



CHALMERS
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG

Intelligent Sound Diagnostics

Acoustic Fault Detection in Truck Steering Systems Using Machine Learning and Deep Learning

Master's thesis in Computer science and engineering

Karthikeyan Natarajan
Kiruthiga Krishnamoorthy

MASTER'S THESIS 2026

Chalmers University of Technology

Intelligent Sound Diagnostics

Karthikeyan Natarajan
Kiruthiga Krishnamoorthy



UNIVERSITY OF
GOTHENBURG



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2026

© Karthikeyan Natarajan,
Kiruthiga Krishnamoorthy
2026.

Supervisor: Kivanç Tatar, Computer Science and Engineering Department
Examiner: Simon Olsson, Computer Science and Engineering Department
Advisors: Jin Zhou, Patrik Johansson, Volvo Trucks

Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Gothenburg, Sweden 2026

KARTHIKEYAN NATARAJAN
KIRUTHIGA KRISHNAMOORTHY

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

This thesis investigates acoustic fault diagnosis for heavy duty truck steering components, focusing on two safety critical parts, the Power Steering Pump (PSP) and the Steering Gear, using a real world dataset of 326 recordings collected under operational conditions. The problem is formulated as a seven class classification task covering four sound types under normal and faulty conditions. Classical machine learning models trained on hand crafted acoustic features are compared against a convolutional neural network trained from scratch and Pretrained Audio Neural Network (PANN) models used as frozen feature extractors, to identify which modelling approach is most effective for this task. Seven individual acoustic descriptors, MFCC, log-Mel spectrogram statistics, sub-band energy ratios, spectral shape descriptors, time-domain statistics, amplitude envelope kurtosis, and discrete wavelet transform coefficients, are also evaluated in isolation to determine which is most discriminative for fault detection. All models are evaluated using StratifiedGroupKFold cross validation with file level grouping to prevent leakage between overlapping audio segments, and performance is reported using the Matthews Correlation Coefficient (MCC) and macro F1, both robust to class imbalance. A linear Support Vector Machine trained on a 180-feature ANOVA selected set achieves the strongest overall performance, reaching a held out test MCC of 1.000, ahead of every deep learning approach evaluated, including a pretrained PANN backbone with over 80 million parameters. A compact CNN trained entirely from scratch matches the best pretrained PANN variant despite requiring no external pretraining, while the largest pretrained backbone performs weakest of all models compared. Among the individual descriptors, log-Mel spectrogram statistics and MFCCs prove the most discriminative, while amplitude envelope kurtosis alone carries almost no discriminative power. These findings suggest that carefully engineered acoustic features combined with a simple classical classifier can outperform learned and transferred deep representations on a dataset of this scale, offering a lightweight and interpretable alternative for industrial predictive maintenance.

Keywords: acoustic fault diagnosis, machine learning, convolutional neural networks, pretrained audio models, PANN, MFCC, log-Mel spectrogram, DWT, Stratified-GroupKFold, heavy duty trucks, steering system.

Acknowledgements

We thank to our supervisor, examiner, company advisors and others who encourage us.

Karthikeyan Natarajan,
Kiruthiga Krishnamoorthy
Gothenburg, 2026-10-22

Contents

List of Figures	xii
List of Tables	xiii
1 Introduction	1
1.1 Context	1
1.2 Problem Statement	3
1.3 Research Gap and Positioning	3
1.4 Research Questions	4
1.5 Contributions	4
1.6 Limitations	5
2 Background	6
2.1 Sound Based Fault Diagnosis	6
2.2 Audio Signal Processing for Machine Sounds	8
2.2.1 Sampling and the Sampling Theorem	8
2.2.2 Windowing and the Short-Time Fourier Transform	8
2.2.3 Segmentation of Recordings into Fixed Length Clips	9
2.3 Acoustic Feature Representations	9
2.4 Learning Paradigms for Acoustic Fault Detection	10
2.4.1 Classical Ensemble Learning and Stacked Generalisation	10
2.4.2 Data Augmentation via Same-Class Mixup	11
2.4.3 Convolutional Neural Networks and Transfer Learning via Pretrained Audio Models	11
2.5 Challenges in Real World and Small Data Scenarios	11
3 Theory	13
3.1 Short-Time Fourier Transform	13
3.2 Acoustic Feature Extraction	14
3.2.1 Mel-Frequency Cepstral Coefficients (MFCC)	14
3.2.1.1 Windowing and STFT:	14
3.2.1.2 Mel Filterbank:	14
3.2.1.3 Logarithmic Compression:	15
3.2.1.4 Discrete Cosine Transform (DCT):	15
3.2.1.5 Coefficient Truncation:	15
3.2.2 Log-Mel Spectrogram Features	16

3.2.3	Sub-band Energy Ratios	16
3.2.4	Spectral Descriptors	16
3.2.4.1	Spectral Centroid:	16
3.2.4.2	Spectral Bandwidth:	17
3.2.4.3	Spectral Roll-off	17
3.2.4.4	Spectral Flatness:	17
3.2.4.5	Spectral Contrast	17
3.2.5	Amplitude Envelope Kurtosis	17
3.2.6	Discrete Wavelet Transform Features	17
3.2.7	Time-Domain Features	18
3.3	Temporal Aggregation	18
3.4	ANOVA Based Feature Selection	19
3.5	Data Augmentation Strategies	19
3.6	Convolutional Neural Networks and Transfer Learning	21
3.7	Stacked Generalisation	21
3.8	Hyperparameter Optimisation	22
3.9	Evaluation Metrics	22
4	Related Work	24
4.1	Feature Extraction for Sound Analysis	25
4.2	Feature Selection	26
4.3	Data Augmentation	26
4.4	Base Classifier Selection	27
4.5	Stacked Generalisation	28
4.6	Hyperparameter Optimisation and Evaluation	29
5	Methods	30
5.1	Dataset	30
5.2	Preprocessing	32
5.3	Feature Extraction	32
5.4	Feature Selection	33
5.5	Data Augmentation	33
5.6	Classical Model Training	34
5.7	Deep Learning Model Training	37
5.8	Cross Validation and Held-Out Test Evaluation	39
6	Results and Discussion	41
6.1	RQ1: Model Comparison Across Classical, Convolutional, and Pre-trained Approaches	41
6.1.1	Classical Machine Learning: Round-by-Round Results	41
6.1.2	Deep Learning: CNN Trained from Scratch and Pretrained PANN	42
6.1.3	Overall RQ1 Comparison	43
6.1.4	Out-of-Fold Versus Held-Out Test Gap	45
6.2	Stacking Gain Analysis	45
6.3	RQ2: Individual Descriptor Comparison	47
6.4	Per-Class Performance	49

6.5	Feature Importance and Explainability	50
6.6	Model Efficiency and Deployment Considerations	53
6.7	Summary of Findings	55
7	Conclusion	56
7.1	Findings	56
7.2	Summary	58
7.3	Limitations and Future Work	59
7.4	Use of AI Tools	60
7.5	Ethics	61
	References	61

List of Figures

2.1	Representative waveform and spectrogram for each sound type: Moan, Whine, Grunt, and Hiss	7
5.1	Dataset overview	31
5.2	ANOVA Feature Selection	33
5.3	Round 1 Baseline results	35
5.4	Round 3 final out-of-fold MCC	36
5.5	Out-of-fold MCC	37
6.1	Held out test F1 and MCC for all models	43
6.2	Held out test MCC	44
6.3	Stacking Gain	46
6.4	Descriptor groups and models	48
6.5	Confusion matrix for LinearSVM	49
6.6	Global feature importance	50
6.7	Feature group importance	52
6.8	Top individual features by mean absolute SHAP value.	52
6.9	Training time, estimated energy consumption, model size, and inference latency for all six base classifiers.	53
6.10	Estimated Training Energy	54

List of Tables

2.1	Acoustic characteristics of the four fault sound types.	7
3.1	STFT parameters used, $f_s = 48$ kHz, $N = 4096$, $H = 1024$, $L = 96,000$ samples (2.0 s clip).	14
3.2	Sub-band energy ratio bands and fault associations, $N = 4096$, $f_s = 48$ kHz.	16
3.3	Feature group dimensions after aggregation.	19
4.1	Hyperparameter search spaces for each base classifier	28
5.1	Dataset summary statistics	30
5.2	Feature extraction summary	32
5.3	StratifiedGroupKFold class distribution per fold (training pool, 2519 clips).	39
6.1	Out-of-fold MCC for all six base classifiers and the stacked meta-learner, using the full 180-feature ANOVA selected feature set. (T) indicates a model tuned with Optuna.	42
6.2	Cross validation and held out test results for the convolutional and pretrained deep learning models.	42
6.3	RQ1 summary: best model from each family, ranked by held out test MCC.	43
6.4	Out-of-fold MCC versus held out test MCC for all classical base classifiers and the stacked meta-learner.	45
6.5	Stacking gain: stacked meta-learner MCC minus best single base classifier MCC.	45
6.6	RQ2 summary: best model per descriptor group, evaluated in isolation, file level out-of-fold MCC.	47
6.7	Per-class precision, recall, and F1 for LinearSVM on the held out test set (clip level).	49
6.8	Model efficiency for the classical base classifiers and the stacked ensemble, measured during final model fitting on the full training pool.	53
6.9	Overall summary: best configuration identified in this thesis.	55

1

Introduction

Consider a long haul truck driver navigating a highway when an unfamiliar sound emerges from the steering system, a subtle hiss or moan that was not there before. Is it a minor issue or an early warning of a critical failure? The ability to identify and interpret such acoustic signatures before they escalate into costly breakdowns is at the heart of predictive maintenance. While experienced mechanics can often diagnose faults by listening to a vehicle, translating this expertise into an automated, intelligent system remains a significant challenge.

Heavy duty trucks operate continuously under demanding and highly variable conditions, where unexpected mechanical failures can result in significant economic losses, safety risks, and unplanned downtime [1]. As fleet sizes grow and vehicle architectures become increasingly complex, traditional schedule based maintenance strategies are no longer sufficient [1]. Predictive maintenance, enabled by continuous condition monitoring, has therefore become a critical research and industrial priority [2]. Acoustic signals offer a practical and non contact alternative that naturally captures mechanical interactions, fluid dynamics, and structural vibrations, often making faults detectable well before they escalate into critical failures [3].

Our thesis focuses on two safety critical steering components, the Power Steering Pump (PSP) and the Steering Gear, selected because they fail frequently enough to affect vehicle uptime and produce acoustically distinct signatures under different fault conditions. The problem is formulated as a seven class supervised classification task covering four sound types under normal and faulty operating conditions.

1.1 Context

Acoustic based fault diagnosis has been approached through a wide range of feature representations and modelling strategies [4], and no single method has established itself as universally superior across industrial domains. Surveys on acoustic source identification highlight that combining signal processing techniques with machine learning leads to more reliable diagnostic systems across diverse industrial environments, since the choice of feature representation strongly shapes how well a downstream classifier can separate normal operation from emerging faults [5].

Among the most established feature extraction methods, Mel Frequency Cepstral Coefficients (MFCC) have been widely adopted for capturing the perceptually relevant

spectral characteristics of audio signals, forming the foundation of many sound classification pipelines. Their popularity stems from their ability to compress a high dimensional spectrum into a compact, perceptually meaningful representation, which has made them a default choice well beyond their original speech processing context [6]. Building on this spectral view, wavelet based methods remain particularly relevant for mechanical fault diagnosis [7], as they are well suited to capturing transient and non stationary signal characteristics that arise during component degradation, a property that purely spectral descriptors can struggle to resolve when a fault manifests as a short, localised event rather than a sustained tonal change.

Beyond feature choice, the modelling strategy itself has a substantial impact on diagnostic performance. Studies on audio based machine learning for vehicle applications demonstrate that combining interpretable acoustic features with supervised classifiers leads to competitive fault detection performance in real world operational conditions, suggesting that carefully engineered features can rival more complex learned representations when data is limited and interpretability matters [8]. Ensemble strategies have also been explored as a way to combine the strengths of multiple classifiers. Stacked generalisation frameworks, where predictions from multiple base learners are combined through a meta learner, have been shown to improve classification robustness and reduce variance compared to individual models [9]. Gradient boosting methods [10] combined with automated hyperparameter optimisation [11] further strengthen ensemble performance, particularly in settings with limited data, where careful tuning can meaningfully offset the disadvantage of a small training set. In a related direction, data augmentation techniques such as MixUp, which construct synthetic training samples through convex combinations of real examples, have been proposed as an effective strategy for improving model generalisation when labelled training data is scarce, by encouraging smoother decision boundaries during training [12].

The way raw recordings are segmented before feature extraction is equally consequential. The choice of temporal segmentation strategy directly influences model performance in acoustic tasks [13]. Sliding window approaches introduce dependencies between adjacent clips, since consecutive windows from the same recording share most of their underlying audio content, making file level grouping in cross validation essential for obtaining unbiased performance estimates [14]. Benchmark datasets such as Malfunctioning Industrial Machine Investigation and Inspection (MIMII) have supported systematic evaluation of acoustic anomaly detection methods under controlled conditions, though real world vehicle recordings introduce additional variability, background noise, and operating-condition drift that such curated datasets may not fully capture [15], motivating the use of a real, operationally collected dataset in this thesis rather than a laboratory benchmark.

Once a rich feature set has been extracted, selecting the subset that is genuinely informative becomes important, both to reduce overfitting risk on a small dataset and to improve interpretability. Analysis of Variance (ANOVA) based feature selection has been shown to effectively identify the most discriminative features for bearing and mechanical fault diagnosis, reducing dimensionality while preserving classification accuracy [5]. Careful feature selection is only useful, however, if

paired with an evaluation protocol that reports it fairly. In settings where class distributions are unbalanced, as is typical for fault datasets where healthy operation is far more common than any individual fault type, evaluation metrics such as the Matthews Correlation Coefficient known as MCC provide a more reliable measure of classifier performance than accuracy or F1 score alone, since accuracy in particular can be misleadingly high on imbalanced data even for a classifier that largely ignores minority classes [16]. Finally, given the number of models, feature groups, and hyperparameter configurations considered in this kind of comparative study, automated hyperparameter optimisation frameworks such as Optuna enable efficient search over complex model configurations, making them well suited to tuning multiple base learners within a single, consistent pipeline [11]. Where deep learning models are used in this thesis, they build directly on pretrained audio neural networks (PANNs), convolutional architectures originally trained on a large scale, general purpose sound corpus [17], used here as frozen feature extractors and compared directly against models trained entirely from scratch.

1.2 Problem Statement

Despite advances in sound based fault diagnosis, several challenges remain unresolved for real world vehicle applications:

- Acoustic recordings collected in operational environments exhibit significant noise and variability.
- The number of labeled fault recordings is limited, increasing the risk of overfitting.
- Sliding window segmentation produces clips that are not independent observations, requiring careful cross validation design to avoid data leakage.
- There is a lack of systematic comparisons between classical ensemble methods and deep audio models using the same dataset and evaluation protocol.
- It is rarely made explicit which individual acoustic descriptor actually drives classification performance, since most pipelines combine many feature types into a single vector without isolating their individual contribution.

1.3 Research Gap and Positioning

The literature reviewed above shows steady progress on individual pieces of the acoustic fault diagnosis problem: feature representation, segmentation strategy, ensemble modelling, and evaluation methodology have each been studied in some depth. What is largely missing, however, is a single, controlled comparison that places classical machine learning, convolutional networks trained from scratch, and large pretrained audio models side by side on the same real world dataset, under the same file level, group aware evaluation protocol. Most existing studies either focus on one modelling family in isolation, or evaluate on curated benchmark datasets that

do not reflect the noise and variability of recordings collected directly from operating vehicles.

