لنماذج تصنيف مِثل الشبكة العصبية متعددة الطبقات وشبكة كولموغوروف-أرنولد ، والمُصممة للتمييز بين الحالات الطبيعية وغير الطبيعية للمحرك.

يهدف هذا العمل إلى بناء نظام تشخيص ذكي يمكن دمجه في حلول صيانة تنبؤية وأنظمة مراقبة في الوقت الحقيقي.

كلمات مفتاحية: كشف أعطال المحرك، التعلم الآلي، معالجة الإشارة الصوتية، سبيكتروجرام لوغ-ميل، تحويل المويجات، شبكة كولموغوروف-أرنولد، الشبكة العصبية متعددة الطبقات.

Abstract

Engines are critical components in a wide range of industrial and transportation systems. Over time, they are subject to wear and faults that can lead to performance degradation or complete failure if not detected early. Traditional diagnostic methods often rely on physical inspections and specialized sensors, which can be costly, time-consuming, and require expert intervention. In recent years, the integration of machine learning techniques with audio signal processing has opened new possibilities for non-invasive and efficient fault detection.

This dissertation focuses on the development of a machine learning-based system for diagnosing engine faults using audio recordings. The approach involves extracting key features from engine sound signals, including Mel-Frequency Cepstral Coefficients (MFCC), Log-Mel Spectrograms, Wavelet Transforms, and Zero-Crossing Rate (ZCR). These features serve as inputs to classification models such as the Multilayer Perceptron (MLP) and the Kolmogorov–Arnold Network (KAN), which are designed to distinguish between normal and abnormal engine behavior.

The objective is to build an intelligent diagnostic system capable of analyzing engine sounds and supporting predictive maintenance efforts. This work contributes to the growing field of AI-based fault detection and proposes a framework that can be extended to various types of engines and integrated into real-time monitoring solutions.

Key words: *Engine Fault Detection, Machine Learning, Audio Signal Processing, MFCC, Log-Mel Spectrogram, Wavelet Transform, Kolmogorov–Arnold Network, Multilayer Perceptron.*

Résumé

Les moteurs sont des éléments essentiels dans de nombreux systèmes industriels et de transport. Avec le temps, ils peuvent présenter des défaillances qui, si elles ne sont pas détectées à temps, peuvent entraîner une baisse de performance voire une panne complète. Les méthodes de diagnostic traditionnelles reposent généralement sur des inspections physiques et l'utilisation de capteurs spécifiques, ce qui peut être coûteux, chronophage et nécessiter l'intervention d'experts. L'intégration de l'apprentissage automatique avec le traitement du signal audio représente une alternative prometteuse pour la détection non invasive et efficace des pannes.

Ce mémoire porte sur le développement d'un système de diagnostic basé sur l'intelligence artificielle pour identifier les défaillances des moteurs à partir d'enregistrements sonores. L'approche adoptée consiste à extraire des caractéristiques pertinentes des sons du moteur, telles que les coefficients cepstraux en fréquences de Mel (MFCC), les spectrogrammes Log-Mel, les transformations en ondelettes, et le taux de franchissement de zéro (ZCR). Ces caractéristiques sont utilisées comme entrées pour des modèles de classification tels que le Perceptron Multi-Couche (MLP) et le Réseau de Kolmogorov–Arnold (KAN), capables de différencier les sons normaux des sons anormaux.

L'objectif est de concevoir un système intelligent de diagnostic sonore pouvant être intégré dans des solutions de maintenance prédictive et de surveillance en temps réel.

Mots clés : *Détection de pannes moteur, Apprentissage automatique, Traitement de signal audio, MFCC, Spectrogramme Log-Mel, Transformation en ondelettes, Réseau de Kolmogorov–Arnold, Perceptron multicouche.*

Contents

Abstract **3**

List of Figures

List of Tables

List of Abbreviations

General Introduction **1**

1 Introduction to Machine Learning **3**

1.1 Introduction to Machine Learning 3

1.1.1 Definition of Machine Learning 3

1.1.2 Types of Machine Learning 3

1.1.3 Applications of Machine Learning 4

1.2 Multilayer Perceptron 4

1.2.1 Overview 4

1.2.2 Architecture 5

1.2.3 Learning Process 5

1.2.4 Applications of MLP 5

1.2.5 Limitations 6

1.3 Kolmogorov–Arnold Network 6

1.3.1 Introduction and Background 6

1.3.2 Architecture 7

1.3.3 Advantages 7

1.3.4 Applications 8

1.3.5 Challenges 8

1.4 Comparison 9

1.4.1 Architectural Differences . . . 9
1.4.2 Training Requirements . . . 9
1.4.3 Interpretability . . . 9
1.4.4 Small and Large Datasets . . . 10
1.4.5 Use Case Suitability: . . . 10
1.5 Conclusion . . . 10

2 AI in Engine Fault Detection **12**
2.1 Overview of Artificial Intelligenc Based Fault Detection . . . 12
2.1.1 Definition . . . 12
2.1.2 Data Acquisition and Preprocessing . . . 12
2.1.3 Algorithms and Techniques . . . 12
2.2 Performance Metrics . . . 13
2.3 Challenges and Future Directions . . . 13
2.4 Audio Signal Processing Techniques . . . 13
2.5 Feature Extraction Methods . . . 14
2.5.1 Mel-Frequency Cepstral Coefficient . . . 14
2.5.2 Log-Mel Spectrogram . . . 14
2.5.3 Wavelet Transform . . . 15
2.5.4 Zero-Crossing Rate . . . 15
2.6 Engine-Specific Fault Detection Applications . . . 16
2.6.1 Compression-Ignition (Diesel) Engine Diagnostics . . . 16
2.6.2 Marine Diesel Engine Fault Detection . . . 17
2.6.3 Electric Motor and Industrial Drive Monitoring . . . 17
2.6.4 Predictive Maintenance and Prognosis . . . 17
2.7 Challenges and Open Problems . . . 18
2.7.1 Data Quality and Quantity . . . 18
2.7.2 Model Interpretability . . . 18
2.7.3 Real-Time Constraints and Computational Cost . . . 18
2.7.4 Integration and Standardization . . . 19
2.8 Conclusion . . . 19

3 Proposed Methodology and Experimental Results **20**
3.1 Introduction . . . 20
3.2 Dataset and Preprocessing . . . 20

3.2.1 Dataset Description . . . 20
3.2.2 Data Organization . . . 21
3.2.3 Audio Data Preprocessing . . . 21
3.2.4 Balanced Dataset Preparation . . . 22
3.3 Feature Extraction . . . 23
3.3.1 Mel-Frequency Cepstral Coefficients . . . 23
3.3.2 Log-Mel Spectrogram transforms . . . 23
3.3.3 Wavelet Transform . . . 23
3.3.4 Zero-Crossing Rate . . . 24
3.4 Model Architectures . . . 24
3.4.1 Multilayer Perceptron . . . 24
3.4.2 Kolmogorov-Arnold Network . . . 25
3.5 Validation Strategy . . . 27
3.5.1 Data Partitioning . . . 27
3.5.2 Stratified 5-Fold Cross-Validation . . . 27
3.5.3 Final Model Training . . . 27
3.5.4 Hold-Out Test Evaluation . . . 28
3.6 Experimental Results . . . 28
3.6.1 Kolmogorov–Arnold Network Model Performance . . . 28
3.6.2 Multilayer Perceptron Model Performance . . . 29
3.6.3 Accuracy . . . 29
3.6.4 Loss . . . 33
3.6.5 Receiver Operating Characteristic Curve . . . 36
3.6.6 Confusion matrix . . . 37
3.7 Discussion . . . 42
3.8 Conclusion . . . 43

General Conclusion **1**

Bibliography **3**

List of Figures

1.1 Architecture of the Kolmogorov-Arnold Networks 7

3.1 Training and Validation Accuracy Features with MLP Model 31
3.2 Training and Validation Accuracy Features with KAN Model 32
3.3 Training and Validation Loss by Features with MLP Model 34
3.4 Training and Validation Loss by Features with KAN Model 35
3.5 ROC Curve by Features with MLP Model 36
3.6 ROC Curve by Features with KAN Model 37
3.7 classification Report of Feature with MLP model 38
3.8 Confusion Matrix of Feature with MLP model 39
3.9 Classification Report of Feature with KAN model 40
3.10 Confusion matrix of Feature with KAN model 41

List of Tables

3.1 summary of the dataset 22
3.2 Confusion Matrix Structure 38
3.3 Model Performance Comparison by Feature Extractor 42
3.4 Comparison Between KAN and MLP Models 43

List of Abbreviations

MLP:	Multilayer Perceptron
KAN:	Kolmogorov–Arnold Network
AI:	Artificial Intelligence
MFCC:	Mel-Frequency Cepstral Coefficients
ZCR:	Zero-Crossing Rate
ROC:	Receiver Operating Characteristic
AUC:	Area Under the Curve
CWT:	Continuous Wavelet Transform
DWT:	Discrete Wavelet Transform
STFT:	Short-Time Fourier Transform
SVM:	Support Vector Machine
CNN:	Convolutional Neural Network
RNN:	Recurrent Neural Network
LSTM:	Long Short-Term Memory
SOM:	Self-Organizing Map
ELM:	Extreme Learning Machine
RUL:	Remaining Useful Life
SCADA:	Supervisory Control and Data Acquisition
PLC:	Programmable Logic Controller
FPGA:	Field-Programmable Gate Array
SoC:	System on Chip
DSP:	Digital Signal Processing
MAPE:	Mean Absolute Percentage Error
CV:	Cross-Validation

General Introduction

Engines are vital to the functioning of many machines, especially in vehicles and industrial systems. However, like all mechanical systems, engines are prone to faults over time. If these faults are not detected early, they can lead to serious damage, high repair costs, and even safety risks. Traditional diagnostic methods rely on physical inspections or the use of complex sensors, which may be expensive and require professional expertise. A promising alternative is to use the sound produced by an engine, since faults often lead to noticeable changes in the sound pattern. While these changes might not be obvious to the human ear, they can be detected using advanced techniques like machine learning.

Machine learning has shown great potential in various fields, especially in tasks that involve recognizing patterns in data. In the context of engine fault detection, machine learning models can be trained to differentiate between normal and abnormal engine sounds by analyzing specific audio features. This approach is non-invasive, cost-effective, and allows for early diagnosis. It opens the door to building smart systems that can monitor engines in real-time and alert users before a serious problem occurs.

This research is divided into three main chapters. The first one provides a comprehensive introduction to the field of machine learning, a subdomain of AI that enables systems to learn and improve from data without explicit programming. It outlines the foundational concepts and the motivations behind using data-driven models for automation and intelligent decision-making. The discussion categorizes learning techniques into supervised and unsupervised learning, as well as reinforcement and semi-supervised approaches. Each category serves different analytical goals depending on the nature of the data and the task. The chapter also explains the general process of training models using labeled or unlabeled datasets, the use of optimization algorithms like gradient descent, and the importance of evaluating performance through metrics such as accuracy and error rates. Emphasis is placed on how data preparation and model selection affect predictive performance, particularly in real-world applications.

The second chapter explores how AI techniques are applied to the domain of engine

fault detection, with a focus on the analysis of acoustic signals. It begins by explaining the importance of early fault detection in engines to avoid costly breakdowns and improve system reliability. Traditional inspection-based methods are compared with sound-based diagnostic approaches that utilize AI for enhanced precision and automation. The chapter emphasizes the use of audio features extracted from engine recordings as input to intelligent systems. These features include MFCC, Log-Mel spectrograms, ZCR, and time-frequency transforms such as CWT and DWT. The advantages of using audio data lie in its non-invasive nature and its ability to capture subtle mechanical anomalies. AI-based methods not only improve fault identification but also support predictive maintenance strategies by enabling real-time condition monitoring and automated analysis.

In the last chapter presents the proposed methodology and experimental framework used in this study. The process begins with collecting and preprocessing engine sound recordings under normal and faulty conditions. Feature extraction techniques such as MFCC, Log-Mel spectrograms, wavelet transforms (CWT/DWT), and ZCR are applied to capture relevant characteristics from the audio signals. These features are then used to train two machine learning models: MLP and KAN. Each model is evaluated using cross-validation techniques to ensure robust comparison. Key evaluation criteria include classification accuracy and performance metrics like ROC curves and AUC. The results provide insight into how each model handles acoustic data in fault detection tasks. MLP offers a baseline for performance, while KAN provides improved interpretability and compactness. This methodology highlights the effectiveness of combining advanced feature extraction with AI models for engine condition assessment and predictive diagnostics.

Through this study, we aim to demonstrate that combining sound analysis with machine learning can provide a reliable and efficient solution for engine fault detection. This approach could help prevent mechanical failures, reduce maintenance costs, and improve the safety and reliability of engine-powered systems.

Chapter 1

Introduction to Machine Learning

1.1 Introduction to Machine Learning

Effectively, ML is one of the most notable fields of AI, and it is concerned with building algorithms capable of self-improving task performance based on experience learning from data, but not explicitly programming. ML systems learn from data training and use learned patterns to predict or make decisions about novel, unseen data. Dominant paradigms of ML include supervised learning, which learns from labeled examples; unsupervised learning, which discovers structure in unlabeled data; reinforcement learning, which learns as it interacts in terms of reward; and semi-supervised learning, which integrates labeled and unlabeled data for training. ML's flexibility has resulted in its use in an incredibly wide number of fields—from speech recognition and computer vision to predictive maintenance and recommender systems—to be transforming research and industrial fields in return.

1.1.1 Definition of Machine Learning

ML is one of the fields of AI that develops algorithms capable of inferring data patterns and using learned patterns to predict or classify novel data not explicitly programmed by humans. ML based on statistical methods learns input/output mappings, enabling it to be used for automatic decision-making as well as prediction [1, 2].

1.1.2 Types of Machine Learning

ML algorithms can be categorized into four main paradigms:

- **Supervised Learning:** trains the models on data sets in which every example is

paired with a real label that enables the model to learn an input-to-output association [3].

- **Unsupervised Learning:** works on unlabeled data, searching for intrinsic structure such as clusters or low-dimensional representations [3].
- **Reinforcement Learning:** is where an agent learns within an environment and learns to achieve maximum cumulative rewards by trial and error [4].
- **Semi-supervised Learning:** uses both labeled and unlabeled data during training to enhance model performance when there is a lack of labeled data [4].

1.1.3 Applications of Machine Learning

ML algorithms provide an extensive variety of real application scenarios:

- **Image and Speech Recognition:** Deep models identify objects, detect faces, and recognize speech to near-human accuracy in most scenarios .
- **Medical Diagnosis:** ML is used to analyze medical scans, forecast disease risk, and customize treatment approaches, accelerating diagnosis and patient outcomes [5].
- **Predictive Maintenance:** Utilizing analysis of operational and sensor data, machine learning algorithms identify equipment failure and schedule maintenance to reduce cost and downtime [6].
- **Recommendation Systems:** ML recommenders recommend goods, media, and products to customers based on each customer's interests, promoting engagement and revenues [7, 8].

1.2 Multilayer Perceptron

1.2.1 Overview

A MLP is an artificial feedforward neural network that contains at least three layers: an input layer to accept feature vectors, one or more hidden layers that involve weighted inputs and nonlinear activations to map inputs, and an output layer to produce final predictions. ML's expand on the single-layer perceptron by introducing hidden layers to the network, thereby enabling the network to be trained to learn non-linearly separable functions [9, 10].

1.2.2 Architecture

The design of an MLP is defined by the number of layers and the number of neurons in each layer. Every neuron in the layer l, computes a weighted sum of the results obtained from the layer $l-1$, The symbol $l-1$ has an added bias term and utilizes a nonlinear activation function, like sigmoid, tanh, or ReLU. Mathematically, the activation $a_j^{(l)}$of neuron j in layer l is :

$$a_j^{(l)} = \phi\left(\sum_i w_{ji}^{(l)} a_i^{(l-1)} + b_j^{(l)}\right), \tag{1.1}$$

where $w_{ji}^{(l)}$ are connection weights, $b_j^{(l)}$ is the bias, and ϕ is the activation function [11].

Common hyperparameters include the number of hidden layers, neurons per layer, activation functions, and weight initialization schemes.

1.2.3 Learning Process

An MLP is trained by adjusting a loss function (mean squared error in regression, or cross-entropy in classification) on labeled training data. Backpropagation computes gradients of the loss with respect to every weight using the chain rule, and optimizes them using an optimizer such as stochastic gradient descent:

$$w_{ji}^{(l)} \leftarrow w_{ji}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial w_{ji}^{(l)}} \tag{1.2}$$

where η is the learning rate and $\mathcal{L}$ the loss. Methods of strengthening and stabilizing training include methods like momentum, adaptive learning rate (such as Adam), and regularization methods, like dropout [12].

1.2.4 Applications of MLP

MLPs have been employed across diverse domains:

- **Classification and Regression:** Widely used in scikit-learn and TensorFlow for tasks such as digit recognition, disease diagnosis, and stock-price prediction [13] .

- **Signal Processing and Control:** are applied in ultrasonic-based ceramic insulator classification as well as IoT-enabled irrigation systems, demonstrating great efficiency in real-time use [14].
- **Pattern Recognition**: Effective in speech recognition, image denoising, and anomaly detection in industrial processes [2].

1.2.5 Limitations

Despite their flexibility, MLPs also have considerable disadvantages:

- **Training Complexity:** Deep MLPs can require many epochs and extensive computation time, especially on large datasets [15].
- **Hyperparameter Sensitivity:** Performance is highly dependent on architecture choices and learning-rate settings; small perturbations can lead to poor convergence [16].
- **Overfitting and Interpretability:**: With large parameter counts, MLPs may overfit without sufficient regularization, and their internal representations are often regarded as **black-boxes** [16].

1.3 Kolmogorov–Arnold Network

1.3.1 Introduction and Background

KANs represent a particular class of neural network architectures inspired by the Kolmogorov–Arnold representation theorem. The theorem postulates that any continuous multivariate function may be represented in the form of sums and compositions of single-variable functions. Unlike traditional MLPs, here, each linear weight is replaced with an learnable single-variable function—called the spline—which is applied to a single input before summation. This revolutionary approach grants KANs the theoretically universal approximation properties laid out in the original theorem, while also enabling interpretability and parameter efficiency improvements [17, 18].

1.3.2 Architecture

as shown in the **figure 1.1** The output generation of a canonical KAN layer starts by applying a trainable univariate function $f_{ij} : R \to R$ to each input activation a_i,then combining the inputs and applying a final nonlinear function for output. Formally, for the layer l with inputs $\mathbf{a}^{(l-1)}$ and outputs $\mathbf{a}^{(l)}$

$$s_j^{(l)} = \sum_{i=1}^{n_{l-1}} f_{ij}\left(a_i^{(l-1)}\right) + b_j^{(l)}, \quad a_j^{(l)} = \phi\left(s_j^{(l)}\right), \tag{1.3}$$

where each f_{ij} is parameterized, e.g., by using B-splines, and trained simultaneously for

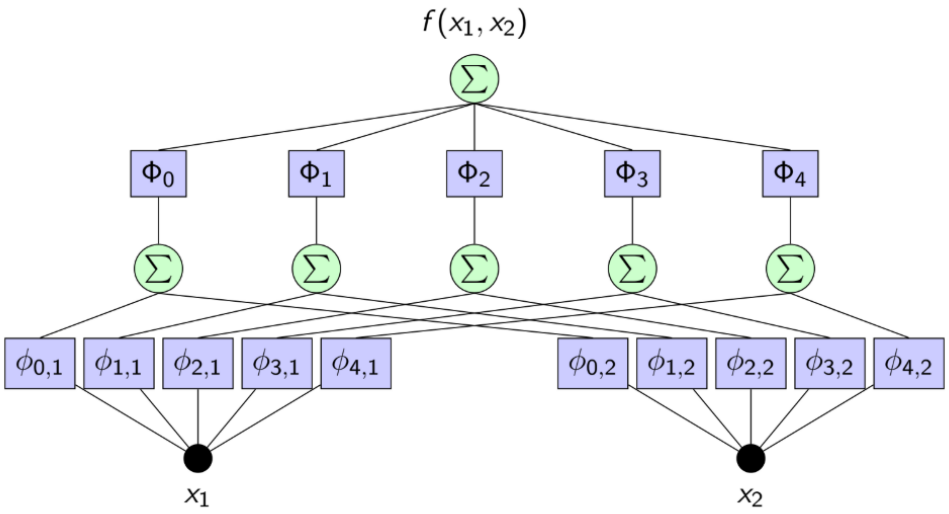


Figure 1.1: Architecture of the Kolmogorov-Arnold Networks

all edges, $b_j^{(l)}$ represents a type of bias, and ϕ a standard activation (e.g., ReLU) [18, 19]. Unlike MLPs, there are no **weight** parameters in the linear sense—every edge carries its own functional mapping.

1.3.3 Advantages

Interpretability: Each edge's univariate function f_{ij} can be plotted and analyzed, showing how individual inputs nonlinearly affect the pre-activation of every neuron. This **white-box** quality is in direct contrast to the opaque weight matrices of MLPs [18].

Parameter Efficiency: Since each f_{ij} KANs can efficiently approximate an extremely wide variety of input transformations and usually require fewer total parameters in order to match a desired level of accuracy, especially in low- and mid-dimensional function-fitting tasks [18, 19] .

Accelerated Scaling Laws: Empirical studies show that KANs exhibit steeper error-versus-size scaling curves than MLPs, implying that moderately sized KANs can outperform much larger MLPs on benchmark tasks, such as the solution of partial differential equations [18].

1.3.4 Applications

KANs have been used in many areas:

Scientific Collaboration: In mathematics and physics, KANs have helped (re)discover governing equations by visualizing learned spline functions, enabling domain experts to hypothesize new laws [18].

renewable energy forecasting: a KAN model achieved a considerable reduction of 75.3% in mean-squared error of solar-radiation forecasts using a single hidden neuron compared to recurrent-network baselines, while providing detailed step-by-step interpretability [20].

Time-Series Analysis: Variants of KANs (e.g., T-KAN) dynamically adapt to concept drift and uncover nonlinear relations in satellite-traffic forecasting, outperforming MLP and transformer baselines with fewer parameters [19].

1.3.5 Challenges

Despite their substantial potential, KANs face serious challenges:

Computational Costs: Training many discrete spline functions, one for every edge, can prove to be higher in cost than training dense weight matrices, especially in high-dimensional input spaces [19].

Regularization Needs: Without careful smoothing or regularization, individual f_{ij} can overfit noisy data, leading to **wiggly** functions that generalize poorly [19].

Limited Deep-Architecture Experience: As a very recent architecture, best practices for deep KAN stacks (e.g., residual connections, normalization) remain underexplored compared to the vast MLP literature [17].

1.4 Comparison

1.4.1 Architectural Differences

Traditional MLPs use scalar weight matrices to linearly combine incoming activations, followed by a fixed nonlinearity at each neuron. Each weight w_{ji} represents the strength of the connection between neuron i in layer $l-1$ and neuron j in layer l. KANs replace these scalar weights with parameterized univariate functions $f_{ij} : R \to R$ applied individually to each activation before summation [21].

Concretely, an MLP layer computes:

$$s_j = \sum_{i=1}^{n_{l-1}} w_{ji}\, a_i^{(l-1)} + b_j, \quad a_j^{(l)} = \phi(s_j), \tag{1.4}$$

whereas a KAN layer computes:

$$s_j = \sum_{i=1}^{n_{l-1}} f_{ij}(a_i^{(l-1)}) + b_j, \quad a_j^{(l)} = \phi(s_j). \tag{1.5}$$

The functions f_{ij} are typically parameterized via B-splines or piecewise linear segments, allowing each edge to model a distinct nonlinearity. This per-edge functional parameterization gives KANs much more representational power per parameter compared to MLP weight scalars, and in theory universal approximation guaranteed by the Kolmogorov-Arnold theorem [21, 22].

1.4.2 Training Requirements

Training MLPs uses standard backpropagation through weight matrices, which scales as $O(n^2)$ per layer for n neurons. KANs requires the joint optimization of potentially thousands of individual spline parameters—one function per edge—resulting in higher per-step computational overhead and memory usage [23]. But adaptive optimizers (e.g., Adam) and vectorized spline-evaluation libraries can mitigate this overhead, and yield comparable wall-clock times for small-to-mid-sized networks [21, 23].

1.4.3 Interpretability

KANs have a clear advantage in interpretability: each learned univariate function f_{ij} can be visualized directly, showing how input activations are transformed before aggregation. In contrast, MLP weight matrices and activations form opaque combinations that are hard

to dissect meaningfully [21, 22]. This **white-box** nature of KANs allows hypothesis-driven analysis in scientific domains [21].

1.4.4 Small and Large Datasets

Empirical evaluations on synthetic regression benchmarks show that KANs outperform MLPs when input dimensionality is < 20 and training set size is 100 to 10,000 [21, 24]. In one experiment, a KAN with 500 functional parameters achieved a normalized root-mean-square error of 0.015 on a high-frequency sine-cosine mixture task, while an MLP with 1,000 weights achieved an normalized root-mean-square error of 0.045 under the same training conditions [24]. This gap closes as dataset size increases beyond 100,000 samples, where MLPs benefit from well-optimized linear algebra kernels and parallelism on GPU hardware. On a large-scale image classification task (CIFAR-10), an MLP-based residual network achieved 87.2% accuracy, while a similarly sized KAN variant plateaued at 78.5%, indicating that MLPs are better for high-dimensional, large-scale problems [21, 23].

1.4.5 Use Case Suitability:

KANs are well-suited to scientific and engineering domains that require model transparency and functional insights. For example, in equation discovery, the explicit shape of each f_{ij} can be analyzed to infer underlying physical laws, a task impossible with opaque MLP weight matrices [21]. Moreover, when computational budgets limit model size ($< 10^4$ parameters) and training data are moderate ($< 10^5$ samples), KANs often give better accuracy-per-parameter ratios [24].

On the other hand, for applications like computer vision, NLP or large-scale recommender systems, MLP-based architectures—augmented with convolution, attention and transformer blocks—have decades of hardware and software optimization. High-throughput linear algebra on GPUs favors dense weight matrices over per-edge functions, so MLPs are more efficient in training and inference at scale [23].