A similar gap exists at the feature level. While hand crafted acoustic descriptors such as MFCC, log-Mel spectrograms, and wavelet coefficients are all individually well established, they are almost always combined into a single feature vector before being passed to a classifier. This makes it difficult to say which descriptor is actually responsible for a model’s diagnostic performance, and whether some descriptors contribute meaningfully at all.

We position our thesis to address both gaps directly. We compare classical ensemble methods, a convolutional network (CNN) trained from scratch, and a pretrained audio model (PANN) under one unified protocol, and we isolate seven individual acoustic descriptor types to evaluate their standalone discriminative power.

1.4 Research Questions

1. **Which machine learning or deep audio model performs most effectively for detecting truck faults from audio recordings?** This question compares classical ensemble methods, convolutional networks, and pretrained audio models (PANNs) on the same dataset, examining how each represents spectral and temporal structure in truck sounds and how consistently it maintains performance across different operating conditions.
2. **Which audio descriptor provides the most robust and discriminative information for fault detection in truck acoustic signals?** This question compares seven individual descriptor types, MFCC, log-mel spectrogram, sub-band energy, spectral shape features, time-domain statistics, envelope kurtosis, and wavelet (DWT) coefficients, examining how each captures transient and steady fault patterns and how consistently it performs across the naturally varying real-world recordings in the dataset.

1.5 Contributions

Our thesis makes the following contributions:

- We design and implement a complete acoustic fault diagnosis pipeline for real world truck recordings, spanning preprocessing, hand crafted feature extraction, feature selection, model training, and file level, group aware evaluation.
- We conduct a systematic, controlled comparison between classical machine learning models trained on hand crafted acoustic features, a convolutional neural network trained from scratch, and a pretrained audio model (PANN) used as a frozen feature extractor, evaluated under an identical cross validation and test protocol. Our results show that a classical linear model outperforms every deep learning approach evaluated, including the largest pretrained backbone, indicating that for a dataset of this size and specificity, engineered features

combined with a simple classifier can be more effective than either learned or transferred deep representations.

- We investigate acoustic descriptor design directly, rather than only as a component of a larger combined feature vector, by isolating and evaluating seven individual descriptor types on their standalone ability to discriminate between healthy and faulty classes. Our results identify log-Mel spectrogram statistics and MFCCs as the most discriminative descriptors, and amplitude envelope kurtosis as the least, which indicates that not all commonly used acoustic descriptors contribute equally to fault detection performance.
- We evaluate all models not only on classification performance, but also on practical deployment considerations, including model size, inference latency, and estimated energy consumption, providing a more complete picture of the trade offs involved in choosing a modelling approach for industrial predictive maintenance.
- We complement the quantitative comparison with a feature importance analysis using SHapley Additive exPlanations (SHAP) [18], identifying which individual features and descriptor groups drive each model’s predictions, and which features are most associated with each class.

1.6 Limitations

The dataset consists of 326 recordings from operational trucks, which are segmented into several thousand overlapping clips through a sliding window procedure. These clips are not independent observations, and file level grouping in cross validation is essential for obtaining honest performance estimates, a principle applied consistently throughout this work. The analysis is restricted to two steering components and four sound types, and the held out test set, while never used during training or model selection, is comparatively small, which means single test set results should be interpreted alongside the cross validation results rather than in isolation. Recordings were collected under real operational conditions with varying background noise, which limits the ability to systematically isolate the effect of individual noise sources on model performance. Finally, all reported results are based on a single random seed for data splitting and model initialisation; repeating the pipeline across multiple seeds would strengthen confidence in the stability of the reported comparisons, and is left for future work.

2

Background

This chapter provides the conceptual background necessary to understand the problem, dataset, and modelling choices made in our thesis. We cover sound based fault diagnosis, the digital signal processing fundamentals underlying audio feature extraction, the acoustic feature representations used throughout the pipeline, the learning paradigms applied to the classification task, and the practical challenges that arise when working with real world industrial audio.

2.1 Sound Based Fault Diagnosis

Predictive maintenance aims to detect early signs of component degradation before an unexpected failure occurs. Traditional fault diagnosis systems rely on vibration, temperature, or oil based sensors, all of which require physical contact with the monitored component and dedicated installation effort. Acoustic signals offer a non contact alternative, capturing mechanical interactions, fluid flow, and structural vibrations through the sound a component naturally emits during operation [19]. Because these acoustic changes often begin well before a fault becomes severe enough to trigger a physical sensor threshold, sound based monitoring can, in principle, provide earlier warning than conventional condition monitoring approaches.

Machine learning has been applied extensively to acoustic fault diagnosis, allowing models to learn fault relevant patterns directly from data without relying on manually defined thresholds or expert crafted rules [2]. In automotive and heavy vehicle contexts specifically, acoustic methods have demonstrated early stage fault detection under realistic, uncontrolled operating conditions, which is a substantially harder setting than the clean, laboratory recorded signals used in many foundational studies.

In our work, the task is formulated as a seven class supervised classification problem. Each class corresponds to a specific combination of component, sound type, and condition, either healthy (OK) or faulty (NOK). This design choice produces a more actionable diagnostic output in practice, since it tells a maintenance engineer not just that something may be wrong, but which component is affected and what kind of acoustic symptom was observed, information that a binary healthy or faulty classifier cannot provide on its own. Table 2.1 summarises the acoustic characteristics of the four sound types studied in this thesis. Moan and Whine are both associated with the Power Steering Pump, but differ in their temporal character, a sustained periodic pulsation for Moan versus a tonal, harmonic whine concentrated in a higher

frequency band. Grunt and Hiss are both associated with the Steering Gear, but again differ substantially, Grunt being a short, impulsive transient event, while Hiss is a broadband, noise like sound that can extend across almost the entire audible spectrum. This spread in both frequency content and temporal character across the four sound types is central to why a single, narrow feature representation is unlikely to be sufficient, and it directly motivates the broad set of acoustic descriptors introduced later in this chapter.

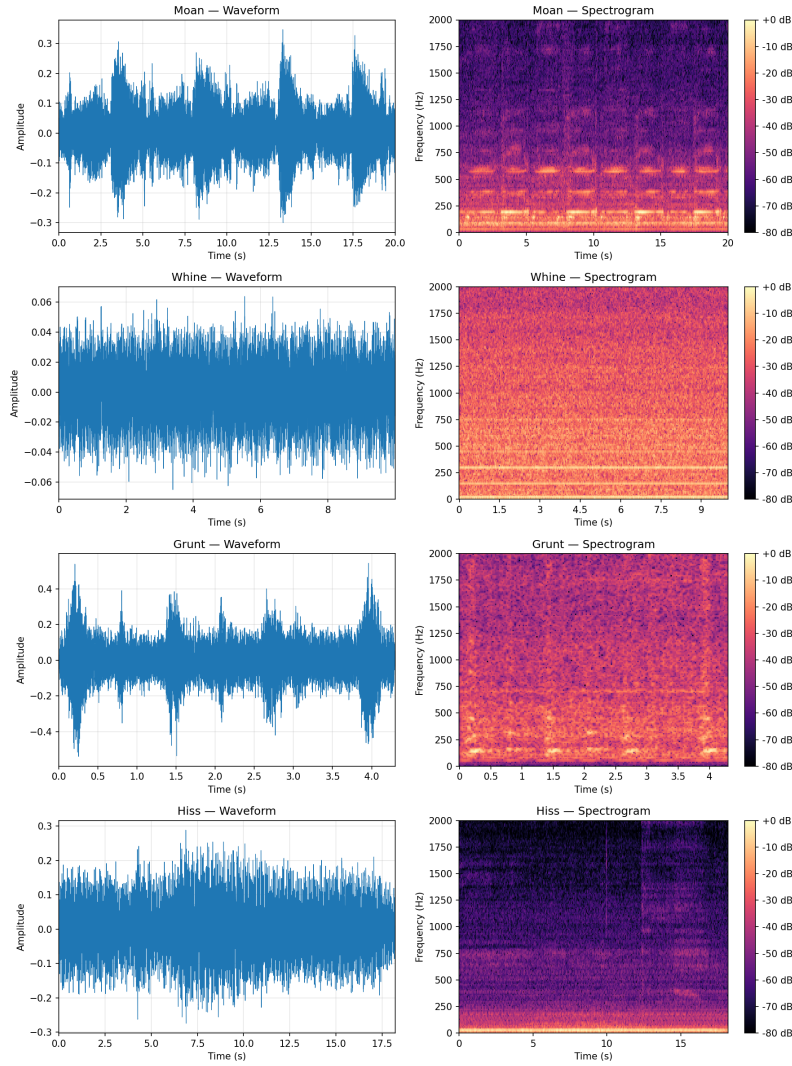


Figure 2.1: Representative waveform and spectrogram for each sound type: Moan, Whine, Grunt, and Hiss

Table 2.1: Acoustic characteristics of the four fault sound types.

Sound	Component	Character	Frequency range
Moan	PSP	Sustained, periodic	20–300 Hz
Whine	PSP	Tonal, harmonic	500–3000 Hz
Grunt	Steering Gear	Impulsive, transient	20– 600 Hz
Hiss	Steering Gear	Broadband, noise-like	400–20000 Hz

To illustrate the acoustic differences summarised in Table 2.1, Figure 2.1 shows one representative waveform and spectrogram from each of the four sound types. The sustained, periodic character of Moan and the tonal, narrow-band structure of Whine are visible as regular, repeating patterns in both the waveform and spectrogram, in contrast to the short, high-amplitude burst characteristic of Grunt and the diffuse, broadband energy spread of Hiss. Representative audio clips for each sound type are provided as supplementary material alongside this thesis.

2.2 Audio Signal Processing for Machine Sounds

Before any feature can be extracted, a physical sound wave must be converted into a digital representation that a computer can process. This section introduces the digital signal processing concepts that underpin every acoustic feature used later in our work.

2.2.1 Sampling and the Sampling Theorem

A microphone captures a continuous pressure wave, which must be sampled at discrete time intervals to be stored and processed digitally. The sampling theorem states that a signal can be perfectly reconstructed from its samples only if the sampling rate is at least twice the highest frequency component present in the signal, a limit known as the Nyquist frequency. Sampling below this rate causes aliasing, where high frequency content is misrepresented as spurious lower frequency content, permanently corrupting the recording. This has a direct practical consequence for this thesis, since Table 2.1 shows that the Hiss sound type carries energy up to roughly 24 kHz, meaning that a sampling rate of at least 48 kHz is required to represent it without aliasing. All recordings used with the classical hand crafted feature pipeline are therefore resampled to 48 kHz. Where a pretrained deep audio model is used instead, discussed in Section 2.4, audio is resampled to the fixed 32 kHz rate that the pretrained model was originally trained on [17], a constraint imposed by the pretrained model itself rather than a free design choice.

2.2.2 Windowing and the Short-Time Fourier Transform

Mechanical sounds are non stationary, meaning their frequency content changes over time with operating state, load, and speed. A single Fourier transform applied to an entire recording would only reveal which frequencies are present overall, discarding all information about when they occur, which is of limited use for a transient fault symptom such as Grunt. The Short-Time Fourier Transform (STFT) [20] addresses this by dividing the signal into short, overlapping frames, applying a window function to each frame to reduce spectral leakage at the frame edges, and computing a Fourier transform within each frame independently. The result is a spectrogram, a two dimensional representation of how acoustic energy is distributed across both time and frequency, which forms the basis for several of the acoustic descriptors introduced in the next section.

The choice of frame length in the STFT involves a fundamental trade off between time resolution and frequency resolution. A longer frame yields finer frequency resolution, since the frequency bins become more closely spaced, but coarser time resolution, since events are averaged over a longer window. A shorter frame gives the opposite trade off. In this thesis, a frame length of 4096 samples is used at a sampling rate of 48 kHz, giving a frequency resolution of approximately 11.7 Hz per bin, which is sufficient to resolve the fault relevant frequency components identified in Table 2.1 while retaining adequate temporal resolution for the transient Grunt sound type. The exact mathematical formulation of the STFT and the frame and hop length parameters used throughout this thesis are given in Chapter 3.

2.2.3 Segmentation of Recordings into Fixed Length Clips

Recordings in this dataset vary substantially in duration, while classifiers generally require fixed length input. To address this, a sliding window is applied to each recording, extracting fixed length clips with a fixed overlap between consecutive clips. Files shorter than the target clip length are handled by tiling and repeating the waveform until the target length is reached, rather than discarding them. This segmentation step has an important consequence for evaluation. Because consecutive clips from the same recording overlap and therefore share most of their underlying audio content, they cannot be treated as statistically independent observations. Any cross validation scheme that allows clips from the same source file to appear in both the training and validation folds risks leaking information between them and producing an overly optimistic performance estimate. All clips extracted from a given source recording are therefore grouped together and assigned to the same cross validation fold throughout this thesis, a point discussed further in Section 2.5 and detailed in Chapter 5.

2.3 Acoustic Feature Representations

Raw waveforms and spectrograms are high dimensional and not directly suitable as input to classical machine learning classifiers. Feature extraction reduces this high dimensional representation to a compact set of descriptive values that emphasise fault relevant characteristics of the signal while discarding much of the redundancy present in the raw audio. Our work compares seven categories of acoustic descriptor, described conceptually below. The precise mathematical definitions and formulas for each descriptor are given in Chapter 3.

MFCCs are derived by applying a discrete cosine transform to log-Mel filterbank energies [21], capturing the broad spectral envelope of a sound in a compact, decorrelated form that has proven effective for distinguishing different mechanical sound types.

Log-Mel spectrogram features retain the full band-by-band energy distribution of the Mel filterbank without the decorrelating DCT compression step used by MFCCs, preserving finer frequency band structure that is complementary to the more compact MFCC representation.

Sub-band energy ratios measure the fraction of total spectral energy contained within four physically motivated frequency bands spanning 20–1000 Hz. Because these bands are chosen to align directly with the known frequency characteristics of each sound type in Table 2.1, this descriptor is both interpretable and robust to overall amplitude variation between recordings.

Spectral descriptors, including spectral centroid, bandwidth, roll-off, flatness, and spectral contrast, quantify how frequency content is distributed across the spectrum and whether a sound is tonal or broadband in character.

Amplitude envelope kurtosis quantifies how impulsive a signal is, computed from the envelope of the signal obtained via the Hilbert transform. This descriptor is included specifically to help distinguish the sharply transient Grunt sound type from the more sustained Moan and Whine sound types.

DWT features, extracted using a Daubechies-6 wavelet, are able to capture both transient and periodic fault characteristics simultaneously, at multiple resolution levels, making them well suited to a dataset containing both impulsive and sustained fault symptoms [7].

Time-domain features, including RMS energy, zero crossing rate, peak amplitude, crest factor, skewness, and kurtosis, capture amplitude dynamics of the raw waveform that are sensitive to fault induced changes in signal shape, independent of any frequency domain transform.

2.4 Learning Paradigms for Acoustic Fault Detection

The acoustic descriptors introduced above are only useful once combined with an appropriate learning paradigm. Our thesis draws on three broad families of approach, described in turn below.

2.4.1 Classical Ensemble Learning and Stacked Generalisation

Classical ensemble methods combine multiple individually trained classifiers to produce a prediction that is generally more robust than that of any single model. Stacked generalisation extends this idea by training a meta learner on the out of fold prediction probabilities of several base classifiers, rather than combining their outputs through a fixed rule such as averaging or majority vote. This allows the meta learner to learn which base classifiers tend to be more reliable for which classes, potentially capturing complementary strengths across models with different inductive biases [9]. Training a large number of base classifiers, each with its own hyperparameters, is only practical with an efficient search strategy, which motivates the use of automated hyperparameter optimisation frameworks such as Optuna [11] in combination with stacked ensembles, an approach that has been shown to achieve strong performance even on comparatively small datasets.

2.4.2 Data Augmentation via Same-Class Mixup

When labelled training data is scarce, as is the case in this dataset, data augmentation can help a model generalise beyond the exact examples seen during training. Mixup is an augmentation technique that generates synthetic training samples by linearly interpolating between two real examples, encouraging the model to behave more smoothly between training points rather than memorising them individually [12]. In this thesis, mixup is restricted to pairs of samples from the same class, generating synthetic examples that increase within class coverage without introducing ambiguous, cross class interpolations that could blur the decision boundary between genuinely different fault types.

2.4.3 Convolutional Neural Networks and Transfer Learning via Pretrained Audio Models

Where classical models operate on a fixed set of hand crafted descriptors, convolutional neural networks (CNNs) instead learn their own feature representation directly from a spectrogram [2], through layers of learned convolutional filters. This removes the need to manually design features, at the cost of typically requiring more training data to learn a useful representation from scratch. Two convolutional approaches are considered here, the first trains a small CNN entirely from scratch on the available dataset. The second uses a pretrained audio neural network (PANN) [17], a convolutional architecture originally trained on AudioSet, a large scale, general purpose collection of labelled sound clips spanning several thousand hours of audio across hundreds of everyday sound categories. Rather than training such a large network from scratch on a dataset of only a few hundred recordings, which would be prone to severe overfitting, the pretrained PANN backbone is kept frozen and used purely as a fixed feature extractor, with only a small classifier head trained on top of the resulting embeddings for the truck fault detection task. This transfer learning approach tests whether general purpose acoustic representations, learned from a large and unrelated sound corpus, transfer usefully to a narrow, domain specific fault detection problem.

2.5 Challenges in Real World and Small Data Scenarios

Real world industrial audio presents challenges that are not fully addressed by controlled benchmark datasets such as MIMII [15]. Background noise varies significantly across recordings, reflecting differences in the vehicle, the recording setup, and the surrounding environment at the time each recording was made, and this variability cannot be fully controlled for after the fact. As discussed above, sliding window segmentation increases the total number of clips available for training, but does not increase the number of independent physical observations, since clips from the same recording remain correlated with one another. File level grouping in cross validation is therefore essential to obtain an honest estimate of how well a model generalises

to a truly unseen recording, rather than to an unseen window of an already seen recording [14].

Class imbalance is common in our datasets, since healthy operation is generally observed more often than any individual fault condition, and some fault types are inherently rarer than others in the available recordings. This makes accuracy an unreliable evaluation metric, since a classifier that performs well on the majority classes while performing poorly on rare fault classes can still achieve a misleadingly high accuracy score. The Matthews Correlation Coefficient is used throughout our work as the primary evaluation metric precisely because it remains informative under class imbalance [16].

Finally, the comparatively small size of this dataset, 326 recordings in total, limits how much can be asked of models that require large amounts of labelled data to train effectively from scratch. This motivates the inclusion of both a hand crafted feature and classical ensemble approach, which can perform well with limited data, and a transfer learning approach based on a pretrained audio model, which sidesteps the need for a large labelled dataset by reusing representations learned elsewhere. Comparing these approaches directly, alongside a small convolutional network trained from scratch, forms the central empirical contribution of this thesis.

3

Theory

This chapter presents the mathematical foundations of all signal processing, feature extraction, learning, augmentation, and evaluation components used in this thesis. Every formula corresponds directly to the implementation described in Chapter 5.

3.1 Short-Time Fourier Transform

Mechanical fault sounds have non-stationary frequency content that changes over time with operating state, load, and speed. The STFT divides the signal into overlapping frames and computes the Fourier Transform within each independently:

$$X(m, k) = \sum_{n=0}^{N-1} x[n + mH] w[n] e^{-j2\pi kn/N} \quad (3.1)$$

where N is the frame length, H is the hop size, m is the frame index, k the frequency bin, and $w[n]$ a Hann window that suppresses spectral leakage at frame boundaries. The power spectrogram is $P(m, k) = |X(m, k)|^2$. Each bin corresponds to $f_k = kf_s/N$.

Frame Count and Resolution

For the classical machine learning pipeline, each clip is fixed at $L = 96,000$ samples (2.0 s at $f_s = 48$ kHz, with 50% overlap between successive clips). The number of STFT frames within a clip is:

$$M = \left\lfloor \frac{L - N}{H} \right\rfloor + 1 \quad (3.2)$$

A frame length of $N = 4096$ samples is used, with a fixed hop size of $H = 1024$ samples. This gives $M = 90$ frames per clip, a frame duration of 85.3 ms, a hop duration of 21.3 ms, 75% overlap between adjacent frames, and a frequency resolution of $f_s/N \approx 11.72$ Hz per bin, summarised in Table 3.1.

Table 3.1: STFT parameters used, $f_s = 48$ kHz, $N = 4096$, $H = 1024$, $L = 96,000$ samples (2.0 s clip).

Frame Length	Duration (ms)	Hz/bin	Frames	Overlap
4096	85.3	11.72	90	75.0%

A frame length of 4096 samples was chosen as a fixed operating point, since it provides sufficient frequency resolution to separate the fault relevant bands identified in Table 2.1 while retaining adequate temporal resolution to capture the transient Steering Gear Grunt sound type. The hop size $H = 1024$ is fixed independently, ensuring that the temporal sampling density of the frame level statistics computed in Section 3.3 remains consistent throughout the pipeline.

3.2 Acoustic Feature Extraction

3.2.1 Mel-Frequency Cepstral Coefficients (MFCC)

3.2.1.1 Windowing and STFT:

The audio signal $x[n]$ has length $L = 96,000$ samples (2.0 s at 48 kHz). It is divided into overlapping frames of length $N = 4096$ samples with hop size $H = 1024$ samples, giving M frames as defined above. Each frame $m = 0, \dots, M - 1$ is multiplied by a Hann window to reduce spectral leakage:

$$w[n] = 0.5 \left(1 - \cos \frac{2\pi n}{N-1} \right), \quad n = 0, \dots, N-1. \quad (3.3)$$

The STFT of the m -th frame is

$$X(m, k) = \sum_{n=0}^{N-1} x[mH + n] w[n] e^{-j2\pi kn/N}, \quad (3.4)$$

where only the non redundant positive frequencies are retained:

$$k = 0, \dots, \left\lfloor \frac{N}{2} \right\rfloor. \quad (3.5)$$

The corresponding power spectrum is

$$P(m, k) = |X(m, k)|^2 \quad (3.6)$$

3.2.1.2 Mel Filterbank:

The power spectrum is passed through a bank of $F = 64$ overlapping triangular filters $H_i(k)$, spaced uniformly on the Mel scale. Librosa uses the Slaney mel scale by default, which applies a linear mapping below 1000 Hz and a logarithmic mapping above 1000 Hz:

$$\text{Mel}(f) = \begin{cases} \frac{f}{200/3} & f < 1000 \text{ Hz} \\ \text{Mel}(1000) + \frac{\ln(f/1000)}{\ln(6.4)/27} & f \geq 1000 \text{ Hz} \end{cases} \quad (3.7)$$

Each FFT bin corresponds to a frequency

$$f_k = \frac{k}{N} f_s, \quad (3.8)$$

where $f_s = 48,000$ Hz is the sampling rate. The energy in the i -th Mel band is

$$E(m, i) = \sum_{k=0}^{\lfloor N/2 \rfloor} P(m, k) H_i(k), \quad i = 0, \dots, 63. \quad (3.9)$$

3.2.1.3 Logarithmic Compression:

The Mel energies are converted to a decibel scale relative to the peak energy in the spectrogram:

$$E_{\log}(m, i) = 10 \log_{10} \left(\frac{\max(E(m, i), \varepsilon)}{\max(\mathbf{E}, \varepsilon)} \right), \quad \varepsilon = 10^{-10} \quad (3.10)$$

where $\mathbf{E}$ denotes the full Mel energy matrix and ε a floor threshold preventing numerical instability. The result is expressed in decibels relative to the clip's own peak, so the maximum value is always 0 dB, removing absolute amplitude differences between recordings.

3.2.1.4 Discrete Cosine Transform (DCT):

A type-II DCT decorrelates and compacts the log-Mel energies:

$$c_\ell(m) = \alpha_\ell \sum_{i=0}^{63} E_{\log}(m, i) \cos \left[\frac{\pi \ell}{64} \left(i + \frac{1}{2} \right) \right], \quad \ell = 0, \dots, 63, \quad (3.11)$$

with normalisation factors

$$\alpha_0 = \sqrt{\frac{1}{64}}, \quad \alpha_\ell = \sqrt{\frac{2}{64}} \text{ for } \ell > 0. \quad (3.12)$$

3.2.1.5 Coefficient Truncation:

Only the first $C = 20$ coefficients are retained:

$$\mathbf{c}_m = [c_0(m), c_1(m), \dots, c_{19}(m)]^\top \in \mathbb{R}^{20}. \quad (3.13)$$

c_0 represents overall log-energy, while higher coefficients capture progressively finer spectral detail.

3.2.2 Log-Mel Spectrogram Features

Log-Mel features follow the same pipeline as MFCC but omit the DCT, retaining all 64 Mel band energies per frame:

$$\mathbf{L}(m, i) = 10 \log_{10} \left(\frac{\max(E(m, i), \varepsilon)}{\max(\mathbf{E}, \varepsilon)} \right), \quad i = 0, \dots, 63 \quad (3.14)$$

Where MFCCs compress 64 bands into 20 decorrelated coefficients capturing broad spectral shape, log-Mel preserves fine band-level structure at the cost of higher inter-band redundancy, and is therefore complementary rather than duplicative.

3.2.3 Sub-band Energy Ratios

Sub-band energy ratios measure the fraction of total spectral energy in four physically motivated frequency bands, computed directly from the power spectrogram $P(m, k)$ across all frames and bins:

$$E_{\text{total}} = \sum_{m=1}^M \sum_k P(m, k) + \varepsilon, \quad \varepsilon = 10^{-10} \quad (3.15)$$

For each band b with bounds $[f_l^b, f_u^b]$:

$$E_b = \frac{\sum_{m=1}^M \sum_{k: f_l^b \leq f_k < f_u^b} P(m, k)}{E_{\text{total}}}, \quad b = 1, 2, 3, 4 \quad (3.16)$$

Table 3.2: Sub-band energy ratio bands and fault associations, $N = 4096$, $f_s = 48$ kHz.

Band	Range (Hz)	Bins	Associated sound
E_1	20–100	2–9	PSP Moan
E_2	100–300	9–26	Moan, Whine (low)
E_3	300–600	26–51	PSP Whine
E_4	600–1000	51–86	Steering Gear Hiss

3.2.4 Spectral Descriptors

Eleven spectral descriptors are computed per frame. Let $S(m, k) = |X(m, k)|$ denote the magnitude spectrogram frame and $\tilde{S}(m, k) = S(m, k) / \sum_j S(m, j)$ its column-normalised version.

3.2.4.1 Spectral Centroid:

$$C(m) = \sum_k f_k \cdot \tilde{S}(m, k) \quad (3.17)$$

3.2.4.2 Spectral Bandwidth:

$$BW(m) = \left(\sum_k \tilde{S}(m, k) \cdot |f_k - C(m)|^2 \right)^{1/2} \quad (3.18)$$

3.2.4.3 Spectral Roll-off

, the frequency f^* below which 85% of total frame energy is contained:

$$f^* = \min \left\{ f_k : \sum_{j \leq k} S(m, j) \geq 0.85 \sum_j S(m, j) \right\} \quad (3.19)$$

3.2.4.4 Spectral Flatness:

$$F(m) = \frac{\exp\left(\frac{1}{K} \sum_k \log\left(\max(S(m, k)^2, \varepsilon)\right)\right)}{\frac{1}{K} \sum_k \max(S(m, k)^2, \varepsilon)}, \quad \varepsilon = 10^{-10} \quad (3.20)$$

where $K = \lfloor N/2 \rfloor + 1$.

3.2.4.5 Spectral Contrast

, computed over 7 octave-spaced sub-bands using a 2% quantile for peak and valley estimation:

$$SC_s(m) = 10 \log_{10}(\mu_{\text{peak},s}(m)) - 10 \log_{10}(\mu_{\text{valley},s}(m)), \quad s = 1, \dots, 7 \quad (3.21)$$

Total: $1 + 1 + 1 + 1 + 7 = 11$ descriptors per frame, $\mathbf{D} \in \mathbb{R}^{11 \times M}$ per clip.

3.2.5 Amplitude Envelope Kurtosis

A clip-level scalar quantifying signal impulsiveness. The analytic signal is obtained via the Hilbert transform, $z(t) = x(t) + j\mathcal{H}\{x(t)\}$, with envelope $a(t) = |z(t)|$. Excess kurtosis (Fisher's definition):

$$\kappa = \frac{\frac{1}{T} \sum_{t=1}^T (a(t) - \mu_a)^4}{\sigma_a^4} - 3 \quad (3.22)$$

$\kappa \gg 0$ indicates infrequent large spikes (Grunt); $\kappa \approx 0$ indicates a Gaussian-like steady state signal (Moan, Whine, Hiss).

3.2.6 Discrete Wavelet Transform Features

The Daubechies-6 wavelet is compactly supported and has six vanishing moments, giving it enough smoothness to represent the sustained, periodic Moan and Whine sound types efficiently, while remaining short enough in time to localise the transient Grunt sound type, a balance that is well established in the mechanical fault diagnosis literature [7] and that motivated its selection here over other wavelet families.

The DWT decomposes the signal at multiple resolutions simultaneously:

$$a_j[n] = \sum_k a_{j-1}[k] \phi[2n - k], \quad d_j[n] = \sum_k a_{j-1}[k] \psi[2n - k] \quad (3.23)$$

The Daubechies-6 (db6) wavelet is used with six decomposition levels, producing seven sub-bands, one approximation (cA6) and six detail coefficients (cD6–cD1). For each sub-band $\mathbf{c}_j$ of length L_j :

$$E_j = \frac{\sum_n c_j[n]^2}{L_j + \varepsilon}, \quad \mu_j = \frac{1}{L_j} \sum_n c_j[n], \quad \sigma_j = \sqrt{\frac{1}{L_j} \sum_n (c_j[n] - \mu_j)^2} \quad (3.24)$$

yielding $7 \times 3 = 21$ DWT features per clip.