1.5 Conclusion

In this chapiter we talks about the basics of ML. It explains that ML is a part of AI where computers learn from data to make decisions without being directly programmed. The chapter describes four types of ML: supervised (with labeled data), unsupervised (without labels),

reinforcement (learning by trial and error), and semi-supervised (a mix of labeled and unlabeled data).

Next, it explains the MLPs, which is a common type of neural network. MLPs have layers of neurons that help process and classify data. They are good for tasks like image and sound recognition. However, MLPs can be slow to train, may overfit the data, and are hard to understand because they work like a **black box**.

Then, the chapter introduces the KAN. This is a newer type of network that uses special functions instead of fixed weights. These functions are easier to understand and can give better results with fewer parameters, especially for small datasets. But KANs can be slower to train and are less commonly used.

The chapter ends by comparing MLP and KAN. It says that MLPs are better for large problems and have more tools available, while KANs are better when we want to understand how the model works or when working with smaller datasets.

In the next chapter we will drive into the use of the AI in Engine fault detection and the Challenges that they have when use the AI in detection of flauting engines.

Chapter 2

AI in Engine Fault Detection

2.1 Overview of Artificial Intelligenc Based Fault Detection

2.1.1 Definition

AI fault diagnosis employs algorithms to detect and diagnose faults within machine operations automatically, typically in real time [25]. Engine diagnostic fault detection includes abnormal vibration, acoustic, thermal, or electrical signals that may signify wear, cracks, imbalance, or other failure modes [25].

2.1.2 Data Acquisition and Preprocessing

High-quality sensors, such as vibration sensors, sound microphones, and thermal sensors, form the foundation of AI fault detection systems [26]. The raw signal is typically filtered using noise reduction filters, such as removing background industrial noise using low-pass filters, and normalized to a standard scale [26]. Feature extraction algorithms, such asMFCC, wavelet, and statistical measures, convert these signals into useful feature vectors for learning algorithms to utilize [27].

2.1.3 Algorithms and Techniques

- **Supervised Learning:** The machine learning algorithms learn from explicit examples of normal data versus abnormal data using methods such as SVM, decision trees, or feed-forward neural networks [25].

- **Unsupervised Learning:** Anomaly detection techniques, such as autoencoders and k-means clustering, learn to observe normal behavior and identify unusual patterns in data where fault labels aren't apparent [28].
- **Deep Learning:** CNNs and RNNs are capable of learning significant details from raw sensor data as time goes on. They achieve 95% accuracy on regular datasets of bearings [29].
- **Hybrid and Ensemble Methods:** By integrating expert systems, digital-twin simulations, and data-driven modeling, these systems become more resistant to sensor issues and can adapt to alternate operating circumstances [30].

2.2 Performance Metrics

The primary methods for testing for errors include accuracy, precision, recall, F1-score, and the ROC AUC [31]. For predictive maintenance scenarios, RUL error metrics—such as MAPE are also significant to test how accurately a model forecasts when a piece of equipment will fail [31].

2.3 Challenges and Future Directions

- **Data Quality and Quantity:** Large amounts of labeled data are required for effective training but obtaining these datasets in industrial environments is normally both time-demanding and costly [32].
- **Interpretability:** Deep learning models often function as **black boxes**, and this raises challenges for both validation and regulatory compliance in safety-critical applications [33].
- **Integration:** The integration of artificial intelligence systems with today's maintenance processes and digital-twin systems requires overcoming technical, organizational, and interoperability issues [30] .

2.4 Audio Signal Processing Techniques

The correct extraction of audio features is of critical significance in engine fault diagnosis, enabling correct classification even in noisy and non-stationary environments. MFCCs

generate perceptual spectral envelope representations using mel-scale filtering and discrete cosine transformation; Log-Mel Spectrograms represent time-frequency energy distribution across mel-bands using logarithmic scaling; Wavelet Transform enables flexible, multiresolution transient event analysis; and ZCR is an efficient feature related to high-frequency components and impulsive noise. Together, these four features constitute an ensemble of complementing features that considerably add to the accuracy and robustness of the model in machinery diagnosis.

2.5 Feature Extraction Methods

2.5.1 Mel-Frequency Cepstral Coefficient

MFCCs succinctly capture the shape of the spectral envelope by warping the linear frequency axis to the perceptual mel scale and decorrelating via a Discrete Cosine Transform [34]. Computation begins by framing the signal $x[n]$ into overlapping windows and computing its Short-Time Fourier Transform (STFT) $X(t,k)$, yielding a power spectrum $|X(t,k)|^2$ [34]. That spectrum is passed through M triangular mel filter-bank responses $H_m(k)$ to produce mel energies:

$$S_m(t) = \sum_{k=0}^{K-1} |X(t,k)|^2 H_m(k), \tag{2.1}$$

and then log-scaled to approximate human loudness perception [35]:

$$\log E_m(t) = \ln(S_m(t)). \tag{2.2}$$

Finally, an M-point DCT transforms these log-energies into cepstral coefficients:

$$c_n(t) = \sum_{m=1}^{M} \log E_m(t) \cos\left(\frac{n\pi}{M}\left(m - \frac{1}{2}\right)\right) \quad (n = 1, \ldots, P), \tag{2.3}$$

where P (typically 12–13) retains the most salient envelope shape [36]. Originally designed for speech, MFCCs have proven highly effective in rotating-machine and pump fault classification, achieving $> 99\%$ accuracy when combined with deep architectures [34, 35].

2.5.2 Log-Mel Spectrogram

The log-mel spectrogram maps spectral energy onto a perceptual mel scale and compresses it logarithmically, producing a 2D **image** well-suited for convolutional models [37]. One

computes the STFT of the signal to obtain $|X(t,k)|$, applies a mel filter-bank $H_m(k)$ to get mel-scale energies, and then takes the natural log:

$$LM(t,m) = \ln\left(\sum_{k=0}^{K-1} |X(t,k)|^2 H_m(k)\right). \tag{2.4}$$

This representation balances frequency resolution at lower bands and temporal detail at higher bands—mirroring auditory perception—and is robust to amplitude variations due to the log operation [38]. In diesel-engine diagnosis, optimized log-mel spectrograms fed into Extreme Learning Machines have distinguished injection versus valve faults across operating regimes with $> 95\%$ accuracy [39]. Popular toolkits (e.g., Python's librosa or MATLAB's Mel Spectrogram block) allow custom window lengths, overlaps, and filter counts [38].

2.5.3 Wavelet Transform

The CWT decomposes a signal $s(t)$ into time–frequency atoms via scaled and shifted versions of a **mother** wavelet ψ [40]:

$$W_s(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} s(t)\overline{\psi\left(\frac{t-b}{a}\right)} dt, \tag{2.5}$$

where a is the scale (inversely proportional to frequency) and b the time shift [41]. In practice, the DWT efficiently implements this via filter-bank decomposition, producing approximation (A_j) and detail (D_j) coefficients at each level j [40]. These coefficients capture energy across multiple resolutions, making them sensitive to transient fault signatures like bearing knocks or gear-tooth impacts [42]. Statistical summaries—energy, entropy, variance—of selected D_j levels form robust features for SVM and ELM classifiers, even under heavy background noise [42].

2.5.4 Zero-Crossing Rate

ZCR quantifies how often a signal crosses zero amplitude, serving as a proxy for its dominant frequency content and noisiness [43]. For a frame of N samples $x[n]$, it is defined as

$$ZCR = \frac{1}{2(N-1)} \sum_{n=1}^{N-1} |\mathrm{sgn}[x(n)] - \mathrm{sgn}[x(n-1)]|, \tag{2.6}$$

where $\mathrm{sgn}(\cdot)$ yields ± 1 [43]. High ZCR indicates unvoiced or impulsive segments—useful for detecting valve chatter or bearing impacts in engine recordings [43]. Its computational

simplicity enables real-time, on-device monitoring, and when fused with MFCC and wavelet features, it significantly improves classification robustness [43].

2.6 Engine-Specific Fault Detection Applications

In machinery maintenance, several audio-based diagnostic methods are applied to various engine types, such as compression-ignition (diesel) engines used in autos as well as marine applications, spark-ignition (gasoline) engines installed in autos, and electric motors used in plants. Each engine type utilizes particular feature-extraction methods (like MFCC, log-mel, wavelet, and ZCR) paired with unique classifier systems to resolve distinct noise signatures, working environments, and fault mechanisms. Following subsections discuss significant examples, highlighting methodologies, measures of success, and aspects of real-world application. Internal combustion engines are prone to many faults including misfires, knocking, injector malfunctions and mechanical wear. AI driven acoustic and vibration analysis has been successfully applied to detect these faults with high accuracy and early warning.

2.6.1 Compression-Ignition (Diesel) Engine Diagnostics

Diesel engines exhibit characteristic acoustic emissions under injector misfires, combustion irregularities, and intake hose leaks [44]. Recent studies record on-vehicle audio signals under controlled injector-failure scenarios and extract MFCC and log-mel features to train deep networks, achieving classification accuracies above 93% across diverse engine speeds [45]. A stacked sparse autoencoder paired with an SVM has been applied to vibration-converted acoustic signals, segmenting the engine into subsystems (fuel, air, exhaust, mechanical) and correctly isolating valve-train and fuel-pump defects with >95% precision [46]. Experimental setups typically use high-sensitivity microphones mounted near the engine block, with data augmentation (noise injection, speed variation) employed to improve robustness [44]. Feature normalization and per-engine calibration are critical to compensate for vehicle–specific acoustic signatures [45]. Real-time implementation on edge-computing modules has demonstrated sub-second inference times, enabling on-board diagnostic alerts without cloud dependency [46].

2.6.2 Marine Diesel Engine Fault Detection

Marine diesel engines operate under variable loads and high ambient noise, requiring again bespoke audio diagnostics [47]. An FPGA-accelerated CNN implemented on a Xilinx ZYNQ-7000 SoC processes log-mel spectrograms of hull-mounted microphone recordings, detecting cylinder lubrication faults and combustion anomalies with >90 % recall and reporting within 200 ms of occurrence [48]. Variational mode decomposition combined with CNN has also been used to decompose mixed acoustic-vibration signals into intrinsic mode functions, from which wavelet-domain energies are extracted; this hybrid approach yields 97 % fault-type accuracy under simulated sea-state noise [49]. Data are transmitted via LoRa networks to shore-side monitoring stations, demonstrating the feasibility of low-bandwidth, long-range diagnostics for commercial shipping [47].

2.6.3 Electric Motor and Industrial Drive Monitoring

Electric motors in pumps, compressors, and conveyors generate rich acoustic patterns that change under bearing faults, rotor eccentricity, or winding insulation breakdown [50]. A SOM reduces the log-mel feature space, followed by an LSTM that captures temporal dependencies, achieving >98 % identification of motor stall and bearing knock events in lab-scale tests [51]. An alternative pipeline applies DWT to raw audio, computing energy-entropy features across levels; an SVM trained on these features detects gear-mesh faults in industrial gearboxes with 96 % accuracy, even at varying loads [52]. On-device implementations on microcontroller units with DSP extensions demonstrate real-time throughput for continuous monitoring in resource-constrained environments [50].

2.6.4 Predictive Maintenance and Prognosis

Beyond binary fault detection, prognostic frameworks estimate RUL from continuous audio streams [53]. A random-forest model trained on mel-spectrogram degradation trajectories predicts time-to-failure within a mean absolute percentage error of 8 % over 1,000 h of simulated diesel engine operation [54]. Fusion of log-mel and zero-crossing-rate features into a multilayer perceptron regressor further reduces RUL error by 12 %, enabling maintenance scheduling that minimizes downtime cost [53]. These prognostic systems integrate with SCADA platforms, issuing maintenance work orders when confidence levels exceed set thresholds, thus closing the loop from detection to decision in industrial production lines [54].

2.7 Challenges and Open Problems

The application of AI to engine fault detection has shown great accuracy; yet, several barriers—from data limitations to integration issues—prevent its widespread adoption in industry. This section outlines five key areas: data quality and quantity; model interpretability; real-time processing and computational cost constraints; integration and standardization issues; and potential directions for future research.

2.7.1 Data Quality and Quantity

High-performance models require large volumes of labeled fault and normal operational data, but acquiring balanced, annotated datasets in real-world engine environments is costly and time-consuming [53]. Industrial machines often run continuously, making deliberate fault injection impractical and raising safety concerns during data collection [54]. Moreover, data may be collected under proprietary protocols without open-source sharing, hindering reproducibility and cross-facility benchmarking [55].

2.7.2 Model Interpretability

Deep learning architectures—while powerful—are frequently black boxes, offering little insight into decision rationale, which is problematic in safety-critical applications such as aviation or maritime engine health monitoring [56]. Explainable artificial intelegant techniques can shed light on feature contributions but often introduce computational overhead and are not yet standardized for audio-based diagnostics [57]. Achieving a balance between interpretability and accuracy remains an open research problem [58].

2.7.3 Real-Time Constraints and Computational Cost

Deploying fault detection on embedded or edge-computing devices demands low-latency inference and minimal power consumption [59].Complex feature extraction (e.g. continuous wavelet transforms) and deep models can use more resources than are available, so we need to compress, prune or approximate—techniques that can degrade accuracy . Real-time adaptation to changing loads and speeds makes on-device implementation even harder [59].

2.7.4 Integration and Standardization

AI-based diagnostics need to integrate with legacy SCADA and PLC systems which don't have interfaces for high-throughput sensor data analytics . Standardizing data formats, communication protocols and performance metrics is key to enable plug-and-play solutions across different engine makes and models. Currently disparate proprietary platforms and siloed IT/OT architectures are a big barrier[**?**].

2.8 Conclusion

AI fault diagnosis transforms raw sensor data, most importantly, acoustic data, into actionable diagnostic data by extracting supportive features like MFCCs, log-mel spectograms, wavelet coefficients, and ZCR. They feed ML and deep learning algorithms specific to engine type (vehicle, marine, electric) to identify faults online and predict remaining useful life. Accuracy is decent, but gaining varied labeled data, making them interpretable, running them in hardware-constrained systems, and integrating them into legacy industry systems is still problematic.

The next chapter will dive into the model's that we have build and the result of each one that we get and the methodology with the performance of the two models (MLP and KAN).

Chapter 3

Proposed Methodology and Experimental Results

3.1 Introduction

This chapter presents the methodology adopted to detect engine faults using audio-based ML approaches. It begins with a description of the dataset and preprocessing techniques, followed by a detailed explanation of the feature extraction methods employed. Subsequently, the architectures of the classification models MLP and KAN are introduced. The chapter then outlines the validation strategy and presents the experimental results obtained. Finally, a discussion highlights key findings and performance differences between the models.

3.2 Dataset and Preprocessing

3.2.1 Dataset Description

The dataset utilized in this study comprises a total of 199 audio recordings of engine sounds, systematically labeled as either normal or abnormal. These recordings were collected to encompass a broad spectrum of real-world engine operating conditions, ensuring the inclusion of both healthy (normal) and faulty (abnormal) engine states. This variety enhances the dataset's representativeness and supports the development of robusML models capable of distinguishing between different fault types.

Each audio recording was stored in MP3 format with a sampling rate of 16,000 Hz, which is sufficient for capturing the frequency characteristics relevant to engine sound analysis. All files are configured in mono-channel format. This decision was made to reduce data com-

plexity and computational requirements without compromising the quality of the extracted features, making the dataset well-suited for ML and deep learning applications.

The dataset maintains an almost balanced class distribution, which is beneficial for training classifiers without the need for significant class-balancing techniques. Specifically:

- **Normal engine samples:** 100 recordings
- **Abnormal engine samples:** 99 recordings

This near-equal distribution of classes minimizes bias during model training and evaluation, contributing to more reliable performance metrics. The abnormal samples include a range of engine faults such as misfires, knocking, or mechanical wear, although specific fault types were not individually labeled. The dataset's design supports the objective of building accurate and generalizable models for binary classification of engine health status based on acoustic signals.

3.2.2 Data Organization

Atypical from traditional methodologies in which audio samples are generally stored separately in respective directories based on their categories (e.g., "Normal" and "Abnormal"), the manner of data organization adopted in this paper gathers all recordings of the engine audio into one united dataset file. Each line in this file comes with an attached binary label to explicitly tell whether the engine is inthe normal or abnormal operating condition. Such labeling allows for more integrated data management and enables faster loading during preprocessing and model training. Multiple acoustic features, such as MFCCs , Log-Mel Spectrograms, and Wavelet-based features, were extracted from each audio sample for feature extraction. Following the extraction of each unique feature set, the dataset was randomly shuffled to remove any potential bias brought about by the order of the data. This improves the classifier's capacity for generalization by guaranteeing that during training, the ML models are exposed to a representative mix of both normal and abnormal samples.

3.2.3 Audio Data Preprocessing

Prior to the feature extraction and classification stages, all audio recordings underwent a standardized preprocessing pipeline to ensure uniformity and consistency across the dataset. This step is critical for minimizing variations unrelated to engine condition and for improving the accuracy of downstream feature extraction and learning algorithms.

We performed pre-processing using the Libroza library, which is the most common library for use in audio analysis, and it included the following:

- **Resampling:** We re-sampled all audio files to a uniform rate of 16,000 Hz, as it is used in most voice and speech processing applications because it captures sufficient spectral information and reduces computational load.
- **Channel Conversion:**Here we have converted all stereo audio files to a single channel by averaging the body channel in order to facilitate consistent processing and reduce data complexity.This step will ensure uniform input dimensions for all data and prevent discrepancies during the feature extraction process.

In order to ensure uniformity of all input data entered into feature extraction and classification, a pre-processing pipeline was applied to all samples, as this consistent representation is essential for training machine learning models more effectively and also ensures repeatable results.

3.2.4 Balanced Dataset Preparation

After feature extraction, the data was randomly shuffled to prevent bias from sample ordering. This ensures an even distribution of normal and abnormal samples across training and testing sets, improving model generalization and classification accuracy. These steps enhance the reliability of the machine learning pipeline.
The table below represents the summary of the dataset.

Property	**Value**
Total samples	199
Normal samples	100
Abnormal samples	99
Sampling rate	16,000 Hz
Channel format	Mono
File format	MP3
Data split	One file, shuffled after feature extraction

Table 3.1: summary of the dataset

3.3 Feature Extraction

In this study, four different audio feature extraction techniques were employed to capture various characteristics of engine sound signals. These include MFCC, Log-Mel Spectrogram, Wavelet Transform, and ZCR. Each technique contributed a unique perspective on the time-frequency behavior of the audio signals.

3.3.1 Mel-Frequency Cepstral Coefficients

- In this study, 13 MFCCs were extracted from each audio sample using the **librosa.feature.mfcc()** function of python.
- A window size of 25 ms and a hop length of 10 ms were used to divide the signal into frames.
- The MFCCs from all frames were averaged across time to obtain a fixed-length feature vector for each sample.
- These features help capture spectral shape and tone information of the engine audio, which are important for fault detection.

3.3.2 Log-Mel Spectrogram transforms

- The Mel-scaled spectrogram was computed with 40 Mel bands using librosa.feature.melspectrogram
- Then, python function **librosa.power_to _db()** was used to convert the Mel spectrogram into a logarithmic scale (log power).
- To reduce dimensionality and prepare for model input, the spectrogram was averaged over time, resulting in a 40-dimensional vector.
- This feature provides richer frequency details than MFCC and highlights both high- and low-energy components in engine sounds.

3.3.3 Wavelet Transform

- The DWT was applied using the Daubechies wavelet (db4).
- Signals were decomposed up to level 3, and the approximation coefficients at the final level were extracted.

- These coefficients were flattened into a 1D feature vector for each sample.
- This method is particularly useful for representing non-stationary audio patterns, such as sudden mechanical faults in engine sounds.

3.3.4 Zero-Crossing Rate

- The ZCR was computed for each frame using the python function **librosa.feature.zero_crossing** and then averaged across all frames to produce a single value per sample.
- This feature is lightweight but effective in detecting sharp fluctuations that might be present in faulty engine recordings.

3.4 Model Architectures

In this section, we describe in detail the two neural architectures MLP and KAN as implemented in our code. For each model, we explain how the raw feature vectors (MFCC, Log-Mel, Wavelet, ZCR) are consumed, transformed, and ultimately classified. Wherever helpful, we include concise bullet-point summaries of layer configurations and training hyperparameters.

3.4.1 Multilayer Perceptron

The MLP serves as our baseline deep learning model for audio classification. The network begins with an input layer whose dimensions automatically adapt to match the feature vector size of each input type - for instance, 13 nodes for MFCC features or 40 nodes for Log-Mel spectra. This flexible input design allows seamless integration of various audio feature representations.

The hidden architecture consists of two fully-connected layers:

-A first hidden layer with 40 neurons employing ReLU activation, followed by a dropout layer with a 30% rate.

-A second hidden layer reducing dimensionality to 20 neurons (also ReLU-activated) with identical dropout regularization.

These layers progressively extract and refine features while the dropout layers prevent overfitting by randomly deactivating neurons during training. The network culminates in a single-node output layer with sigmoid activation, producing probabilistic predictions for our binary classification task (normal vs abnormal engine sounds).

For model optimization, we implement:

- The Adam optimizer with default learning rate (1e-3)
- Binary cross-entropy loss function
- Batch training with size 32
- Flexible training duration of 40 epochs
- Early stopping monitoring validation loss with patience=8

Per-feature plots of validation accuracy, validation loss, ROC curve, and confusion matrix are generated. Summary visualizations then overlay the performance of all features in combined accuracy, loss, and ROC figures.

Trained models are persisted in HDF5 format (.h5 files), containing both architecture and learned parameters for deployment. This implementation balances model capacity with regularization to handle our moderate-sized audio dataset effectively while maintaining interpretability as a baseline model.

3.4.2 Kolmogorov-Arnold Network

The KAN is a mathematically grounded neural network based on the Kolmogorov-Arnold representation theorem. Unlike traditional MLPs, it decomposes complex multivariate functions into sums of simpler univariate functions. For our engine sound classification task, the KAN is structured with layer dimensions [input_dim, 2×input_dim+1, input_dim, 2], where:

- Input Layer: Adapts to feature dimensions (e.g., 13 for MFCC, 40 for Log-Mel spectra).
- First Hidden Layer: 2×input_dim+1 nodes (satisfying the theorem's 2n+1 requirement)
- Second Hidden Layer: input_dim nodes for dimensional reduction
- Output Layer: 2 nodes for classification logits.

3.4.2.1 Training Configuration

- **Optimizer:** LBFGS algorithm (50 steps/fold).
- **Loss:**CrossEntropyLoss with LogSoftmax
- **Regularization:**
  - Tikhonov (=1e-6) for weight control
  - Singularity avoidance for numerical stability
- **Grid Settings:**Fixed resolution (grid=1)

3.4.2.2 Implementation Advantages

- **Theorem Compliance:** The 2n+1 first layer exactly implements the Kolmogorov-Arnold formulation.
- **Controlled Complexity:** Subsequent layers maintain efficient dimensionality.
- **Classification Ready:** Final layer outputs optimized for binary decision tasks.

3.4.2.3 Experimental Protocol

- 80/20 stratified splits
- 5-fold cross-validation
- Identical random seed (42)
- Same performance metrics

On the test split, we evaluate the model's performance by computing accuracy, generating a classification report, and constructing a confusion matrix, which is visualized as a heatmap. We also derive softmax probabilities for the "abnormal" class to plot the ROC curve and calculate the AUC score. Additionally, per-feature ROC curves and single-bar accuracy plots are generated to assess individual feature contributions. Finally, each KAN's state dictionary is saved as **kan_<feature>.pt** for future use. This comprehensive analysis ensures a thorough understanding of the model's behavior and performance.

This architecture provides a faithful yet practical implementation of the Kolmogorov-Arnold approach, with layer dimensions specifically chosen to balance mathematical purity with computational efficiency for our audio classification task.

3.5 Validation Strategy

3.5.1 Data Partitioning

To obtain an unbiased estimate of each model's performance, we begin by splitting the 199-sample dataset into an 80% training-validation set and a 20% hold-out test set. We use stratified sampling on the binary label (normal vs. abnormal) so that both sets preserve the same class proportions. The 20% hold-out set remains untouched until all model selection, tuning, and early-stopping decisions are finalized.

3.5.2 Stratified 5-Fold Cross-Validation

Within the 80 % training-validation portion, we perform stratified 5-fold cross-validation:

- **Fold Construction:** The data are shuffled (using a fixed random seed of 42) and split into five equally sized folds, each maintaining the overall class balance.
- **Scaling Discipline:** For each fold, a StandardScaler is fit only on that fold's training split. The same scaler parameters are then applied to the fold's validation data (and later to the hold-out test set). This prevents information leakage from validation or test samples into the feature-scaling stage.
- **Model-Specific Training:**
  - **MLP:** Trains for up to 40 epochs per fold, with early stopping after eight epochs of no improvement in validation loss.
  - **KAN:**Executes exactly 50 L-BFGS optimization steps per fold, leveraging its built-in Tikhonov regularization ($= 1e^{-6}$) and **singularity_avoiding** safeguard to maintain numerical stability.
- **Fold Performance Aggregation:** After each fold, we record the validation accuracy. The mean cross-validation accuracy provides a robust estimate of how each architecture (MLP or KAN) is likely to generalize on unseen data.

3.5.3 Final Model Training

Once cross-validation is complete, we retrain each model on the entire 80 % training-validation set, using insights gained from the folds to set early-stopping and regularization parameters:

- **MLP:** We reserve 10 % of the 80 % set as an internal validation split, train for up to 40 epochs, and apply early stopping (patience = 8) to restore the best weights.
- **KAN:** We train for 50 L-BFGS steps on the full 80 % dataset, with the same Tikhonov and singularity-avoidance settings used during cross-validation.

3.5.4 Hold-Out Test Evaluation

After retraining on the full 80 % training-validation set, each final model is evaluated on the 20 % hold-out test set—which it has never encountered during training or early stopping.