3.2.7 Time-Domain Features

Six frame level statistics, computed with `center=False` framing, $\mathbf{F} \in \mathbb{R}^{N \times M}$:

- **RMS energy:** $\text{RMS}(m) = \sqrt{\frac{1}{N} \sum_n F[n, m]^2}$
- **Zero-crossing rate:** $\text{ZCR}(m) = \frac{1}{N-1} \sum_n \mathbf{1}[F[n, m]F[n-1, m] < 0]$
- **Peak amplitude:** $P_{\text{peak}}(m) = \max_n |F[n, m]|$
- **Crest factor:** $CF(m) = P_{\text{peak}}(m) / (\text{RMS}(m) + \varepsilon)$, $\varepsilon = 10^{-8}$
- **Excess kurtosis:** $\kappa(m) = \frac{\frac{1}{N} \sum_n (F[n, m] - \mu_m)^4}{\sigma_m^4} - 3$
- **Skewness:** $\gamma(m) = \frac{\frac{1}{N} \sum_n (F[n, m] - \mu_m)^3}{\sigma_m^3}$

producing $\mathbf{T} \in \mathbb{R}^{6 \times M}$ per clip.

3.3 Temporal Aggregation

A bag-of-features representation is used [22], where each audio clip is described by a fixed length vector capturing overall statistical properties while discarding temporal ordering. Frame level feature matrices from MFCC, Log-Mel, spectral, and time-domain groups are aggregated by mean and standard deviation across M frames:

$$\bar{f}_i = \frac{1}{M} \sum_{m=1}^M F_{i,m}, \quad \sigma_{f_i} = \sqrt{\frac{1}{M} \sum_{m=1}^M (F_{i,m} - \bar{f}_i)^2} \quad (3.25)$$

Sub-band energy, DWT features, and envelope kurtosis are computed at the clip level and included directly without temporal aggregation.

Table 3.3: Feature group dimensions after aggregation.

Group	Frame dim	Aggregation	Final dim
MFCC	20	Mean + Std	40
Log-Mel	64	Mean + Std	128
Spectral descriptors	11	Mean + Std	22
Time-domain	6	Mean + Std	12
Sub-band energy	4	None (clip-level)	4
DWT	21	None (clip-level)	21
Envelope kurtosis	1	None (clip-level)	1
Total			228

This aggregation is applied to every clip extracted from the 326 source recordings, yielding 2886 clips in total, 2519 from the 283 training pool files and 367 from the 43 held-out test files, described further in Chapter 5.

3.4 ANOVA Based Feature Selection

The ANOVA F-statistic measures the ratio of between-class to within-class variance per feature [5]:

$$F_j = \frac{\sum_{c=1}^C n_c (\bar{x}_{j,c} - \bar{x}_j)^2 / (C - 1)}{\sum_{c=1}^C \sum_{i \in c} (x_{ij} - \bar{x}_{j,c})^2 / (N - C)} \quad (3.26)$$

The top $K = 180$ of the 228 features are retained. As a diagnostic on the full training pool, the highest-information groups are retained almost entirely (sub-band energy 100%, spectral descriptors 91%, MFCC 92%), while the largest and most redundant groups are pruned more heavily (Time-domain, which includes the clip-level envelope kurtosis descriptor, 85%; Log-Mel 73%; DWT 71%). Selection is performed inside each cross-validation fold on training data only, preventing leakage; selected indices therefore vary across folds as F-statistics are recomputed per partition.

3.5 Data Augmentation Strategies

Two distinct augmentation strategies are used in this thesis, depending on the model family, alongside a shared waveform-level augmentation applied upstream of feature or embedding extraction.

Waveform-Level Noise and Gain Augmentation

Before feature or embedding extraction, each training clip may be augmented at the raw waveform level, independently with probability 0.5 each, generating additional copies of every clip used only during training.

Additive noise is injected at a randomly sampled signal-to-noise ratio $\text{SNR} \sim U(20, 40)$ dB:

$$\sigma_{\text{noise}} = \sqrt{\frac{P_{\text{signal}}}{10^{\text{SNR}/10}}}, \quad \tilde{x}[n] = x[n] + \mathcal{N}(0, \sigma_{\text{noise}}^2) \quad (3.27)$$

where P_{signal} is the mean squared amplitude of the clip.

Random gain is applied at $g_{\text{dB}} \sim U(-6, +6)$ dB:

$$\tilde{x}[n] = x[n] \cdot 10^{g_{\text{dB}}/20} \quad (3.28)$$

SMOTE-Style Balancing with Same-File Neighbour Exclusion (Classical Models)

For the classical machine learning pipeline, class imbalance is addressed by generating synthetic feature vectors, following the Synthetic Minority Over-sampling Technique (SMOTE) technique [23], up to the majority class count within each cross-validation fold. Because clips overlap by 50%, two clips from the same source file are near-duplicates; standard SMOTE would frequently select such a near-duplicate as the nearest neighbour, producing a near-trivial synthetic sample. This is avoided by restricting the neighbour search to a different source file from the anchor. For each synthetic sample required for class c :

1. An anchor sample x_i is drawn at random from class c .
2. A neighbour pool is built from class c samples originating from a source file other than the anchor's own file.
3. The $k = 5$ nearest neighbours to x_i within this pool are found by Euclidean distance, and one, x_j , is selected at random among them.
4. A synthetic sample is generated by linear interpolation:

$$\tilde{x} = x_i + \lambda(x_j - x_i), \quad \lambda \sim U(0, 1) \quad (3.29)$$

If a class has samples from only one source file within a given fold, the method falls back to same-file interpolation, excluding only the anchor itself. Synthetic generation is applied after feature selection and scaling, so it operates entirely in the model's final input space.

Same-Class Mixup (Deep Learning Models)

For the convolutional models, both the pretrained PANN backbones and the custom CNN trained from scratch, mixup [12] is applied during training, on the frozen PANN embedding or the log-Mel spectrogram respectively. For a training sample x_i of class c , with probability 0.5, a second sample x_j of the same class is drawn (excluding x_i itself), and the two are combined as:

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j, \quad \lambda \sim \text{Beta}(0.4, 0.4) \quad (3.30)$$

Because both samples belong to the same class c , the label remains c and no soft-label interpolation is required. The Beta(0.4, 0.4) distribution is U-shaped, concentrating mass near 0 and 1, so synthetic samples remain acoustically close to one of the two originals rather than sitting exactly at their midpoint. Restricting interpolation to the same class avoids the ambiguous cross-class blends that would result from mixing acoustically distinct fault states.

3.6 Convolutional Neural Networks and Transfer Learning

Convolutional Feature Learning

Where the classical pipeline relies on the hand crafted descriptors above, the deep learning models instead learn their own feature representation directly from a log-Mel spectrogram, through stacked two-dimensional convolutional layers. Each convolutional block applies a learned filter bank, followed by batch normalisation, a ReLU non-linearity, and spatial pooling, progressively reducing the time-frequency resolution of the representation while increasing its channel depth. The custom CNN trained from scratch is built from a small number of such blocks followed by global average pooling and a linear classification layer, learning its entire representation directly from the 326-recording dataset.

Pretrained Audio Neural Networks (PANN)

Pretrained Audio Neural Networks (PANNs) [17] are convolutional architectures originally trained on AudioSet, a large-scale, general purpose collection of labelled sound clips spanning several thousand hours of audio across 527 everyday sound categories. Rather than training a large convolutional network from scratch on a dataset of only a few hundred recordings, which would be highly prone to overfitting, we use PANN backbones as frozen feature extractors, the pretrained convolutional layers are kept fixed, and each input clip is mapped to a fixed dimensional embedding vector through global temporal pooling, combining the maximum and mean activation over time. A small, trainable classifier head is then trained on top of this embedding for the seven class fault detection task. Three PANN variants of increasing depth and capacity are compared in this thesis, differing in convolutional depth, filter size, and resulting embedding dimensionality.

3.7 Stacked Generalisation

Stacking combines base classifiers by training a meta learner on their out-of-fold (OOF) probability outputs [9]. For each fold f , base classifier b is trained on split $\mathcal{T}_f$ and produces probabilities on $\mathcal{V}_f$. After all folds, out-of-fold predictions cover all $N = 2519$ training pool clips exactly once and are stacked into:

$$\mathbf{Z} \in \mathbb{R}^{N \times (B \cdot C)} = \mathbb{R}^{2519 \times 42} \quad (3.31)$$

where $B = 6$ base classifiers each produce a $C = 7$ dimensional probability vector per clip.

A Logistic Regression meta learner with $C_{\text{meta}} = 0.1$ and `class_weight='balanced'` is trained on $\mathbf{Z}$. LinearSVM does not natively produce calibrated probabilities and is wrapped in `CalibratedClassifierCV` using Platt scaling with 3-fold internal cross validation. At inference, each new clip is passed through all six base classifiers, and their concatenated 42 dimensional probability vector is classified by the meta learner.

3.8 Hyperparameter Optimisation

Optuna employs the Tree-structured Parzen Estimator (TPE), a Bayesian optimisation algorithm [11] that models promising regions of the hyperparameter space using probability distributions and samples new configurations from these regions. This enables an efficient search by balancing exploration of new regions against exploitation of known high performing configurations, and is generally more sample-efficient than grid or random search. For each of the top three base models selected by Round 1 out-of-fold MCC, ten optimisation trials are conducted, each evaluated using the same five-fold cross validation protocol on the training data only.

3.9 Evaluation Metrics

MCC is used as the primary evaluation metric [16], as it accounts for all entries of the confusion matrix and remains informative under class imbalance. For binary classification:

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (3.32)$$

For multiclass problems, MCC is generalised using the full confusion matrix, ranging from -1 (complete disagreement) through 0 (random performance) to $+1$ (perfect classification).

Macro F1-score is the secondary metric, averaging per class F1 without class frequency weighting:

$$\text{F1}_{\text{macro}} = \frac{1}{C} \sum_{c=1}^C \frac{2 \cdot P_c \cdot R_c}{P_c + R_c} \quad (3.33)$$

where P_c and R_c are precision and recall for class c .

StratifiedGroupKFold with file-level grouping is used as the single cross validation method, preventing leakage between overlapping clips from the same recording while preserving class balance across folds [14].

File-level majority voting. Because each recording is split into multiple overlapping clips, a model produces one prediction per clip rather than per file. To obtain a file-level prediction, clip level predictions from the same source file are aggregated by majority vote:

$$\hat{y}_{\text{file}} = \text{mode}_{i \in \text{file}}(\hat{y}_i) \quad (3.34)$$

An alternative, soft voting variant instead averages the predicted class probabilities across all clips from a file before taking the arg max:

$$\hat{y}_{\text{file}} = \arg \max_c \left(\frac{1}{n_{\text{file}}} \sum_{i \in \text{file}} p_i(c) \right) \quad (3.35)$$

File-level aggregation is applied consistently across all model families when reporting cross-validation and held-out test results in Chapter 6, since clip-level metrics alone over-represent files with many clips and do not reflect the file-level decision a deployed system would ultimately need to make.

SHAP. To interpret individual model predictions and identify which features drive them, SHAP (SHapley Additive exPlanations) [18] attributes each feature an importance value for a given prediction, based on cooperative game theory, by considering the marginal contribution of that feature across all possible subsets, or coalitions, of the other features. For tree based models such as Random Forest, this can be computed exactly and efficiently via `TreeExplainer`, used in Chapter 6 to identify globally and per-class important features.

4

Related Work

This chapter reviews prior research directly relevant to the methods and design choices made in our thesis, covering acoustic fault diagnosis, feature extraction and selection, data augmentation, classical ensemble learning, and evaluation methodology under realistic operating conditions.

Broadly, they fall into three groups that build on one another. A first group establishes that acoustic signals are a viable, non-contact alternative to conventional condition monitoring, and that classical machine learning on carefully designed features is a credible baseline in this setting. A second group develops the specific feature representations, selection methods, and augmentation strategies this thesis draws on. A third group addresses how such systems should be trained and evaluated, particularly under class imbalance and correlated, sliding window segmented data. The remainder of this chapter follows this structure.

Lei et al. provided a comprehensive review of machine learning methods for fault diagnosis, consistently identifying feature representation, noise robustness, and data scarcity as the central challenges across industrial settings [2]. Their review spans a wide range of machine types and sensing modalities, and notably concludes that no single feature representation or model family dominates across domains, with the best choice depending heavily on dataset size, noise conditions, and the nature of the signature itself. This conclusion motivates the handcrafted feature approach adopted, where domain knowledge about the acoustic character of each type, summarised earlier in Table 2.1, is used to guide feature design, rather than relying on a large labelled dataset for automatic representation learning, which the available 326 recordings would be unlikely to support well on their own.

Nasim et al. demonstrated that acoustic signals reliably detect vehicle faults at an early stage using classical machine learning, even under realistic operating noise, evaluating several standard classifiers on recordings collected directly from vehicles rather than under laboratory conditions [3]. Their central finding, that handcrafted features combined with standard classifiers provide a viable diagnostic baseline when labelled data is limited, directly supports the classical modelling approach, and is one of the empirical questions our thesis re-examines directly against more recent deep learning alternatives through its first research question.

The MIMII dataset [15] is widely used for acoustic anomaly detection evaluation, and has become something of a de facto benchmark for the field, consisting of record-

ings from industrial machines under both normal and induced fault conditions in a controlled setting. However, its controlled recording conditions and binary normal or anomaly formulation differ substantially from the multi class, real world operational setting, where recordings vary in duration, background noise, and recording environment, and where the goal is not merely to detect that a fault exists, but to identify which of several distinct fault symptoms is present. This motivates evaluating models directly on operationally collected vehicle recordings, rather than relying solely on results reported against a controlled benchmark.

4.1 Feature Extraction for Sound Analysis

In this section, we discussed how each work contribute a different piece of the feature representation used in this thesis, which is a general motivation for combining multiple feature groups, a specific rationale for wavelet-based descriptors, and empirical guidance on segmentation duration and window length, all of which directly shaped the design decisions detailed in Chapter 3.

Zaheer et al. surveyed AI based acoustic source identification across a range of application domains and demonstrated that combining multiple complementary feature groups, rather than relying on any single representation, consistently improves classification performance [4]. Their survey covers spectral, cepstral, and temporal descriptors broadly and highlights that different descriptor types tend to capture different, only partially overlapping, aspects of a sound’s character. This directly motivated the seven group, 228 dimensional feature design used in this thesis, where each group, MFCC, log-Mel, sub-band energy, spectral shape, time-domain statistics, envelope kurtosis, and DWT coefficients, is intended to capture a distinct and complementary aspect of the acoustic signal, as detailed in Chapter 3.

Yan et al. reviewed wavelet transform applications in rotary machine fault diagnosis, showing across a range of studies that multi resolution decomposition effectively isolates fault relevant frequency content that can be difficult to resolve with a single, fixed time-frequency resolution such as a standard STFT [7]. Their review particularly emphasises the strength of wavelet methods for transient, impulsive fault signatures, which are poorly localised in a conventional spectrogram. This motivated the inclusion of a dedicated wavelet descriptor group in this thesis, the Daubechies-6 wavelet with six decomposition levels is used, providing 21 features intended to capture both the transient Steering Gear Grunt symptom and the more sustained, periodic Moan and Whine symptoms within a single descriptor family.

Kshirasagar et al. investigated the effect of sliding window and segmentation parameters on acoustic anomaly detection performance in an automotive environment, using a recurrent spiking neural network architecture, and showed that segmentation choices have a measurable and non-trivial impact on downstream classification accuracy [13]. Their broader finding, that segmentation strategy is not a neutral preprocessing step but an active design choice with real performance consequences, supports the careful, fixed 2.0 second sliding window with 50% overlap used, and the corresponding file level grouping applied during cross validation to prevent

information from overlapping windows leaking between training and validation splits.