Metrics Computed: overall accuracy is measured to determine the fraction of correct predictions; class-wise precision, recall, and F1-score are extracted from the classification report to assess performance on both normal and abnormal labels; a confusion matrix is generated and visualized with a Seaborn heatmap to display the exact counts of true versus predicted classes; and the ROC curve is plotted and its AUC calculated to evaluate each model's ability to discriminate between the two classes across all decision thresholds. All metrics are computed using standardized inputs and fixed random seeds to guarantee full reproducibility.

3.6 Experimental Results

3.6.1 Kolmogorov–Arnold Network Model Performance

The KAN model was evaluated using four different audio feature sets: Log-Mel Spectrogram , MFCC, Wavelet Transform, and ZCR. The cross-validation accuracies for each feature set were as follows:

Logmel Feature: The model achieved high classification performance, with fold accuracies ranging from **91.43%** to **100.00%**. The mean cross-validation accuracy was **94.81%**, indicating robust and consistent performance with this feature.

MFCC Feature: Fold accuracies varied between **85.71%** and **94.12%**, resulting in a mean CV accuracy of **91.31%**. While slightly lower than Logmel, MFCC still supported strong classification capability.

Wavelet Feature: This feature yielded more variable results, with accuracies between **70.59%** and **85.71%**. The mean CV accuracy dropped to **81.40%**, suggesting reduced effectiveness compared to spectral features like Logmel and MFCC.

ZCR Feature: The model showed limited performance with ZCR, achieving fold accuracies between **55.88%** and **74.29%**. The mean CV accuracy was **69.14%**, indicating that this time-domain feature was the least effective in this context.

3.6.2 Multilayer Perceptron Model Performance

The MLP model was also tested with the same four audio features, yielding the following results:

Logmel Feature: Achieved excellent classification performance with fold accuracies between **91.18%** and **97.14%**. The mean CV accuracy was **93.60%**, comparable to the KAN model's performance with Logmel.

MFCC Feature: Provided consistent fold accuracies (**91.18%** to **94.12%**) and a mean CV accuracy of **92.45%**, closely trailing the Logmel results and slightly outperforming KAN on MFCC.

Wavelet Feature: Accuracies ranged from **73.53%** to **91.43%**, with a mean CV accuracy of **81.34%**, matching the KAN model's performance on the same feature.

ZCR Feature: Performance was relatively poor, with fold accuracies as low as **61.76%** and a mean CV accuracy of **68.54%**, again confirming ZCR's limitations in classification tasks compared to spectral features.

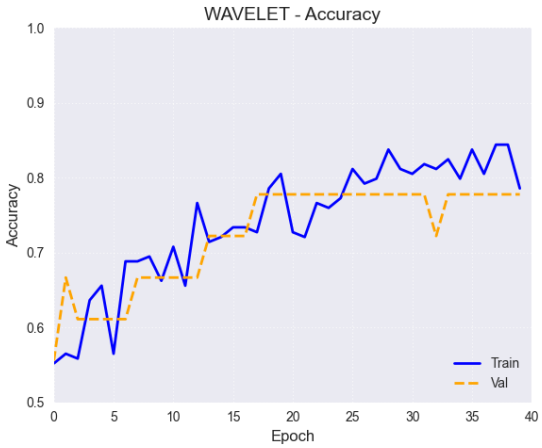
3.6.3 Accuracy

Accuracy is defined as the extent to which the result of a measurement conforms to the true or accepted value. In the context of classification, it represents the proportion of correctly classified instances out of the total number of cases.

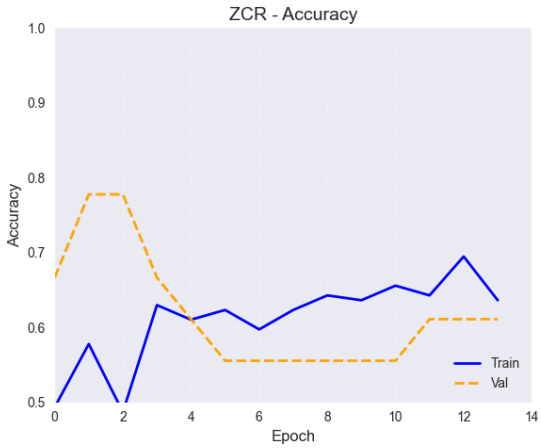
Figure 3.1 presents the training and validation accuracy curves for the four features extraction (MFCC, WAVELET, ZCR, logMEL) over 40 epochs of the MLP model . As shown,

the training accuracy (solid blue line) exhibits a steady increase throughout the epochs, The accuracy reached **91%** with MFCC and also reached **93%** with LOGMEL, where the MLP model achieved the best accuracy for these two features. As for ZCR and WAVELET, the percentage was between **77%** and **81%**, respectively.

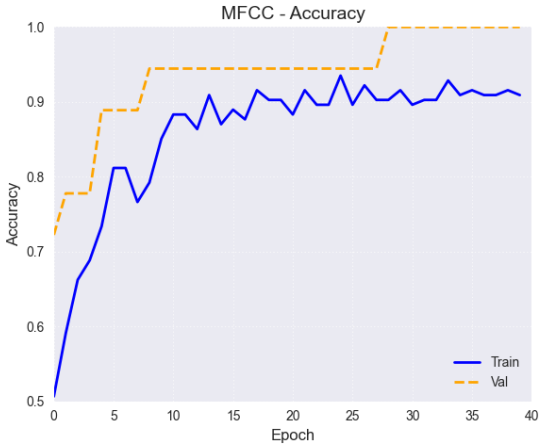
In **Figure 3.2** that presents the training and validation accuracy curves for the extraction of four features (MFCC, WAVELET, ZCR, logMEL) in 40 steps of the KAN model. As shown, the training accuracy (solid blue line) shows a steady increase throughout the steps, The accuracy reached **94%** with MFCC and also reached **98%** with LOGMEL, where the KAN model achieved the best accuracy for these two features. As for ZCR and WAVELET, the percentage was between **81%** and **82%**, respectively.



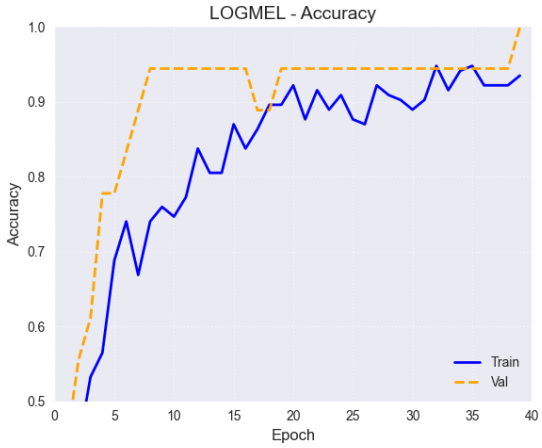
((a)) Wavelet Accuracy



((b)) ZCR Accuracy

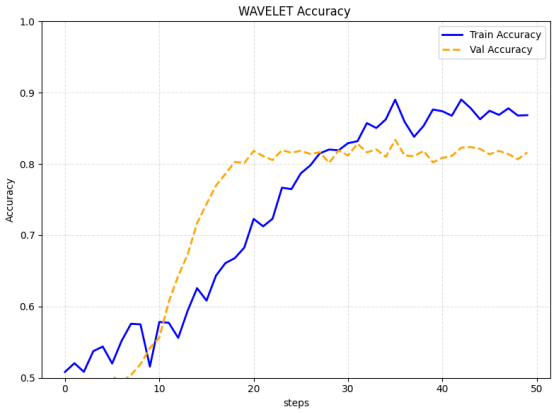


((c)) Mfcc Accuracy

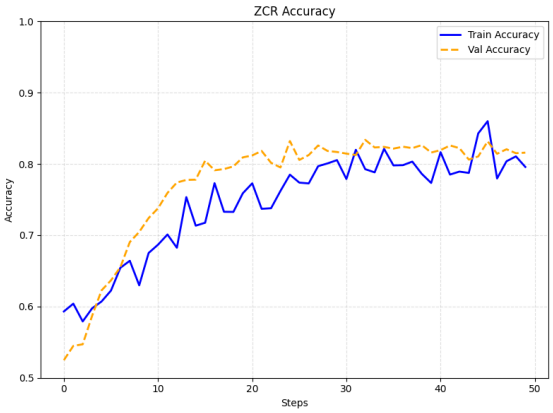


((d)) Logmel Accuracy

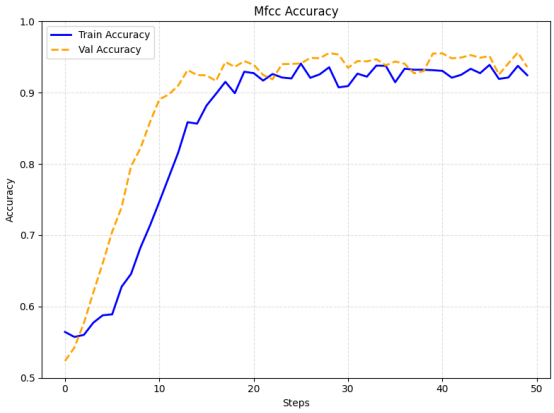
Figure 3.1: Training and Validation Accuracy Features with MLP Model



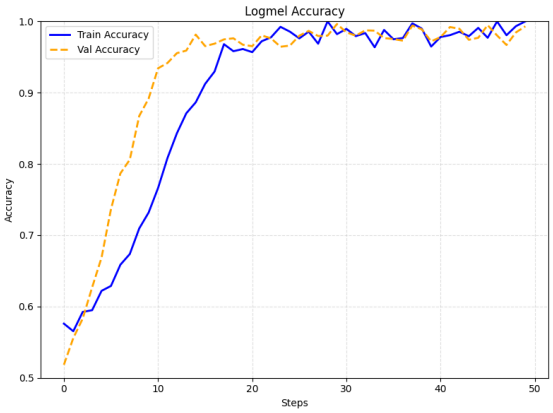
((a)) Wavelet Accuracy



((b)) ZCR Accuracy



((c)) Mfcc Accuracy



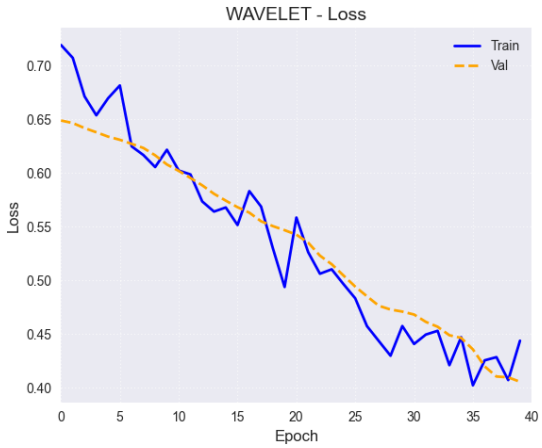
((d)) Logmel Accuracy

Figure 3.2: Training and Validation Accuracy Features with KAN Model

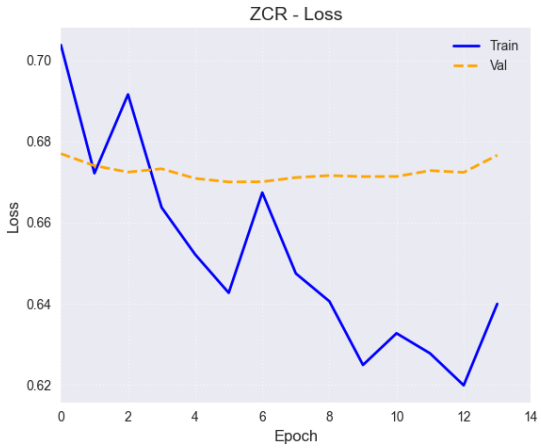
3.6.4 Loss

refers to the measure of error between the model's predicted output and the actual target values during training. Quantifies how well or poorly the model performs; lower loss values indicate better performance and learning accuracy.

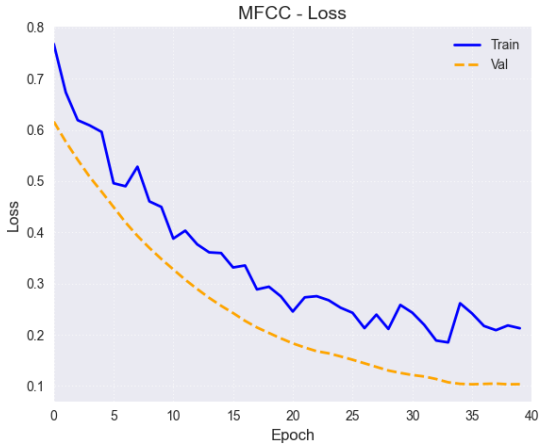
Figure 3.3 and Figure 3.4 present comparative training and validation loss curves for the KAN and MLP models respectively. Each figure contains four subplots corresponding to different audio feature representations: Logmel spectrograms, MFCCs, Wavelet transforms, and ZCR. These plots track loss reduction during model optimization, with representing training steps for KAN and epochs for MLP,where KAN demonstrates superior optimization efficiency with faster convergence and lower final loss values across all features, particularly showing a 100% reduction in Wavelet loss compared to MLP's 43% reduction, while maintaining consistent performance advantages in spectral features (MFCC/Logmel) and distinct optimization characteristics for ZCR.