Badawy et al. used multi-second audio clips for vehicle fault and emergency sound classification and demonstrated that clip durations in this range capture sufficient temporal context to reliably distinguish different mechanical fault sounds, while remaining short enough to keep the number of clips per recording, and therefore the overall dataset size, practical for training [8]. This supports the clip duration used in this thesis, chosen to be long enough to contain multiple STFT frames and capture the periodic structure of sustained fault sounds, while remaining short enough that even the shortest recordings in the dataset yield a usable number of clips.

4.2 Feature Selection

Jamil and Khanam systematically evaluated ANOVA F-test and Kruskal-Wallis feature ranking for bearing fault diagnosis across several classifiers, showing that ANOVA based selection consistently improves classifier performance by retaining discriminative features while removing redundant or noisy ones [5]. Notably, their comparison found that the choice of ranking method mattered less than the act of performing feature selection at all, with both ANOVA and Kruskal-Wallis yielding similar improvements over using the full feature set. This supports the use of ANOVA based `SelectKBest` with $K = 180$ in this thesis, applied independently within each cross validation fold to avoid leaking information about the validation split into the feature selection step, as described in Section 3.3.

4.3 Data Augmentation

Zhang et al. introduced mixup, a data augmentation technique that generates synthetic training samples through convex combinations of two real examples, and showed improved generalisation across image, speech, and tabular classification tasks compared to training on the original data alone [12]. Their original formulation mixes examples across classes, using the mixing coefficient to also interpolate the labels, which encourages smooth, linear behaviour of the classifier between training points of different classes. In this thesis, mixup is restricted to pairs of samples from the same class, and is used for the two deep learning pipelines, applied in embedding space for the PANN models and in spectrogram space for the CustomCNN models, as detailed in Chapter 3. This same-class restriction avoids generating ambiguous synthetic samples near class boundaries that cross class mixing would introduce in a seven class fault setting, where several classes are acoustically close, for example the two Power Steering Pump sound types, and where an artificially blended sample could plausibly resemble neither original fault symptom well. For the classical machine learning pipeline, a SMOTE based synthetic oversampling strategy [23] is used instead, re-implemented manually in feature space with same-class, cross-file neighbour interpolation to avoid generating near-duplicate synthetic samples from overlapping sliding windows of the same recording. Waveform level noise and gain perturbation is applied separately, during training for the two deep learning pipelines

only, increasing robustness to the natural amplitude and background noise variation present in real operational recordings.

4.4 Base Classifier Selection

Six base classifiers from four model families, tree based, gradient boosting, linear, and neural network, are selected to provide complementary coverage of the hypothesis space for the classical machine learning comparison.

Random Forest [24] trains an ensemble of decision trees on bootstrap samples with random feature subsets at each split. This double randomisation, over both samples and features, reduces variance and provides robustness to irrelevant or noisy features in the 180-dimensional selected feature space.

Extra Trees [25] additionally randomises split thresholds, rather than searching for the optimal threshold at each split, further reducing variance and training time compared to Random Forest. Although Extra Trees operates on the same feature space as Random Forest, its different split mechanism produces systematically different individual tree errors, which is useful for building a diverse base classifier ensemble for the later stacking stage. The same four hyperparameters and search ranges as Random Forest are tuned, listed in Table 4.1.

Extreme Gradient Boosting (XGBoost) [10] builds an additive ensemble via gradient boosting with L_1 and L_2 regularisation, where each new tree is trained to correct the residual errors of the ensemble built so far, allowing it to model complex, non-linear feature interactions. Its sequential, residual-correcting training procedure is complementary to the parallel, variance-reducing bagging strategy used by Random Forest and Extra Trees.

Light Gradient-Boosting Machine (LightGBM) [26] is also a gradient boosting method, but uses leaf-wise tree growth combined with Gradient-based One-Side Sampling, which focuses each new tree’s training on the examples the ensemble currently finds hardest to classify correctly. Despite both being gradient boosting methods, LightGBM’s different tree growth strategy produces different split decisions from XGBoost in practice, which is again useful for base classifier diversity within the stacking framework.

Linear Support Vector Classifier (LinearSVM) [27] finds the maximum margin linear decision boundary between classes, with L_2 regularisation controlling the trade-off between margin size and training error. It is particularly well matched to this problem because ANOVA F-test selection retains features based on their linear separability between classes by construction, producing a feature space that already favours a linear decision boundary. Beyond its classification performance, LinearSVM’s minimal parameter count also gives it a very small model footprint and fast inference, both practically relevant properties for deployment in a resource-constrained industrial setting, discussed further in Chapter 6.

Multilayer Perceptron (MLP) is the only base classifier that learns feature interactions through a continuous, differentiable non-linear transformation of the

input space, rather than through the discrete, axis-aligned splits used by the tree based and gradient boosting classifiers above. Its hidden layers can, in principle, approximate interaction structures that are costly for an axis-aligned tree ensemble to represent exactly, complementing rather than duplicating the interaction modelling already provided by RF, ET, XGBoost, and LightGBM. The architecture follows a funnel design, $\text{input}(180) \rightarrow \text{h1} \rightarrow \text{h2} \rightarrow \text{output}(7)$, progressively compressing the representation before the final classification layer.

Table 4.1: Hyperparameter search spaces for each base classifier

Model	Parameter	Search Range
RF / ET	n_estimators	[100, 500]
	max_depth	[5, 30]
	min_samples_split	[2, 10]
	min_samples_leaf	[1, 5]
XGBoost	n_estimators	[100, 500]
	max_depth	[3, 15]
	learning_rate	[0.01, 0.3]
	subsample	[0.6, 1.0]
	colsample_bytree	[0.6, 1.0]
	reg_alpha	[0.001, 10]
	reg_lambda	[0.001, 10]
LightGBM	n_estimators	[100, 500]
	num_leaves	[15, 63]
	learning_rate	[0.01, 0.3]
	subsample	[0.6, 1.0]
	colsample_bytree	[0.6, 1.0]
	reg_alpha	[0.001, 10]
	reg_lambda	[0.001, 10]
LinearSVM	C	[0.01, 100]
MLP	h1	[64, 256]
	h2	[32, 128]
	learning_rate_init	$[10^{-4}, 10^{-2}]$
	alpha	$[10^{-5}, 10^{-1}]$

4.5 Stacked Generalisation

Wolpert originally introduced stacked generalisation as a method for combining multiple learned models more effectively than a fixed combination rule such as

majority voting or averaging, by training a second-level model to learn how to combine the first-level models' outputs from data [9]. This general principle, that a learned meta-level combiner can outperform a fixed combination rule when base models have complementary strengths and weaknesses, is the theoretical basis for the stacking framework adopted in our work, described in detail in Chapter 3.

In this thesis tuning the base learners with Optuna before combining them through a trained meta-learner, closely mirrors the five-round pipeline used in this thesis, in which base classifiers are first tuned individually and then combined through a Logistic Regression meta-learner, targeting the same combination of class imbalance and limited sample size present in the dataset used here.

4.6 Hyperparameter Optimisation and Evaluation

This final section address a common concern, distinct from feature or model design on how the experimental pipeline itself, its tuning strategy, its evaluation metric, and its cross validation procedure, should be constructed so that reported results are trustworthy and not artefacts of the evaluation methodology chosen.

Akiba et al. introduced Optuna and showed that its Tree-structured Parzen Estimator sampling strategy finds good hyperparameter configurations in fewer trials than exhaustive grid search or purely random search, by adaptively concentrating the search on promising regions of the hyperparameter space based on previous trials [11]. This sample efficiency is particularly valuable in our thesis, where the top three of the six base classifiers (selected by Round 1 baseline out-of-fold MCC) are tuned with Optuna, as described in Chapter 3.

Chicco and Jurman demonstrated, across a wide range of binary classification scenarios, that the Matthews Correlation Coefficient is substantially more informative than accuracy or F1 score when class distributions are imbalanced, since MCC accounts for all four cells of the confusion matrix simultaneously rather than only the correctly classified examples [16]. Their argument extends naturally to the multi-class, imbalanced setting of this thesis, motivating the use of MCC as the primary evaluation metric throughout.

Tougui et al. showed that the choice of cross validation strategy has a measurable and sometimes substantial effect on reported diagnostic performance, with naive random splitting tending to produce optimistic performance estimates when the underlying data contains hidden dependencies between samples [14]. This finding directly reinforces the importance of file level grouped cross validation when working with sliding window segmented audio, since clips from the same recording are exactly the kind of hidden, correlated samples their work warns against splitting naively, and is the basis for the `StratifiedGroupKFold` protocol used consistently.

5

Methods

This chapter describes the complete experimental pipeline used in this thesis, from raw audio recordings to final model evaluation, covering the classical machine learning pipeline in detail and the deep learning pipelines used for comparison.

5.1 Dataset

The dataset consists of 326 WAV recordings collected from trucks under operational conditions. Each file is labelled with a component (Power Steering Pump (PSP) or Steering Gear), a sound type (Moan, Whine, Grunt, or Hiss), and a condition (OK or NOK), producing seven classes in total. No Steering_Gear_Grunt_OK class exists, since Grunt is treated as an inherent fault indicator that does not occur under normal operation.

Of the 326 recordings, the large majority were natively recorded at 48 kHz, with a small number captured at 44.1 kHz and 22.05 kHz. All recordings are resampled to a uniform target rate of 48 kHz before any further processing, for the classical machine learning pipeline described in this chapter. This is necessary because feature extraction parameters such as frame length and frequency bin spacing correspond to physical frequencies only when computed at a consistent sample rate across all files. Since the majority of recordings are already natively at 48 kHz, this choice minimises the amount of signal modification required overall. Table 5.1 summarises the full dataset statistics.

Table 5.1: Dataset summary statistics

Total recordings	326
Target sample rate	48 kHz
Duration min	0.5 s
Duration max	46.0 s
Duration mean	9.2 s

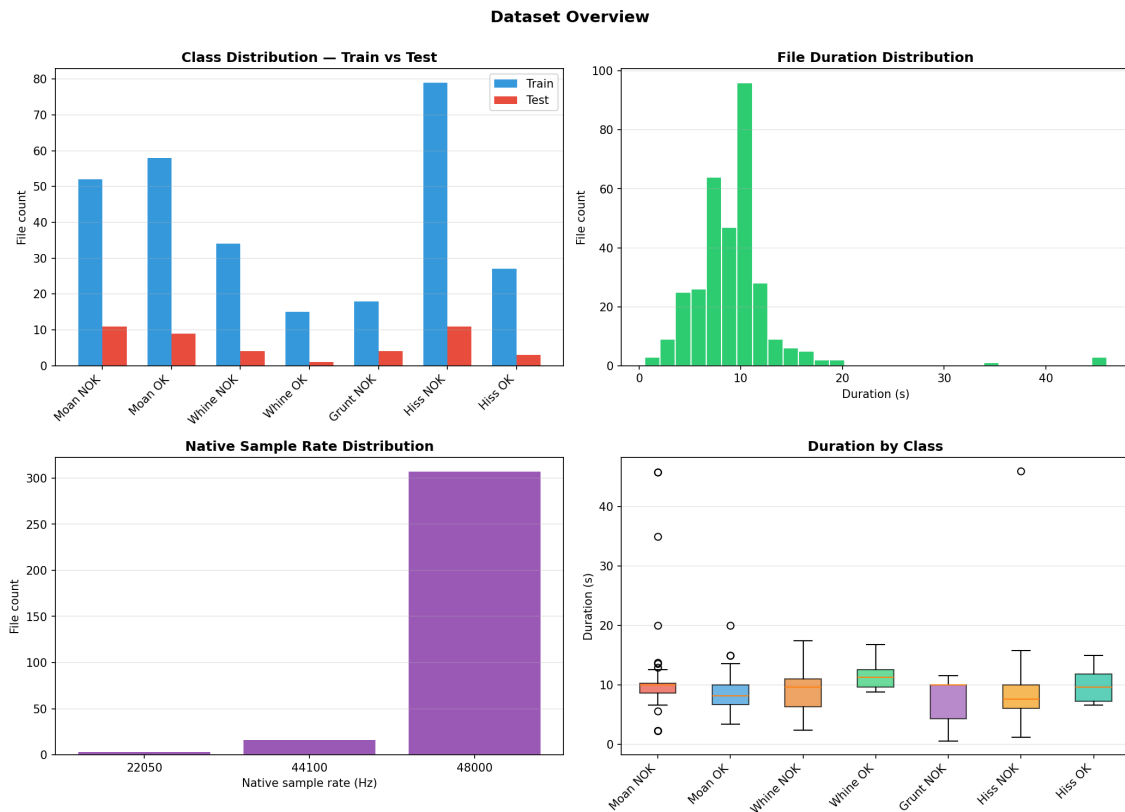


Figure 5.1: Dataset overview

Class distribution across the train and test splits, file duration distribution, native sample rate distribution, and duration by class. The duration histogram confirms that most recordings fall between roughly 5 and 15 seconds, with a small number of outliers up to 46 seconds, motivating the sliding window approach described in Section 5.2. The class distribution shows a clear imbalance, motivating the use of MCC as the primary evaluation metric and class weighted training for all models.

Label Parsing

Every recording in the dataset must be reduced to one of the seven classes before it can be used for training. Since the dataset was assembled from field recordings rather than a purpose built labelling tool, labels are parsed directly from WAV filenames using the following priority rules:

1. Files containing **idle** or **normal** are mapped to PSP_Moan_OK, as idle Power Steering Pump operation is acoustically equivalent to a normal Moan condition.
2. Sound type is identified by the keywords **moan**, **whine**, **grunt**, or **hiss** appearing in the filename.
3. Component is identified by the keywords **psp**, **steering**, or **stearing**, the latter accounting for a recurring spelling variant present in the raw filenames. Files containing **cop** are mapped to PSP for Moan or Whine sound types, and to Steering Gear for Grunt or Hiss sound types.

4. Condition is resolved from explicit `ok` or `nok` markers in the filename, together with a small set of confirmed filename conventions, including files marked `ok_or_nok` and a confirmed `not` typo for `nok`, both of which are treated as NOK under a conservative diagnostic interpretation. Files where the condition cannot be resolved unambiguously from the filename are excluded from the dataset rather than guessed, and are logged separately as ambiguous.

5.2 Preprocessing

All recordings are loaded and resampled to 48 kHz mono using `librosa.load`, and the DC offset is removed by subtracting the mean of the waveform.

A sliding window of 2.0 s (96,000 samples at 48 kHz) with 50% overlap (hop of 1.0 s, 48,000 samples) is then applied to each normalised recording. Three cases are handled explicitly: recordings shorter than 2.0 s are tile repeated to produce exactly one clip of the target length; recordings of at least 2.0 s produce $\lfloor (L - N)/H \rfloor + 1$ full clips, where L is the recording length in samples, $N = 96,000$, and $H = 48,000$; and any remaining tail shorter than a full window is zero padded to the target length and kept as one additional clip, provided it contains at least 10% real signal, to avoid retaining clips that are almost entirely padding. All clips extracted from the same source file are assigned the same label and the same cross validation fold, preventing information from overlapping windows leaking between training and validation splits. Across the full dataset, this segmentation produces 2519 clips from the 283 files in the training pool and 367 clips from the 43 held out test files.

5.3 Feature Extraction

Seven feature groups are extracted per clip using a fixed STFT frame length of 4096 samples and hop length of 1024 samples, both at 48 kHz, as motivated in Section 3.1. Table 5.2 summarises each group; the full mathematical definitions are given in Chapter 3. Any NaN or infinite values produced during extraction, for example from a near-silent clip, are replaced with zero before further processing.

Table 5.2: Feature extraction summary

Group	Key parameters	Frame dim	Final dim
MFCC	20 coeff., 64 Mel bands	20	40
Log-Mel	64 bands, power-to-dB	64	128
Sub-band energy	4 bands: 20–1000 Hz	—	4
Spectral	centroid, BW, rolloff, flatness, contrast (7 bands)	11	22
Envelope kurtosis	Hilbert transform	—	1
DWT	db6, 6 levels, 7 sub-bands	—	21
Time-domain	RMS, ZCR, peak, crest, skewness, kurtosis	6	12
Total			228

5.4 Feature Selection

ANOVA Feature- test selection (Anova F) retains the $K = 180$ most discriminative features out of the full 228 using `SelectKBest` with `f_classif`, applied independently inside each cross validation fold on that fold’s training data only, to avoid leaking information about the validation split into the selection step. Selected feature indices therefore vary slightly across folds, as F-statistics are recomputed on each training partition. Figure 5.2 shows, as a diagnostic on the full training pool, what fraction of each feature group survives selection at $K = 180$: The sub-band energy and spectral descriptor groups are retained almost entirely, whereas the larger Log-Mel and DWT groups undergo greater reduction. The right panel further shows the 20 individual features with the highest ANOVA F-scores across the full training pool.

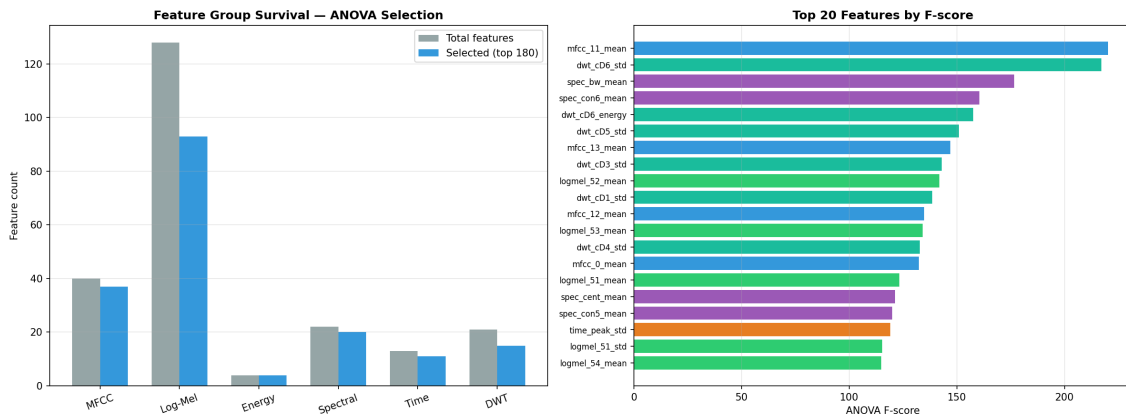


Figure 5.2: ANOVA Feature Selection

Left: number of features in each group before and after ANOVA selection at $K = 180$, showing the number of features retained from each feature group. Right: the top 20 individual features ranked by ANOVA F-score across the full training pool, coloured by feature group.

5.5 Data Augmentation

Two complementary augmentation strategies are applied during training, described in full in Chapter 3.

At the waveform level, before feature extraction, each training clip may independently receive additive Gaussian noise at a randomly sampled SNR between 20 and 40 dB, and a random gain shift between -6 and $+6$ dB, each applied with probability 0.5, generating two augmented copies per real training clip.

At the feature level, after ANOVA selection and scaling, class imbalance is addressed through a SMOTE style [23] synthetic oversampling procedure with an explicit same-file neighbour exclusion: since overlapping clips from the same recording are near-duplicates in feature space, the nearest neighbour search used to generate a synthetic sample is restricted to real samples of the same class originating from a different source file than the anchor sample, falling back to same-file interpolation

only when a class has examples from a single source file within a given fold. This procedure balances every class up to the majority class count within each training fold, and is applied strictly after feature selection and scaling, and strictly to the training partition of each fold, so that no synthetic sample is ever included in a validation split or the held out test set.

This ANOVA based selection step is applied only within the classical machine learning pipeline, which operates on a fixed, handcrafted feature vector, for which explicit feature selection is a natural and standard preprocessing step to reduce dimensionality and remove redundant features before training. The convolutional models, by contrast, are designed specifically to learn their own feature representation directly from a log-Mel spectrogram during training, through their convolutional layers, rather than from a fixed, pre selected input vector. Feeding the CNN a reduced, ANOVA selected feature vector instead of the full spectrogram would defeat the purpose of using a convolutional architecture at all, since the value of a CNN lies precisely in learning which time-frequency regions are informative during training, rather than having that decision made for it in advance by a separate, feature vector based statistical test. This distinction is part of why the classical and deep learning pipelines in this thesis operate on different input representations, described respectively in Section 3.3.

5.6 Classical Model Training

Model training proceeds through five rounds, each building on the result of the previous one.

Round 1: Default Base Classifiers

All six base classifiers, Random Forest, Extra Trees, XGBoost, LightGBM, LinearSVM, and MLP, described in Chapter 4, are trained with default hyperparameters under five fold **StratifiedGroupKFold**. Within each fold, the sequence of steps is, (1) ANOVA feature selection fitted on the training split only, (2) **StandardScaler** fitted on the training split, applied only to LinearSVM and MLP, which require feature scaling, while the tree based and boosting models do not, (3) SMOTE style augmentation applied to the scaled training split only, (4) the model is trained with class weights set inversely proportional to class frequency, where supported natively, or via explicit sample weighting for XGBoost, which does not expose a native class weighting option, (5) predicted probabilities are collected on the held out validation split for that fold. After all five folds, the out-of-fold (OOF) predictions across all training pool clips are pooled and used to compute an overall MCC and macro F1 score per model, and the six models are ranked by out-of-fold MCC. The top three models by this ranking proceed to Round 2 for hyperparameter tuning, while the remaining three retain their default parameters for all subsequent rounds.

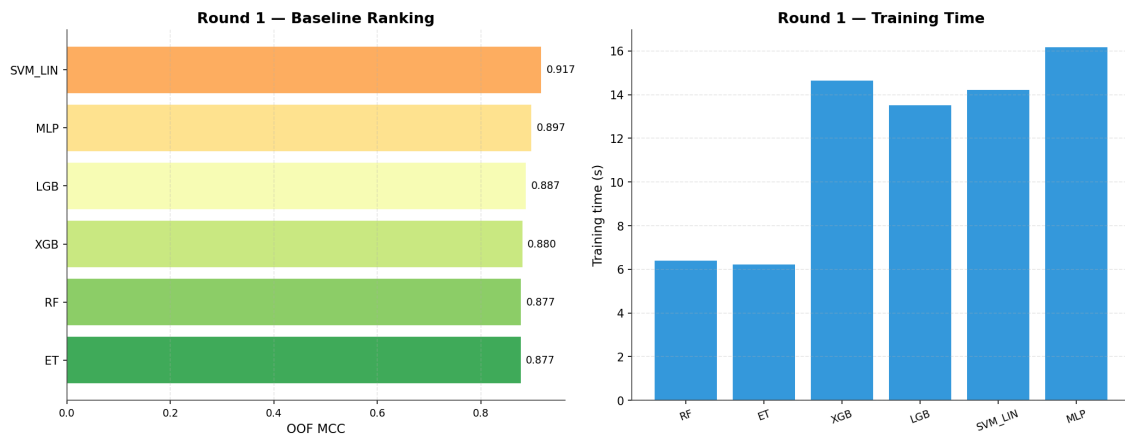


Figure 5.3: Round 1 Baseline results
Out-of-fold MCC ranking of all six default base classifiers (left), and their corresponding training time (right).

Round 2: Optuna Hyperparameter Tuning

Optuna’s search strategy is based on the Tree-structured Parzen Estimator (TPE), a sequential, Bayesian model based optimisation algorithm. Rather than exhaustively evaluating a fixed grid of hyperparameter combinations, or sampling combinations purely at random, TPE builds two probability distributions from the trials already evaluated, one over hyperparameter configurations that produced good results, and one over configurations that produced poor results, and proposes new configurations that are likely under the first distribution and unlikely under the second. This allows the search to concentrate progressively on promising regions of the hyperparameter space as more trials are completed, rather than spending an equal search budget everywhere, making it considerably more sample efficient than grid or random search for a fixed number of trials [11].

The top three models identified in Round 1, LGB, SVM_LIN, and MLP, are tuned using Optuna, run for 10 trials per model. In each trial, Optuna’s TPE sampler proposes a new hyperparameter configuration within the search ranges given in Table 4.1, the full five fold cross validation pipeline identical to Round 1 is executed with that configuration, and the resulting mean out-of-fold MCC is recorded as the trial’s objective score. After all trials, the configuration achieving the highest out-of-fold MCC is selected for each of the three tuned models. Round 2 therefore does not change the training procedure itself, only the hyperparameter values used within it, and only for the three models that already showed the most promise under default settings in Round 1. Tuned parameters are carried forward into every subsequent round.

Round 3: Out-of-Fold Probability Matrix

With tuned parameters now available for the top three models, all six base classifiers are retrained under the same five fold `StratifiedGroupKFold` pipeline used in Round 1, with the top three using their Optuna tuned parameters and the remaining

three retaining their Round 1 defaults. The purpose of this round is not to re-rank the individual models, but to collect the out-of-fold probability output of every base classifier for every training pool clip, to serve as input to the meta-learner. Each base classifier b produces an out-of-fold probability matrix $\hat{P}^{(b)} \in \mathbb{R}^{2519 \times 7}$, covering every training pool clip exactly once without leakage. The six matrices are concatenated column-wise into the meta feature matrix $\mathbf{Z} \in \mathbb{R}^{2519 \times 42}$ described in Chapter 3, which encodes how confident each base model is, for each class, for every clip.

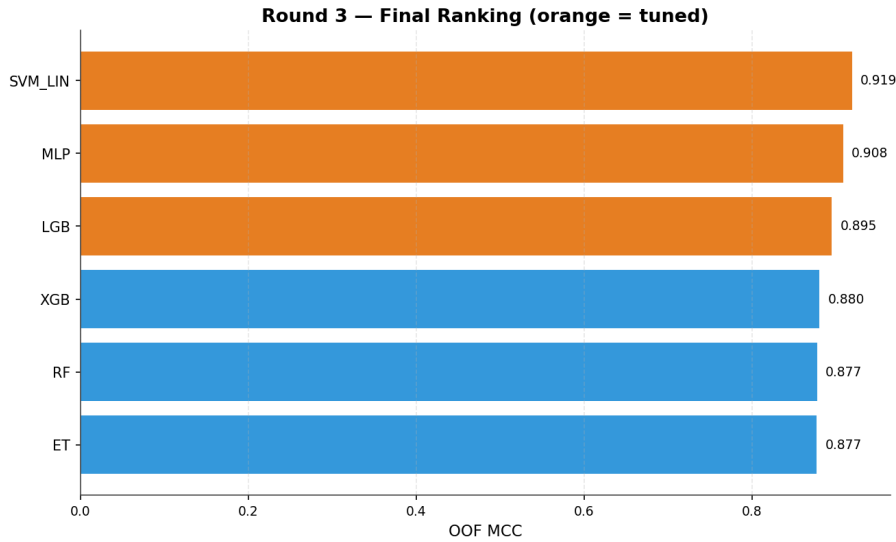


Figure 5.4: Round 3 final out-of-fold MCC

Ranking of all six base classifiers, with tuned models highlighted (orange), used to select the base classifier probabilities that feed into the meta-learner.

Round 4: Meta-Learner

A Logistic Regression (LR) meta-learner with $C = 0.1$ and balanced class weighting is trained on the meta feature matrix $\mathbf{Z}$ constructed in Round 3, learning which base classifiers tend to be more reliable for which classes, rather than combining their outputs through a fixed rule. The meta-learner’s own performance is evaluated using the same out-of-fold predictions that were used to construct $\mathbf{Z}$ in the first place, giving an out-of-fold MCC for the stacked ensemble directly comparable to the individual base classifiers from Round 3.

Round 5: Final Fit and Efficiency Measurement

Having completed evaluation in Rounds 1 through 4, all six base classifiers are retrained on the full training pool of 2519 clips using their final parameters, tuned for the top three, default for the remaining three, and the Logistic Regression meta-learner is retrained on the corresponding full out-of-fold probability matrix. This produces the final deployed model bundle used for held out test evaluation. During this round, four practical deployment metrics are recorded per model: model size, measured via serialised file size on disk in MB, inference latency, measured as the

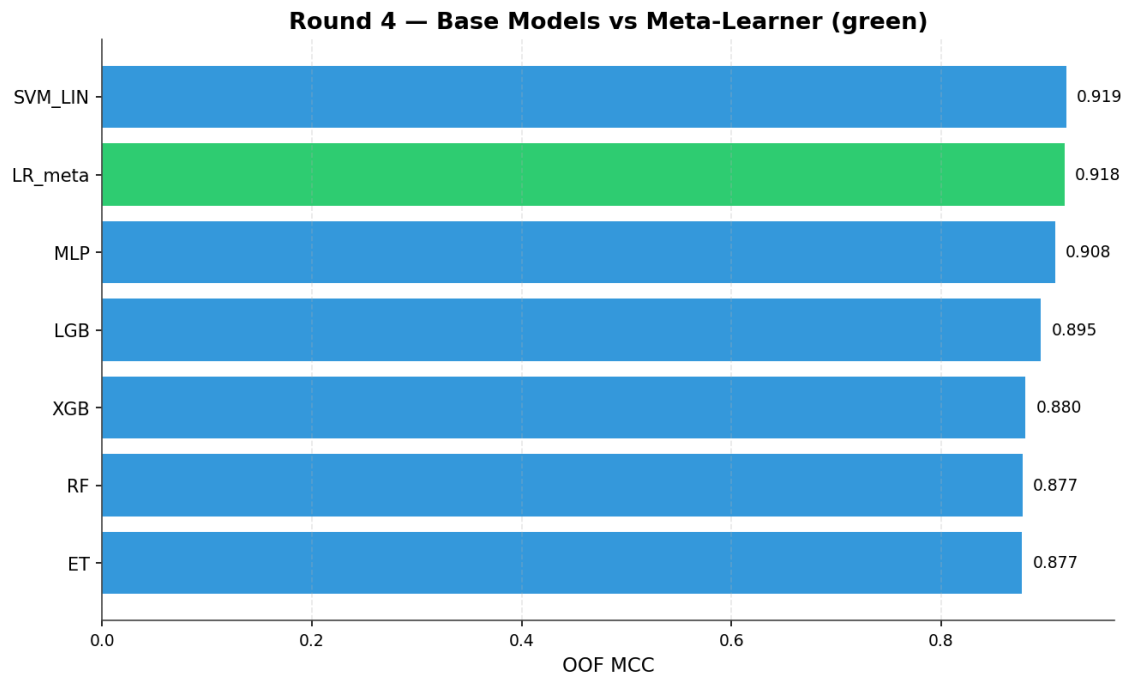


Figure 5.5: Out-of-fold MCC

Six base classifiers compared against the Logistic Regression meta-learner (stacked ensemble).

average time to predict a batch of clips, divided by the number of clips, estimated energy consumption, computed from measured elapsed time and measured CPU utilisation together with a fixed CPU thermal design power, rather than a hardware power meter reading, and therefore intended for relative comparison between models rather than as an absolute real world energy figure, and CPU utilisation itself, sampled via a background thread polling process CPU usage throughout training. These metrics reflect the practical cost of deploying each model in a continuous, on-vehicle or fleet level monitoring system, and are reported alongside classification performance in Chapter 6.