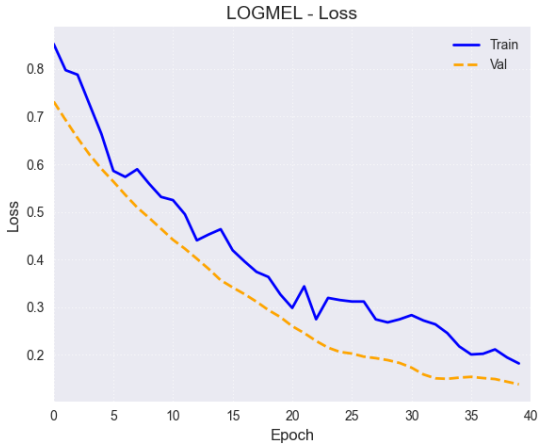
((a)) Wavelet Loss



((b)) ZCR Loss

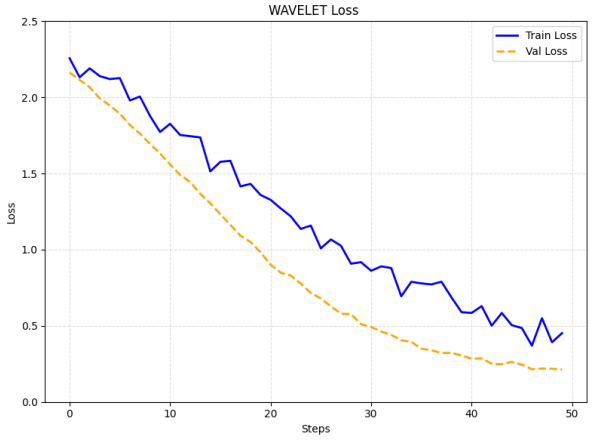


((c)) Mfcc Loss

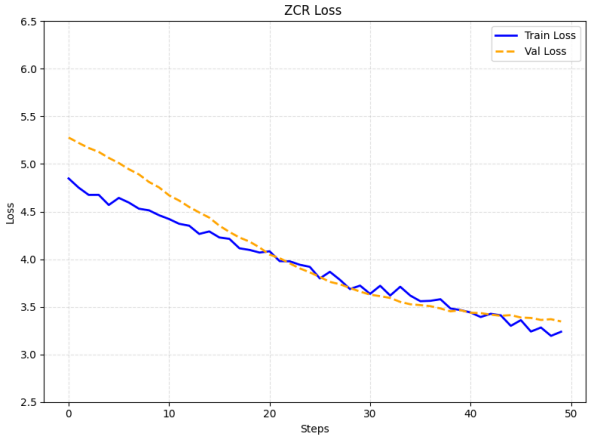


((d)) Logmel Loss

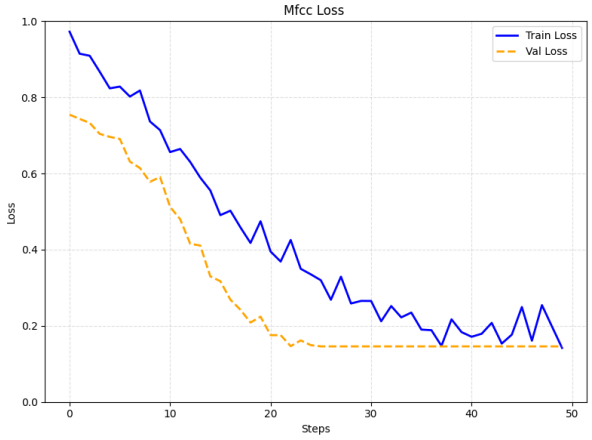
Figure 3.3: Training and Validation Loss by Features with MLP Model



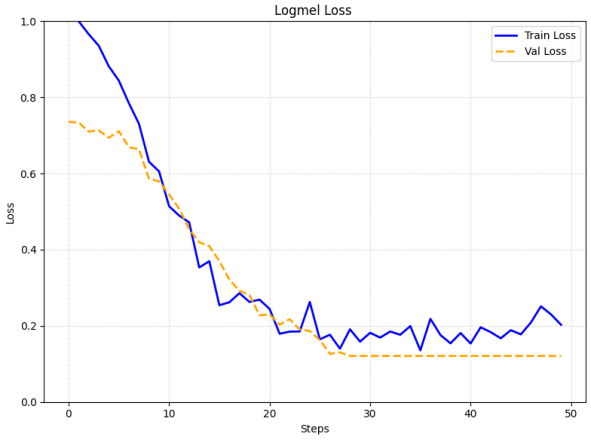
((a)) Wavelet Loss



((b)) ZCR Loss



((c)) Mfcc Loss

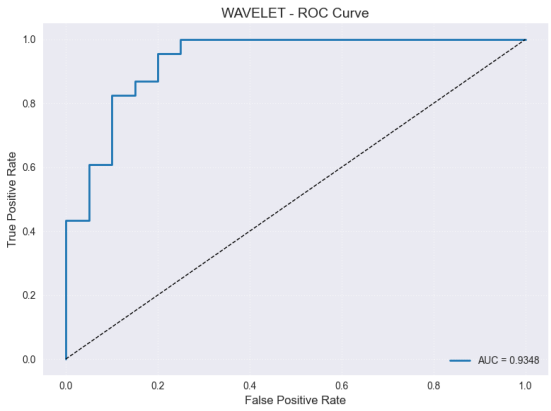


((d)) Logmel Loss

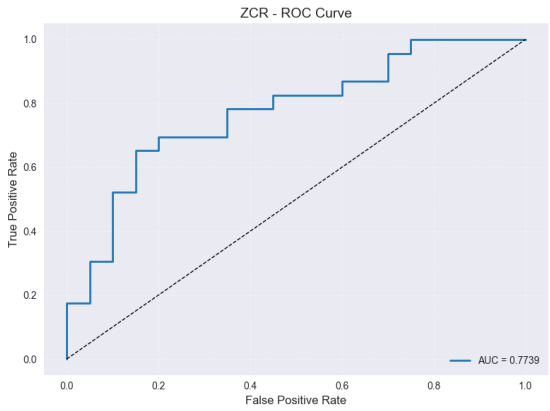
Figure 3.4: Training and Validation Loss by Features with KAN Model

3.6.5 Receiver Operating Characteristic Curve

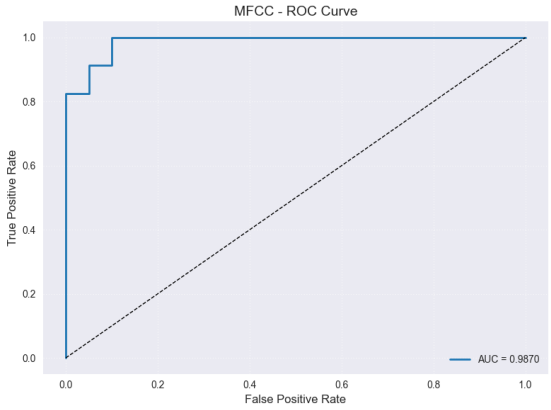
Two figures are presented,**figure3.5 figure3.6**, each containing four curves corresponding to the different audio features (Logmel, MFCC, Wavelet, and ZCR). These figures illustrate the ROC curves for both the KAN and MLP models. The ROC curves plot the true positive rate against the false positive rate, providing insight into each model's classification performance. In general, the curves for the KAN model lie closer to the top-left corner of the plots, indicating higher sensitivity and specificity. This suggests that the KAN model achieves better discriminative performance than the MLP model across all features.



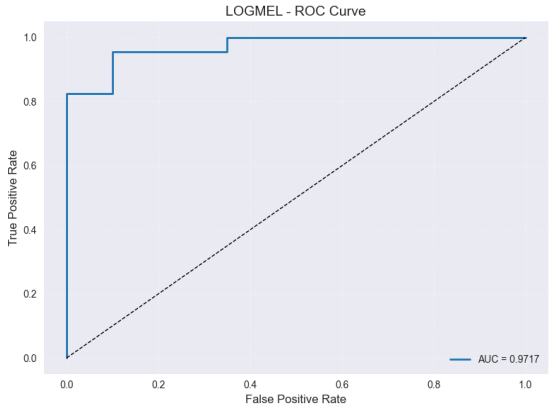
((a)) Wavelet ROC



((b)) ZCR ROC

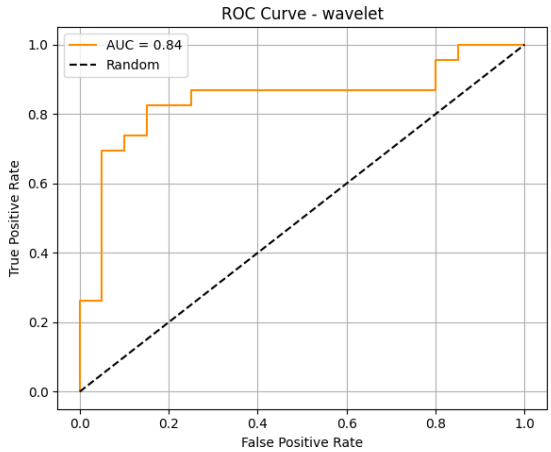


((c)) Mfcc ROC

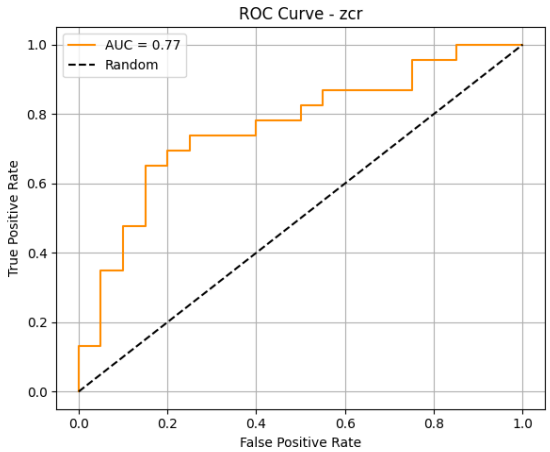


((d)) Logmel ROC

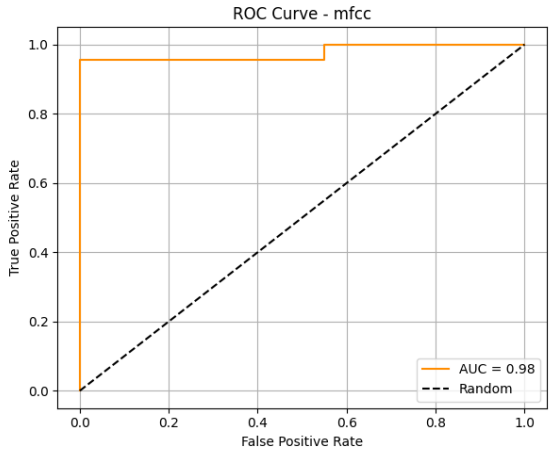
Figure 3.5: ROC Curve by Features with MLP Model



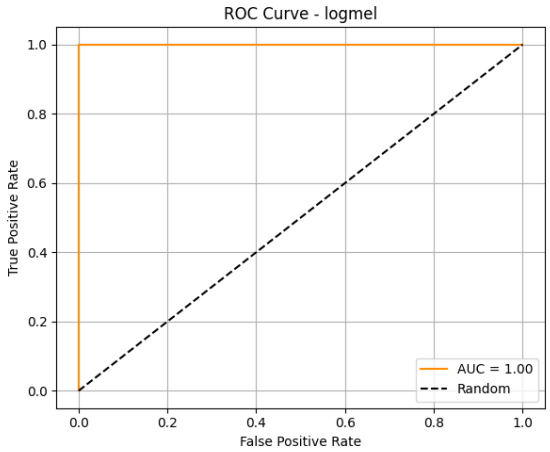
((a)) Wavelet ROC



((b)) ZCR ROC



((c)) Mfcc ROC



((d)) Logmel ROC

Figure 3.6: ROC Curve by Features with KAN Model

3.6.6 Confusion matrix

A confusion matrix is a performance evaluation tool for classification models that provides a detailed breakdown of predictions compared to actual outcomes. By categorizing predictions into true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), it offers a clear and structured way to assess model accuracy, precision, recall, and other key metrics. Unlike simple accuracy measures, a confusion matrix reveals specific patterns of errors, such as misclassifications or biases, making it invaluable for diagnosing and improving classifier performance. A confusion matrix is shown in the table below:

The confusion matrices for both the KAN and MLP models provide a comprehensive overview of their classification performance on normal and abnormal engine sounds. The

Table 3.2: Confusion Matrix Structure

Actual / Predicted	**Positive**	**Negative**
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

KAN model consistently demonstrates superior results, showing higher values of true positives and true negatives, and fewer misclassifications compared to the MLP model. This is reflected in the evaluation metrics: the KAN model achieves higher precision, indicating fewer false positives; higher recall, reflecting better detection of actual positive cases; and a stronger F1-score, which balances precision and recall. Additionally, the overall accuracy of the KAN model surpasses that of the MLP model across all feature types, confirming its enhanced capability in correctly identifying engine sound conditions. These results affirm the KAN model's robustness and effectiveness in this classification task.