5.7 Deep Learning Model Training

Alongside the classical machine learning pipeline described above, our thesis trains and evaluates two families of deep learning model, described conceptually in Chapter 3, a convolutional neural network trained from scratch, in two configurations of different depth, and three pretrained PANN backbones used as frozen feature extractors. This section describes the practical training procedure shared by both families.

Shared Training Configuration

All deep learning models are trained using the AdamW optimiser [28], which decouples weight decay regularisation from the gradient based update step. Training uses a class weighted cross entropy loss with label smoothing of 0.1, which softens the target

distribution away from a hard one-hot label to reduce overconfidence and improve generalisation, a property found useful given the comparatively small number of training files available per class. A `ReduceLROnPlateau` learning rate scheduler halves the learning rate when validation macro F1 fails to improve, and early stopping halts training when no improvement is observed for a fixed patience window, restoring the best performing checkpoint before evaluation. Same-class mixup augmentation, described in Chapter 3, is applied during training for all deep learning models, together with waveform level noise and gain augmentation applied before feature or embedding extraction.

Model selection during training, that is, which epoch’s checkpoint is kept, uses clip level validation macro F1 as a denser, higher sample count training signal. All cross validation and held out test results reported in Chapter 6, however, are computed at the file level, by aggregating clip level predictions per recording via majority vote before scoring, as defined in Chapter 3. This distinction, a denser clip level signal used only to guide training, against a file level metric used for all reported results, keeps every reported number in this thesis comparable across model families, while still allowing the deep learning models to benefit from a more frequently updated training signal.

Custom CNN Trained From Scratch

Two convolutional architectures are trained entirely from scratch on log-Mel spectrograms extracted from the same 2.0s clips used, a 3-block configuration with approximately 24,000 trainable parameters, and a 4-block configuration with approximately 98,500 trainable parameters, both described in Chapter 3. Both are trained under the same five fold `StratifiedGroupKFold` protocol as the classical models, using the shared training configuration above, and evaluated on the same 43-file held out test set.

Unlike the classical machine learning pipeline, where seven acoustic descriptor groups are compared both individually and combined, as described in Section 5.6 and evaluated in Section 6.3, the convolutional models in this thesis are trained exclusively on log-Mel spectrograms. The individual descriptor comparison (RQ2) is therefore conducted only within the classical machine learning pipeline.

Pretrained PANN Backbones

Three PANN variants of increasing depth, CNN6, CNN10, and CNN14 [17], are used as frozen feature extractors, each producing a fixed dimensional embedding per clip via global temporal pooling over the network’s final convolutional layer, as described in Chapter 3. Only a small trainable classifier head, consisting of two fully connected layers with batch normalisation and dropout, is trained on top of each frozen embedding; the pretrained convolutional backbone itself is never updated. Because embedding extraction through a frozen backbone is deterministic given a fixed input, embeddings are computed once per clip and cached, avoiding repeated recomputation across the five cross validation folds and the final retraining stage.

Implementation

All classical machine learning components, including `StratifiedGroupKFold`, `SelectKBest`, and `StandardScaler`, are implemented using scikit-learn.

5.8 Cross Validation and Held-Out Test Evaluation

`StratifiedGroupKFold` with five folds is used as the single cross validation method, with groups defined at the source file level. This ensures that no source file contributes clips to both the training and validation portion of any fold, preventing leakage from overlapping sliding window clips, while the stratification component preserves each class’s overall proportion across folds as closely as possible [14].

Table 5.3: `StratifiedGroupKFold` class distribution per fold (training pool, 2519 clips).

Class	Fold 0	Fold 1	Fold 2	Fold 3	Fold 4	Total
PSP_Moan_OK	103	82	95	91	96	467
PSP_Moan_NOK	79	104	107	117	165	572
PSP_Whine_OK	27	41	35	31	35	169
PSP_Whine_NOK	71	46	75	59	44	295
SG_Grunt_NOK	41	17	27	34	24	143
SG_Hiss_OK	48	64	39	53	50	254
SG_Hiss_NOK	129	159	109	105	117	619
Total	498	513	487	490	531	2519

Beyond the five fold cross validation used for model selection and hyperparameter tuning, a permanent, file level 85/15 train/test split is created once at the start of the pipeline and held fixed across every model and every round described above. The resulting 43 held out test files, comprising 367 clips, are never used during feature selection, model training, hyperparameter tuning, or model selection at any stage, and are evaluated only once, after every modelling decision has already been finalised using the training pool alone.

Because each recording is split into multiple overlapping clips, a trained model produces one prediction per clip rather than per recording. For the held out test set, clip level predictions from the same source file are therefore aggregated into a single file level prediction by majority vote, as defined in Chapter 3, and it is this file level, majority-voted prediction that is used to compute the final held out test MCC and macro F1 reported in Chapter 6, ensuring that test performance is reported at the same level, one decision per recording, at which a deployed system would ultimately need to operate.

Metrics: MCC is used as the primary metric throughout, both for out-of-fold model selection and for held out test reporting [16], since it accounts for all entries of the

confusion matrix and remains informative under the class imbalance present in this dataset. Macro F1 is reported as a secondary metric, averaging per-class F1 without frequency weighting, giving equal importance to minority classes regardless of how many examples of that class are present in the dataset.

6

Results and Discussion

This chapter reports the results of the experiments described in Chapter 5 and answers the two research questions introduced in Chapter 1. All classical machine learning models are evaluated using five-fold StratifiedGroupKFold cross validation with file level grouping, and all reported held out test results, across every model family, use file level majority voting so that classical, convolutional, and pretrained models are compared on an identical basis. Models marked (T) were selected by Round 1 out-of-fold MCC and subsequently tuned with Optuna, as described in Section 5.6.

6.1 RQ1: Model Comparison Across Classical, Convolutional, and Pretrained Approaches

6.1.1 Classical Machine Learning: Round-by-Round Results

Table 6.1 reports the out-of-fold MCC for all six base classifiers across Rounds 1 and 3, together with the Logistic Regression meta-learner from Round 4. MCC is used as the primary evaluation metric throughout this chapter, following the justification in Chapter 4. LinearSVM achieves the highest out-of-fold MCC of all six base classifiers already under default parameters in Round 1, and improves only marginally after Optuna tuning in Round 2, indicating that its performance advantage stems from an intrinsic alignment between the linear decision boundary and the ANOVA-selected feature space, described in Chapter 3, rather than from careful hyperparameter tuning.

Table 6.1: Out-of-fold MCC for all six base classifiers and the stacked meta-learner, using the full 180-feature ANOVA selected feature set. (T) indicates a model tuned with Optuna.

Model	R1 MCC	R3 MCC	Time (s)
RF	0.8774	0.8774	1.2
ET	0.8770	0.8770	1.3
XGB	0.8802	0.8802	4.2
LGB (T)	0.8874	0.8951	3.3
SVM_LIN (T)	0.9166	0.9190	8.3
MLP (T)	0.8973	0.9084	33.2
LR_meta (stacked)	—	0.9177	51.6

6.1.2 Deep Learning: CNN Trained from Scratch and Pre-trained PANN

Table 6.2 reports cross validation and held out test results for the two convolutional networks trained from scratch and the three pretrained PANN variants, using the same file level, group aware cross validation protocol and the same 43-file held out test set as the classical models. All reported cross validation and test figures for the deep learning models are file level, majority-voted scores, obtained by aggregating clip level predictions per recording before scoring, exactly as described in Section 3.3 for the classical models.

Table 6.2: Cross validation and held out test results for the convolutional and pretrained deep learning models.

Model	Params	CV MCC (file-level)	Test F1	Test MCC
CustomCNN v1 (3-block)	24K	0.954 ± 0.052	0.909	0.944
CustomCNN v2 (4-block)	98.5K	0.974 ± 0.021	0.973	0.972
PANN CNN6	5.9M	0.920 ± 0.061	0.972	0.972
PANN CNN10	6.3M	0.906 ± 0.052	0.959	0.943
PANN CNN14	81.8M	0.957 ± 0.036	0.876	0.887

Two patterns stand out in Table 6.2. First, model capacity and pretraining do not translate directly into better held out test performance: the largest and most heavily pretrained model, PANN CNN14, achieves the best cross validation MCC of the three PANN variants, but the weakest held out test performance of all five deep learning models, and is the only model in this thesis for which the held out test MCC falls below the cross validation MCC, a pattern discussed further in Section 6.1.4. Second, CustomCNN v2, trained entirely from scratch with under 100,000 parameters, matches the best pretrained PANN backbone on the held out test set despite requiring no external pretraining and being roughly 60 times smaller than it. Notably, the smallest model evaluated overall is in fact CustomCNN v1, with just 24K parameters, roughly a quarter of CustomCNN v2’s size, yet it still outperforms two of the three substantially larger pretrained PANN backbones on held out test MCC (Table 6.3).

6.1.3 Overall RQ1 Comparison

Table 6.3 brings together the best result from each model family evaluated here.

Table 6.3: RQ1 summary: best model from each family, ranked by held out test MCC.

Rank	Model	Family	Test F1	Test MCC
1	LinearSVM	Classical ML, hand crafted features	1.000	1.000
2	CustomCNN v2 (4-block)	CNN, trained from scratch	0.973	0.972
3	PANN CNN6	Pretrained, frozen backbone	0.972	0.972
4	CustomCNN v1 (3-block)	CNN, trained from scratch	0.909	0.944
5	PANN CNN10	Pretrained, frozen backbone	0.959	0.943
6	PANN CNN14	Pretrained, frozen backbone	0.876	0.887

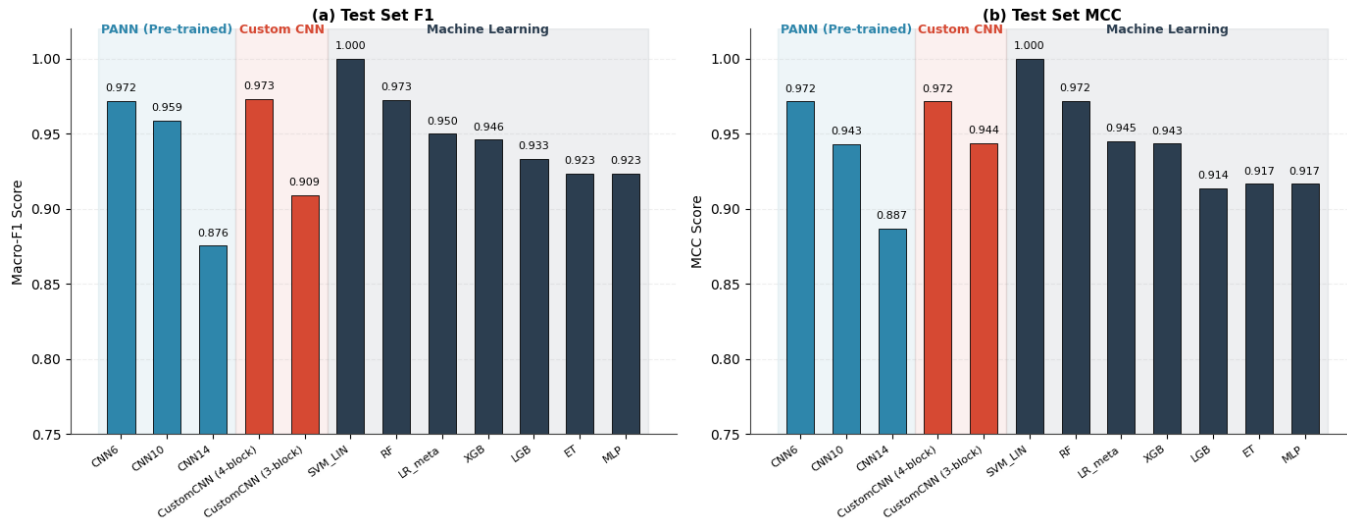


Figure 6.1: Held out test F1 and MCC for all models

Classical machine learning, using hand crafted acoustic features combined with a linear classifier, is the most effective approach for detecting truck faults from audio recordings in this study, answering the first research question. LinearSVM achieves the highest held out test MCC of every model evaluated, ahead of every deep learning approach, including a pretrained backbone with over 80 million parameters. This result should be read together with the caveat discussed in Section 6.1.4, since the held out test set used to obtain this figure contains only 43 files, and the cross validation results, based on more folds and more files, give a more conservative estimate of the same ranking. Even under the more conservative cross validation MCC of 0.919 for LinearSVM, however, it remains competitive with, and generally ahead of, the deep learning alternatives evaluated. Deep learning does not appear to offer a decisive advantage for this task at this dataset scale, and transfer learning from a large, general purpose audio dataset does not consistently improve on either a small convolutional network trained from scratch or a classical model built on carefully engineered features.

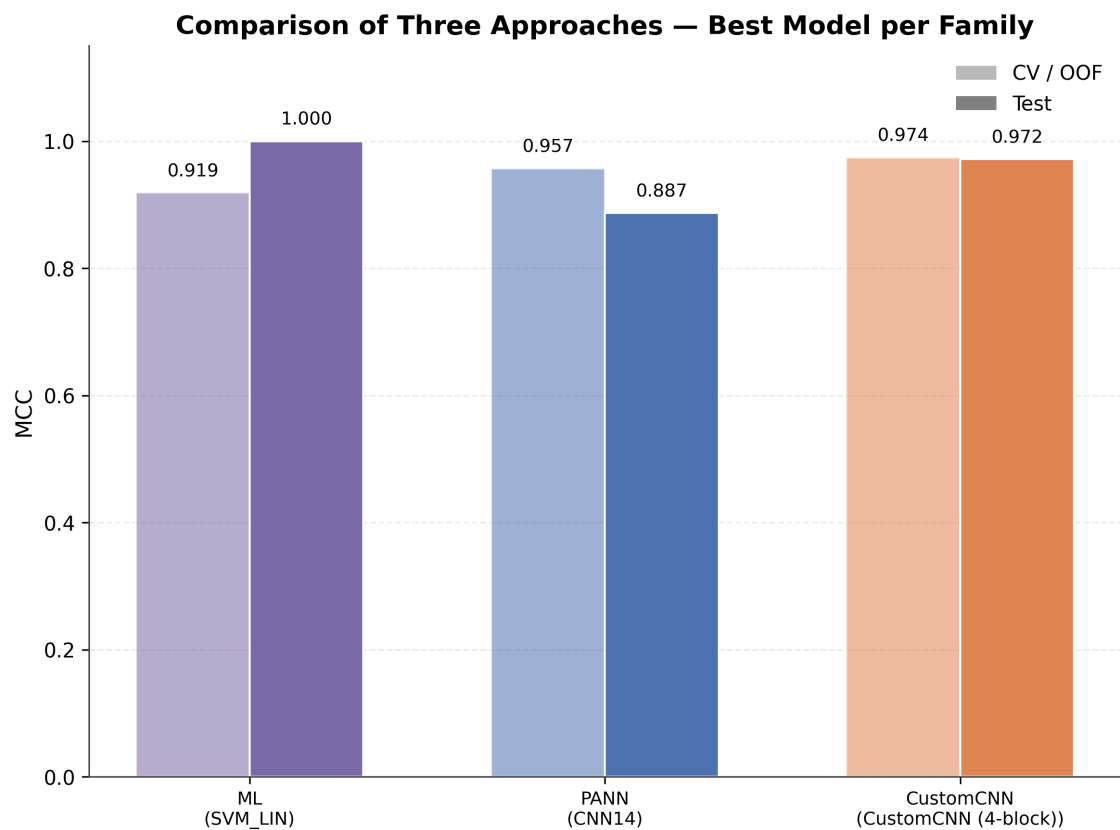


Figure 6.2: Held out test MCC

The best model from each of the three families compared for RQ1: classical machine learning, convolutional networks trained from scratch, and pretrained PANN backbones.