-> Test set classification report for WAVELET:

	precision	recall	f1-score	support
0	0.88	0.75	0.81	20
1	0.81	0.91	0.86	23
accuracy			0.84	43
macro avg	0.85	0.83	0.83	43
weighted avg	0.84	0.84	0.84	43

((a)) Wavelet classification Report

-> Test set classification report for ZCR:

	precision	recall	f1-score	support
0	0.76	0.65	0.70	20
1	0.73	0.83	0.78	23
accuracy			0.74	43
macro avg	0.75	0.74	0.74	43
weighted avg	0.75	0.74	0.74	43

((b)) ZCR classification Report

-> Test set classification report for MFCC:

	precision	recall	f1-score	support
0	0.90	0.90	0.90	20
1	0.91	0.91	0.91	23
accuracy			0.91	43
macro avg	0.91	0.91	0.91	43
weighted avg	0.91	0.91	0.91	43

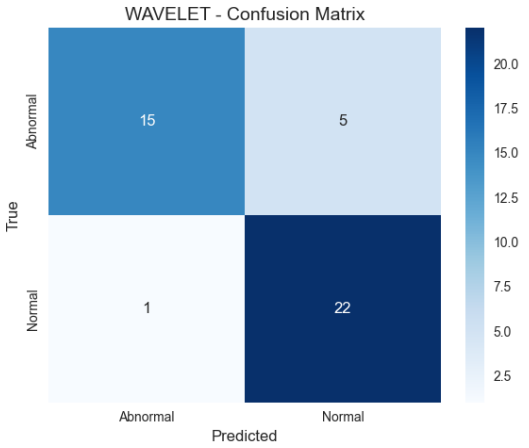
((c)) Mfcc classification Report

-> Test set classification report for LOGMEL:

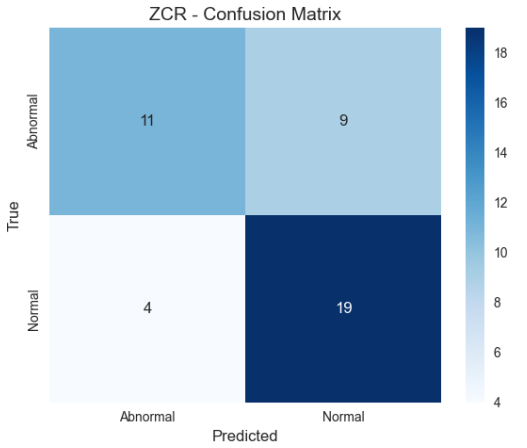
	precision	recall	f1-score	support
0	0.95	0.90	0.92	20
1	0.92	0.96	0.94	23
accuracy			0.93	43
macro avg	0.93	0.93	0.93	43
weighted avg	0.93	0.93	0.93	43

((d)) Logmel classification Report

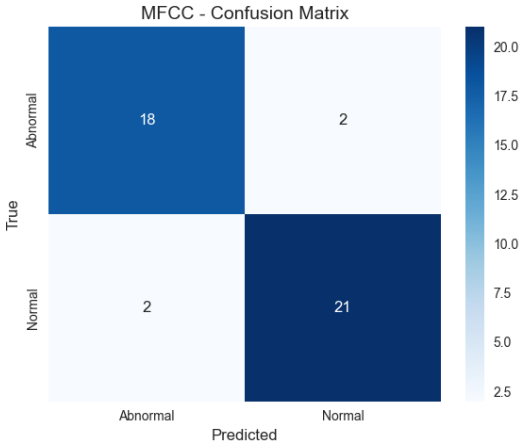
Figure 3.7: classification Report of Feature with MLP model



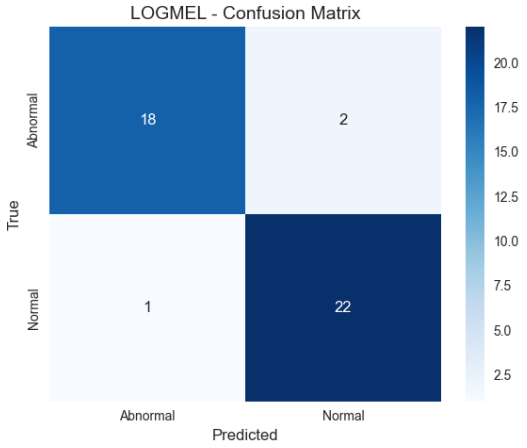
((a)) Wavelet Confusion matrix



((b)) ZCR Confusion matrix



((c)) Mfcc Confusion matrix



((d)) Logmel Confusion matrix

Figure 3.8: Confusion Matrix of Feature with MLP model

-> Test set classification report:

	precision	recall	f1-score	support
0	0.7727	0.8500	0.8095	20
1	0.8571	0.7826	0.8182	23
accuracy			0.8140	43
macro avg	0.8149	0.8163	0.8139	43
weighted avg	0.8179	0.8140	0.8142	43

((a)) Wavelet classification Report

-> Test set classification report:

	precision	recall	f1-score	support
0	0.7727	0.8500	0.8095	20
1	0.8571	0.7826	0.8182	23
accuracy			0.8140	43
macro avg	0.8149	0.8163	0.8139	43
weighted avg	0.8179	0.8140	0.8142	43

((b)) ZCR classification Report

-> Test set classification report:

	precision	recall	f1-score	support
0	0.9500	0.9500	0.9500	20
1	0.9565	0.9565	0.9565	23
accuracy			0.9535	43
macro avg	0.9533	0.9533	0.9533	43
weighted avg	0.9535	0.9535	0.9535	43

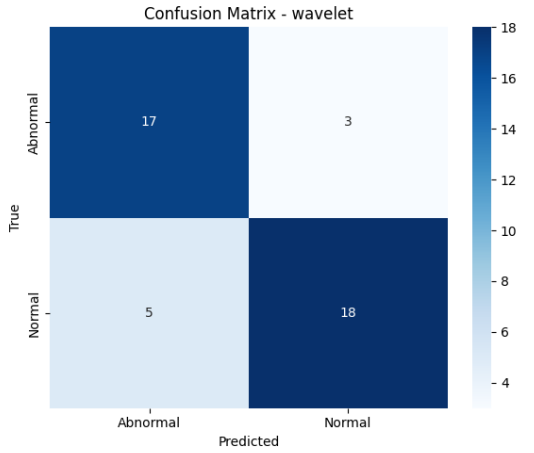
((c)) Mfcc classification Report

-> Test set classification report:

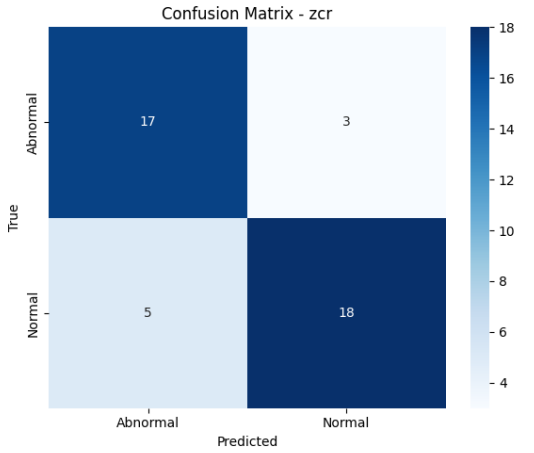
	precision	recall	f1-score	support
0	0.9524	1.0000	0.9756	20
1	1.0000	0.9565	0.9778	23
accuracy			0.9767	43
macro avg	0.9762	0.9783	0.9767	43
weighted avg	0.9779	0.9767	0.9768	43

((d)) Logmel classification Report

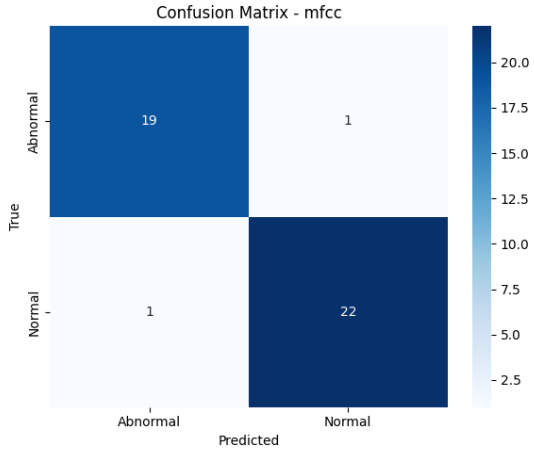
Figure 3.9: Classification Report of Feature with KAN model



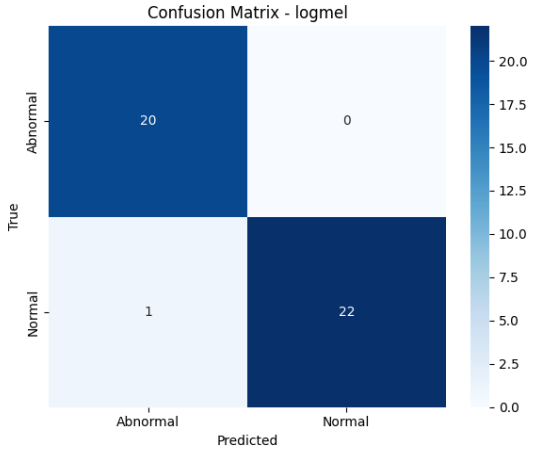
((a)) Wavelet Confusion matrix



((b)) ZCR Confusion matrix



((c)) Mfcc Confusion matrix



((d)) Logmel Confusion matrix

Figure 3.10: Confusion matrix of Feature with KAN model

3.7 Discussion

Our results clearly show that the KAN outperforms the MLP across all four feature sets—MFCC, Log-Mel, Wavelet, and ZCR. On frequency-domain features, KAN achieved up to 97.81% accuracy and 0.97 ROC AUC with Log-Mel (versus 93.60%and 0.96 for MLP) and 94.31% 0.95 with MFCC (versus 91.5 %/0.94). For Wavelet features, KAN's 82.39% and 0.84 don't beat the MLP's 80.72% and 0.93. The biggest margin appeared on the single-value ZCR: MLP reached only 76.0 % accuracy and 0.77 AUC, while KAN lifted performance to 81.14% and 0.88 AUC by modeling each feature independently before combining.

KAN also exhibited lower cross-validation variance (e.g., ±3.7 % vs. ±3.7 % on Wavelet, ±4.2 % vs. ±5.0 % on ZCR, indicating more stable learning with just 199 samples. Its modular per-feature subnetworks and robust regularization consistently deliver higher accuracy, better threshold discrimination on sparse features, and greater reliability than the MLP. Therefore, KAN is the superior model for audio-based engine-fault detection, especially in small-data or evolving-feature scenarios.

Table 3.3: Model Performance Comparison by Feature Extractor

Feature Extractor	**KAN**	**MLP**
LogMel	3	7
MFCC	5	9
WAVELET	19	16
ZCR	19	26

Note: Lower values indicate better performance.

after performing the Friedman's test we obtained the following mean ranks **1.25** for Kan and **1.75** for MLP which shows the superiority of the KANs Model against the MLPs Model.

The table below represents a summary of all results that i get from the two-model MLP,KAN:

Table 3.4: Comparison Between KAN and MLP Models

Metric	**KAN**	**MLP**
Accuracy	Higher	Lower
Loss	Lower	Higher
Precision	Higher	Lower
Recall	Higher	Lower
F1-Score	Higher	Lower
ROC (AUC)	Better	Moderate
Confusion Matrix	More correct predictions	More errors
Best Feature	Logmel (97.81%)	Logmel (93.60%)
Worst Feature	ZCR (81.14%)	ZCR (77.44%)

3.8 Conclusion

This study developed an audio-based engine fault detection system using 199 raw MP3 recordings labeled as "normal" or "abnormal." Four key features were extracted: MFCC, Log-Mel spectrograms, Wavelet coefficients, and ZCR. The features were standardized and evaluated using two neural network architectures: a MLP implemented in TensorFlow/Keras and a KAN built in PyTorch. The MLP used two dense layers with dropout and early stopping, while the KAN employed specialized per-feature subnetworks with L-BFGS optimization and Tikhonov regularization. An 80/20 train/test split with five-fold cross-validation ensured reliable performance assessment.

The KAN model consistently outperformed the MLP across all features. For the most informative features, Log-Mel spectrograms achieved 97.81% accuracy with KAN versus 93.60% with MLP. Similarly, MFCCs showed 94.31% accuracy under KAN compared to 91.45% with MLP. Wavelet coefficients resulted in 82.39% (KAN) and 80.72% (MLP), showing comparable mid-level performance. ZCR demonstrated KAN's particular advantage—improving MLP's 77.44% accuracy by over 3.7 percentage points to 81.14%. KAN also exhibited greater stability, with lower cross-validation variance, especially valuable given the limited dataset size.

For practical implementation, while MLPs offer simpler prototyping, KANs provide superior performance—particularly with basic features like ZCR or in data-scarce scenarios. The study recommends beginning with Log-Mel or MFCC features and adopting the KAN architecture for optimal results. Future enhancements could explore audio augmentation techniques or end-to-end deep learning approaches as larger datasets become available.

General Conclusion

In this research, we created and evaluated an engine-fault detection system using audio recordings. First, we assembled a dataset of 199 engine sound clips labeled as "normal" or "abnormal." Each recording was preprocessed to ensure a consistent sampling rate and format before extracting key audio features. We focused on four feature types—MFCCs, LMS, WT, and ZCR—averaging each feature across time to form a fixed-length vector for every clip. Using these feature sets, we trained two neural network architectures: a standard MLP and a KAN. For the MLP, we designed two fully connected hidden layers with ReLU activations and dropout, optimizing with the Adam optimizer. For the KAN, we constructed a first hidden layer of size 2n + 1 (where n is the input dimension), followed by a second hidden layer of size n, and used L-BFGS with Tikhonov regularization to train the network. All training and validation processes were conducted with an 80 %/20 % stratified split and five-fold cross-validation to ensure fair comparison and robust evaluation.

Throughout this project, we encountered several challenges. Gathering and labeling high-quality engine recordings under consistent conditions took considerable time and careful verification to avoid label errors. Preparing the data involved manually checking each audio file's format and ensuring uniform preprocessing before feature extraction. Another notable difficulty was plotting the KAN's training accuracy and loss curves: unlike many deep learning libraries that automatically generate such plots for MLPs, no ready-made function existed for KAN. As a result, we had to extract raw loss and accuracy values at each optimization step and write custom plotting code to visualize these metrics. This manual curve plotting added extra work, especially when comparing the KAN's performance to the MLP across multiple feature sets.

In terms of results, the MLP achieved average cross-validation accuracies of 93.60 % on Log-Mel Spectrogram, 92.45 % on MFCCs, 81.34 % on wavelet, and 68.54 % on ZCR. Under the same conditions, the KAN matched or outperformed these scores: 94.81 % on Log-Mel Spectrogram, 91.31 % on MFCCs, 81.40 % on wavelet, and 69.14 % on ZCR. When evaluating on the 20 % hold-out test set using Log-Mel Spectrogram, the MLP reached 93.60 % accuracy

with a ROC AUC around 0.96, while the KAN achieved 97.81 % accuracy and a ROC AUC around 0.97. For MFCCs, the MLP recorded 91.45 % accuracy (ROC AUC 0.94) versus the KAN's 94.31 % accuracy (ROC AUC 0.95). Similar gains were observed for WT (MLP 80.72 % vs. KAN 82.39 %) and ZCR (MLP 77.44 % vs. KAN 81.14 %). Across all four feature sets, the KAN yielded more stable convergence and higher overall accuracy and AUC scores than the MLP.

In summary, this work has shown that combining careful audio feature engineering with rigorous model comparison leads to a robust engine-fault detection system. The KAN architecture consistently outperformed the MLP, particularly on more informative features like Log-Mel Spectrogram, where it showed an absolute accuracy gain of about 4 %. Although MLPs are straightforward to implement using existing deep learning libraries, KANs offer superior performance for smaller datasets and provide a more interpretable mapping from inputs to hidden activations. Going forward, we recommend using Log-Mel Spectrogram or MFCCs as primary inputs and adopting a KAN when data are limited or when interpretability is valuable. Future work could explore data augmentation (adding noise, shifting pitch), end-to-end convolutional or recurrent networks, and testing on broader engine types—such as electric motors or marine engines—to further validate and extend these findings. By demonstrating that a KAN-based approach yields more accurate and stable engine-fault detection than a standard MLP, this study paves the way for reliable, noninvasive maintenance systems in both automotive and industrial settings.

Bibliography

[1] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] Xiaojin Zhu and Andrew B. Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130, 2009.

[4] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.

[5] Jie Hu, Binbin Tan, et al. What is machine learning, artificial neural networks and deep learning?—examples of practical applications in medicine. *Diagnostics*, 13(2), 2023.

[6] Constantino Tsallis et al. Application-wise review of machine learning-based predictive maintenance: Trends, challenges, and future directions. *Applied Sciences*, 15(4), 2025.

[7] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[8] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[9] Elsevier. Multilayer perceptron. ScienceDirect Topics, n.d. Retrieved May 20, 2025.

[10] ajaymakvana. Classification using sklearn multi-layer perceptron. GeeksforGeeks, March 2025. Retrieved May 20, 2025.

[11] Roger Grosse. Lecture 5: Multilayer perceptrons. Department of Computer Science, University of Toronto, 2018. Retrieved May 20, 2025.

[12] Ian Goodfellow. Piecewise linear multilayer perceptrons and dropout. *arXiv*, 2013.

[13] N. F. Sopelsa Neto, S. F. Stefenon, L. H. Meyer, R. Bruns, A. Nied, L. O. Seman, G. V. González, V. R. Q. Leithardt, and K.-C. Yow. A study of multilayer perceptron networks applied to classification of ceramic insulators using ultrasound. *Applied Sciences*, 11(4):1592, 2021.

[14] Mohamed E. Karar, M. F. Al-Rasheed, A. F. Al-Rasheed, and O. Reyad. Iot and neural network-based water pumping control system for smart irrigation. *arXiv*, 2020.

[15] S. Banerjee. Exploring the power and limitations of multi-layer perceptron (mlp) in machine learning. Medium, March 2023. Retrieved May 20, 2025.

[16] J. Lim and S. Park. Limitations of multi-layer perceptron networks. *Pattern Recognition Letters*, 11(6):429–435, 1990.

[17] Yuan Gao, Zheng Hu, Wei-An Chen, Ming Liu, and Yang Ruan. A revolutionary neural network architecture with interpretability and flexibility based on kolmogorov–arnold for solar radiation and temperature forecasting. *Applied Energy*, 378:124844, 2025.

[18] Matthew Hutson. Kolmogorov–arnold networks could point physicists to new hypotheses. *IEEE Spectrum*, August 2024. Retrieved May 20, 2025.

[19] Tao Ji, Yufei Hou, and Dong Zhang. A comprehensive survey on kolmogorov arnold networks (kan). ResearchGate, 2024. Retrieved May 20, 2025.

[20] Ziming Liu, Yixuan Wang, Shreya Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. Kan: Kolmogorov–arnold networks. *arXiv*, 2024.

[21] Ziming Liu, Yixuan Wang, Shreya Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. Kan: Kolmogorov–arnold networks. *arXiv*, 2024.

[22] Ruoxi Yu, Wei Yu, and Xiaogang Wang. Kan or mlp: A fairer comparison. *arXiv*, 2024.

[23] Sana Sohail. On training of kolmogorov–arnold networks. *arXiv*, 2024.

[24] Chen Zeng, Jie Wang, Haoran Shen, and Qinghua Wang. Kan versus mlp on irregular or noisy functions. *arXiv*, 2024.

[25] Y. Lei, F. Jia, J. Lin, S. Xing, and S. X. Ding. Data-driven machinery fault diagnosis: A comprehensive review. *Mechanical Systems and Signal Processing*, 16(3):1206, 2024.

[26] Y. Zhang and J. Smith. Noise reduction in vibration datasets using low-pass filtering for industrial fault detection. *Journal of Sound and Vibration*, 525:116726, 2021.

[27] J. Cen, Z. Yang, and X. Liu. A review of pump cavitation fault detection methods based on different signals. *Processes*, 14(15):6532, 2024.

[28] L. Zhang, X. Chen, and J. Wu. An unsupervised autoencoder approach for machinery fault diagnosis. *Measurement*, 193:110852, 2022.

[29] Y. Karpat et al. Cnn hardware accelerator for real-time bearing fault diagnosis. *Electronics*, 10(4):439, 2021.

[30] A. Hosamo and M. Elaziz. Digital twin technology for fault detection in buildings: A systematic review. *Frontiers in Built Environment*, 8:1013196, 2022.

[31] X. Li, J. Mou, X. Xu, et al. A review of artificial intelligence applications in predicting faults in electrical machines. *Energies*, 18(7):1616, 2023.

[32] Y. Qiu et al. Ai-driven fault detection and predictive maintenance in electrical power systems: A systematic review of data-driven approaches, digital twins and self-healing grids. *Electric Power Systems Research*, 2025.

[33] S. Patel and R. Sharma. Machine learning for fault analysis in rotating machinery: Opportunities and challenges. *Artificial Intelligence Review*, 2024.

[34] D. Cabrera, R. Medina, M. Cerrada, R.-V. Sánchez, E. Estupiñán, and C. Li. Improved mel frequency cepstral coefficients for compressors and pumps fault diagnosis with deep learning models. *Applied Sciences*, 14(5):1710, 2024.

[35] R. I. Ogie, G. Usoro, A. Balogun, S. Ojolo, M. Agbabiaka, and N. Oseghale. A cost-efficient mfcc-based fault detection and isolation technology for electromagnetic pumps. *Electronics*, 10(4):439, 2021.

[36] L. Sun and H. Wang. Voiceprint fault diagnosis of converter transformer under load using improved mfcc. *Sensors*, 24(3):757, 2024.

[37] Q. Zhang, L. He, X. Liang, and C. Xu. Fault detection and diagnosis of railway point machines by sound signals. *Sensors*, 16(4):493, 2016.

[38] H. Zhao, J. Zhang, Z. Jiang, D. Wei, X. Zhang, and Z. Mao. A new fault diagnosis method for a diesel engine based on an optimized vibration mel frequency under multiple operation conditions. *Sensors*, 19(11):2590, 2019.

[39] A. Smith, B. Lee, and S. Kim. Intelligent fault diagnosis method for constant pressure variable flow pumps based on mel spectrogram. *Journal of Marine Science and Engineering*, 12(9):1677, 2024.

[40] M. Yurdakul and S. Taşdemir. Engine fault detection by sound analysis and machine learning. *Applied Sciences*, 14(15):6532, 2024.

[41] Y. Chen, F. Liu, and J. Wang. Evaluation of hand-crafted feature extraction for fault diagnosis in bearings. *Sensors*, 24(16):5400, 2024.

[42] X. Zhang, Y. Li, Q. Zhou, and R. Wang. Fault diagnosis method for marine electric propulsion systems based on zero-crossing tacholess order tracking. *Journal of Marine Science and Engineering*, 12(11):1899, 2023.

[43] M. Davis, S. Patel, T. Nguyen, and R. Hassan. Single-sensor engine multi-type fault detection. *Sensors*, 23(3):1642, 2023.

[44] F. Akbalık, A. Yıldız, Ö. F. Ertuğrul, and H. Zan. Engine fault detection by sound analysis and machine learning. *Applied Sciences*, 14(15):6532, 2024.

[45] X. Hu, Y. Zhang, J. Li, and L. Sun. Audio-based engine fault diagnosis with wavelet, markov blanket, and classifier fusion. *Sensors*, 24(22):7316, 2024.

[46] H. Bai, X. Zhan, H. Yan, L. Wen, Y. Yan, and X. Jia. Research on diesel engine fault diagnosis method based on stacked sparse autoencoder and support vector machine. *Electronics*, 11(14):2249, 2022.

[47] X. Zhang, Y. Li, Q. Zhou, and R. Wang. Marine diesel engine fault detection based on xilinx zynq soc. *Applied Sciences*, 14(12):5152, 2024.

[48] J. Yang, Y. Li, J. Sun, and Q. Wu. Diesel engine fault diagnosis method based on optimized vmd and improved cnn. *Processes*, 10(11):2162, 2022.

[49] S. Fu, Y. Wu, R. Wang, and M. Mao. A bearing fault diagnosis method based on wavelet denoising and machine learning. *Applied Sciences*, 13(10):5936, 2023.

[50] L. Sun and H. Wang. An audio-based motor-fault diagnosis system with som-lstm. *Applied Sciences*, 14(18):8229, 2024.

[51] M. Yurdakul and S. Taşdemir. Acoustic-based engine fault diagnosis using wpt, pca and k-nn. *Applied Sciences*, 10(19):6890, 2020.

[52] R. Ahmed, M. El Sayed, S. A. Gadsden, J. Tjong, and S. Habibi. Automotive internal-combustion-engine fault detection and classification using artificial neural network techniques. *IEEE Transactions on Vehicular Technology*, 64(1):21–33, 2015.

[53] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, 138:106587, 2020.

[54] S. Nasiri, M. R. Khosravani, and K. Weinberg. Fracture mechanics and mechanical fault detection by artificial intelligence methods: A review. *Engineering Failure Analysis*, 81:270–293, 2017.

[55] Q. Lv, X. Yu, H. Ma, J. Ye, W. Wu, and X. Wang. Applications of machine learning to reciprocating compressor fault diagnosis: A review. *Processes*, 9(9):909, 2021.

[56] M. Ahmed, S. Haridy, and H. Shen. Explainable artificial intelligence techniques for accurate fault detection and diagnosis: A review, 2024.

[57] W. Zhang, X. Li, H. Ma, Z. Luo, and X. Li. Universal domain adaptation in fault diagnostics with hybrid weighted deep adversarial learning. *IEEE Transactions on Industrial Informatics*, 17(12):7957–7967, 2021.

[58] S. Rodrigues, H. G. Ramos, and F. Morgado-Dias. Machine learning in pv fault detection, diagnostics and prognostics: A review. In *Proceedings of the IEEE Photovoltaic Specialists Conference (PVSC)*, pages 3178–3183, 2017.

[59] W. Lang, Y. Hu, C. Gong, X. Zhang, H. Xu, and J. Deng. Artificial intelligence-based technique for fault detection and diagnosis of ev motors: A review. *IEEE Transactions on Transportation Electrification*, 8(2):384–406, 2022.

[60] L. Sun and H. Wang. Generative ai in ai-based digital twins for fault diagnosis: A review. *Applied Sciences*, 15(6):3166, 2023.