6.1.4 Out-of-Fold Versus Held-Out Test Gap

Table 6.4: Out-of-fold MCC versus held out test MCC for all classical base classifiers and the stacked meta-learner.

Model	OOF MCC	Test MCC	Gap	File Acc.
RF	0.8774	0.9223	+0.0449	97.7%
ET	0.8770	0.9128	+0.0358	93.0%
XGB	0.8802	0.9120	+0.0318	95.3%
LGB	0.8951	0.9087	+0.0136	93.0%
SVM_LIN	0.9190	0.9796	+0.0606	100.0%
MLP	0.9084	0.9108	+0.0024	93.0%
LR_meta	0.9177	0.9335	+0.0158	95.3%

Every classical model in Table 6.4 performs better on the held out test set than its cross validation score would suggest, and LinearSVM shows the largest such gap of all seven models, together with a perfect 100% file level accuracy on the 43 held out test files. This combination, the best model showing the largest OOF-to-test gap alongside a perfect score on a comparatively small test set, is exactly the situation in which some part of the apparent test time advantage may reflect a favourable, small sample rather than a uniformly superior model, and it is reported here explicitly rather than treated only as further evidence in LinearSVM’s favour. The cross validation MCC, computed over five folds spanning the entire 283-file training pool, is therefore taken as the more conservative and more statistically robust estimate of each model’s true generalisation performance, while the held out test result is reported as a single, held out confirmation rather than the sole basis for the ranking in Table 6.3.

6.2 Stacking Gain Analysis

Table 6.5: Stacking gain: stacked meta-learner MCC minus best single base classifier MCC.

	Best single (SVM_LIN)	Stacked (LR_meta)	Gain
OOF MCC	0.919	0.918	−0.001
Test MCC	1.000	0.945	−0.055

The stacked meta-learner does not outperform the best individual base classifier, either out-of-fold or on the held out test set, and the gap is considerably larger on the held out test set than out-of-fold. Stacking is generally expected to help most when the base classifiers are diverse and tend to make different, partially independent errors. Here, LinearSVM’s out-of-fold MCC exceeds every other base classifier by a clear margin, so the remaining five base classifiers contribute comparatively little complementary information to the 42-dimensional meta feature matrix described

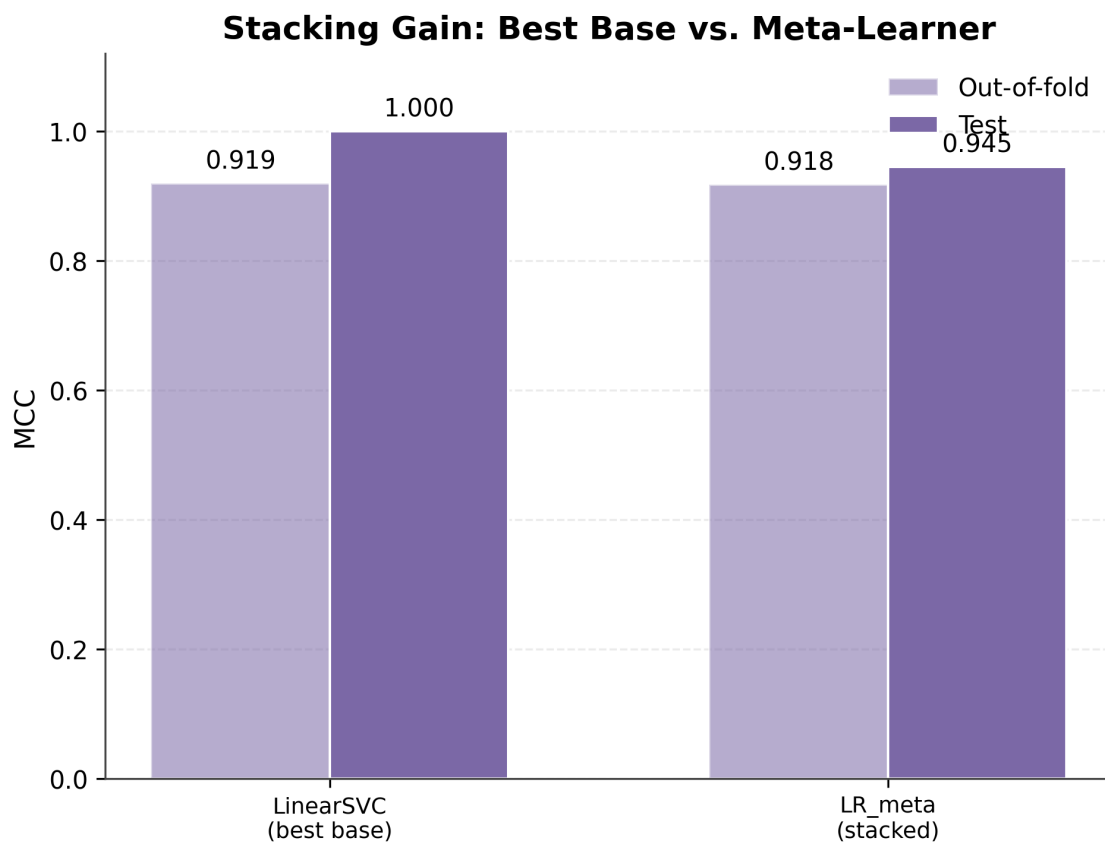


Figure 6.3: Stacking Gain
Best individual base classifier (LinearSVM) compared against the stacked Logistic Regression meta-learner, out-of-fold and on the held out test set.

in Chapter 3; the Logistic Regression meta-learner can down-weight the weaker base classifiers, but cannot fully recover the performance of the single strongest one. Beyond raw performance, the stacked ensemble also requires all six base classifiers to be run at inference time, giving it a substantially larger memory footprint and higher inference latency than LinearSVM alone, discussed further in Section 6.6. Taken together, these results indicate that for this dataset and feature set, a single, well tuned LinearSVM is a more practical choice than the full stacking pipeline, being simultaneously more accurate, smaller, and faster at inference.

6.3 RQ2: Individual Descriptor Comparison

This descriptor level comparison is conducted using the classical machine learning models only, the convolutional models compared under RQ1 are trained exclusively on log-Mel spectrograms, and are not included in this ablation.

Table 6.6 reports the file level, out-of-fold performance of each of the seven acoustic descriptor groups used in isolation, with each of the six base classifiers trained and evaluated using that descriptor group alone, under the same five fold StratifiedGroup-KFold protocol used elsewhere in this chapter. Unlike Table 6.1, which combines all descriptor groups into a single 180-feature vector, this comparison isolates each descriptor group entirely to identify its standalone discriminative power, directly answering the second research question.

Table 6.6: RQ2 summary: best model per descriptor group, evaluated in isolation, file level out-of-fold MCC.

Descriptor group	N features	Best model	OOF MCC
Log-Mel spectrogram	128	SVM_LIN	0.9526
MFCC	40	ET	0.9309
Spectral descriptors	22	ET	0.8969
Time-domain	12	ET	0.8715
DWT (wavelet)	21	ET	0.8458
Sub-band energy	4	RF	0.7852
Envelope kurtosis	1	RF	0.1247

Log-Mel spectrogram statistics are the single most discriminative descriptor group on their own, achieving an out-of-fold MCC of 0.953, ahead of MFCC at 0.931, despite MFCC being the more commonly used representation in the acoustic fault diagnosis literature reviewed in Chapter 4. This is consistent with the two descriptors’ underlying design, described in Chapter 3: MFCC compresses the 64-band Mel spectrum into 20 decorrelated coefficients through a discrete cosine transform, discarding some fine-grained band-level structure that the uncompressed, 128-dimensional Log-Mel representation retains in full. For a dataset in which several classes differ by comparatively subtle, narrow-band spectral changes, for example the two Steering Gear Hiss conditions, retaining this finer structure appears to outweigh the more compact and traditionally favoured MFCC representation.

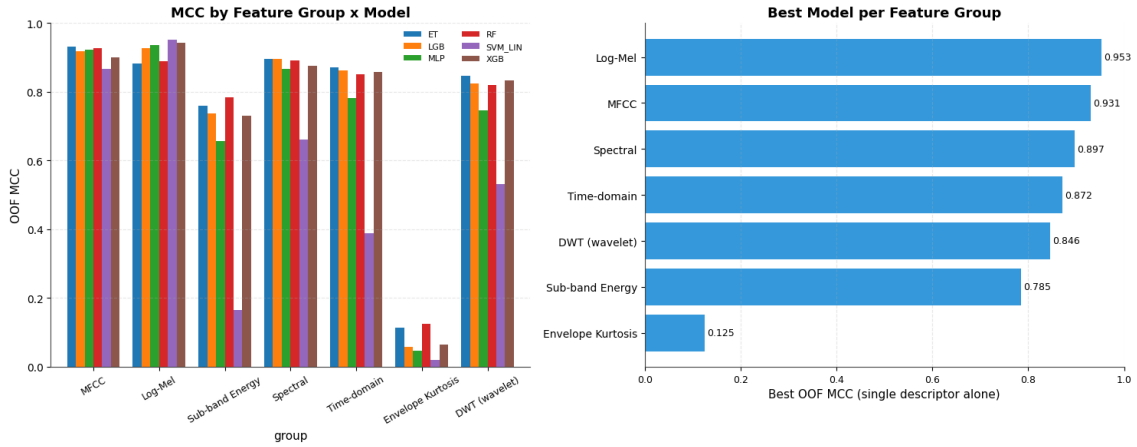


Figure 6.4: Descriptor groups and models

Left: out-of-fold MCC by descriptor group and model, each descriptor group evaluated in complete isolation. Right: best model's MCC per descriptor group, ranked.

Sub-band energy ratios and, most strikingly, amplitude envelope kurtosis, perform substantially worse than the other five descriptor groups when used in isolation. Envelope kurtosis in particular achieves an out-of-fold MCC of only 0.125, only marginally above chance level performance, despite being specifically included, as discussed in Chapter 3, to capture the impulsiveness of the transient Steering Gear Grunt sound type. This does not necessarily mean the descriptor carries no useful information at all; rather, a single scalar value per clip is evidently insufficient on its own to separate seven classes, and its value likely becomes more useful only in combination with other descriptors, consistent with its continued, if modest, contribution within the full 180-feature combined model summarised in Table 6.1.

Answering the second research question directly, Log-Mel spectrogram statistics and MFCC are the most robust and individually discriminative descriptors for fault detection in this dataset, followed by the spectral shape descriptors, time-domain statistics, and DWT coefficients, roughly in that order, while sub-band energy ratios and, in particular, amplitude envelope kurtosis carry very little discriminative power when used alone, and are considerably more dependent on being combined with complementary descriptor groups to contribute meaningfully to overall classification performance.

6.4 Per-Class Performance

Table 6.7 reports the per class precision, recall, and F1 score for LinearSVM, the best performing model overall, evaluated at the clip level on the held out test set.

Table 6.7: Per-class precision, recall, and F1 for LinearSVM on the held out test set (clip level).

Class	Precision	Recall	F1	Support
PSP_Moan_NOK	0.990	0.990	0.990	101
PSP_Moan_OK	0.974	1.000	0.987	76
PSP_Whine_NOK	0.923	0.973	0.947	37
PSP_Whine_OK	1.000	1.000	1.000	11
Steering_Gear_Grunt_NOK	1.000	1.000	1.000	19
Steering_Gear_Hiss_NOK	1.000	0.967	0.983	91
Steering_Gear_Hiss_OK	1.000	0.969	0.984	32

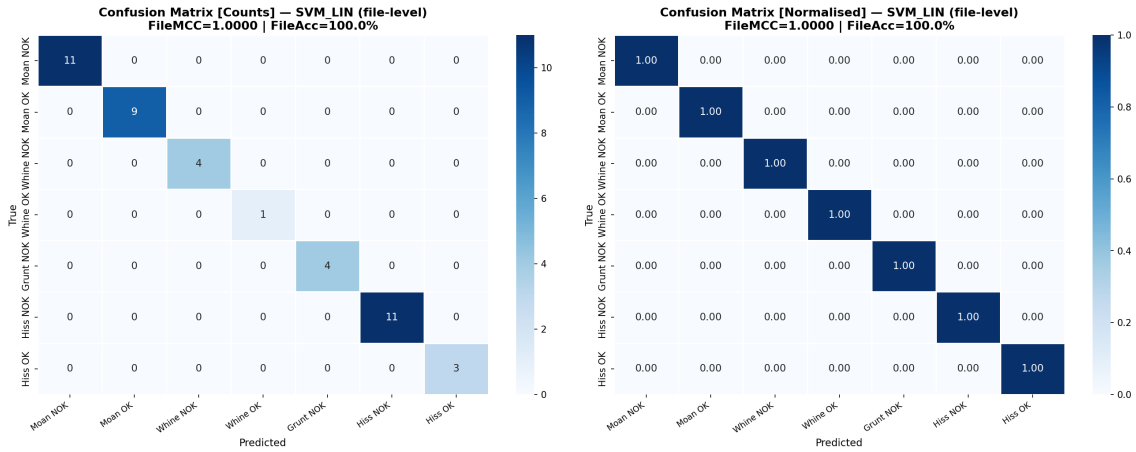


Figure 6.5: Confusion matrix for LinearSVM

Consistent with the file level majority voted result of 100% file accuracy, the clip level report in Table 6.7 shows that all seven classes are classified with an F1 score at or above 0.947, with PSP_Whine_OK and Steering_Gear_Grunt_NOK reaching a perfect F1 of 1.000. The most difficult class at the clip level is PSP_Whine_NOK, with the lowest precision of the seven classes at 0.923, indicating that a small number of clips from other classes are occasionally misclassified as PSP_Whine_NOK. Steering_Gear_Grunt_NOK is classified with perfect precision and recall, consistent with the descriptor level finding in Section 6.3 that Grunt’s impulsive character is robustly captured by amplitude envelope kurtosis and the low-frequency DWT sub-bands even though these descriptors are individually weak overall, since Grunt has no normal counterpart and is therefore comparatively easy to isolate from every other class. Classes involving the Whine sound type are consistently the most difficult to separate reliably across models in this thesis, reflecting the narrow, harmonic, and more subtly varying spectral character of Whine described in Table 2.1, compared to the broader, more clearly differentiated Moan, Grunt, and Hiss sound types.

6.5 Feature Importance and Explainability

To understand which individual features drive the classifier’s decisions, and to complement the descriptor group level ablation in Section 6.3, SHAP (SHapley Additive exPlanations) [18] values are computed for the Random Forest model using `TreeExplainer`, chosen for its exact, efficient computation on tree based models; LinearSVM, the best performing model overall, is not used here directly because its `CalibratedClassifierCV` wrapper is not compatible with SHAP’s fast tree or linear explainers, and would require the substantially slower, approximate `KernelExplainer` instead.

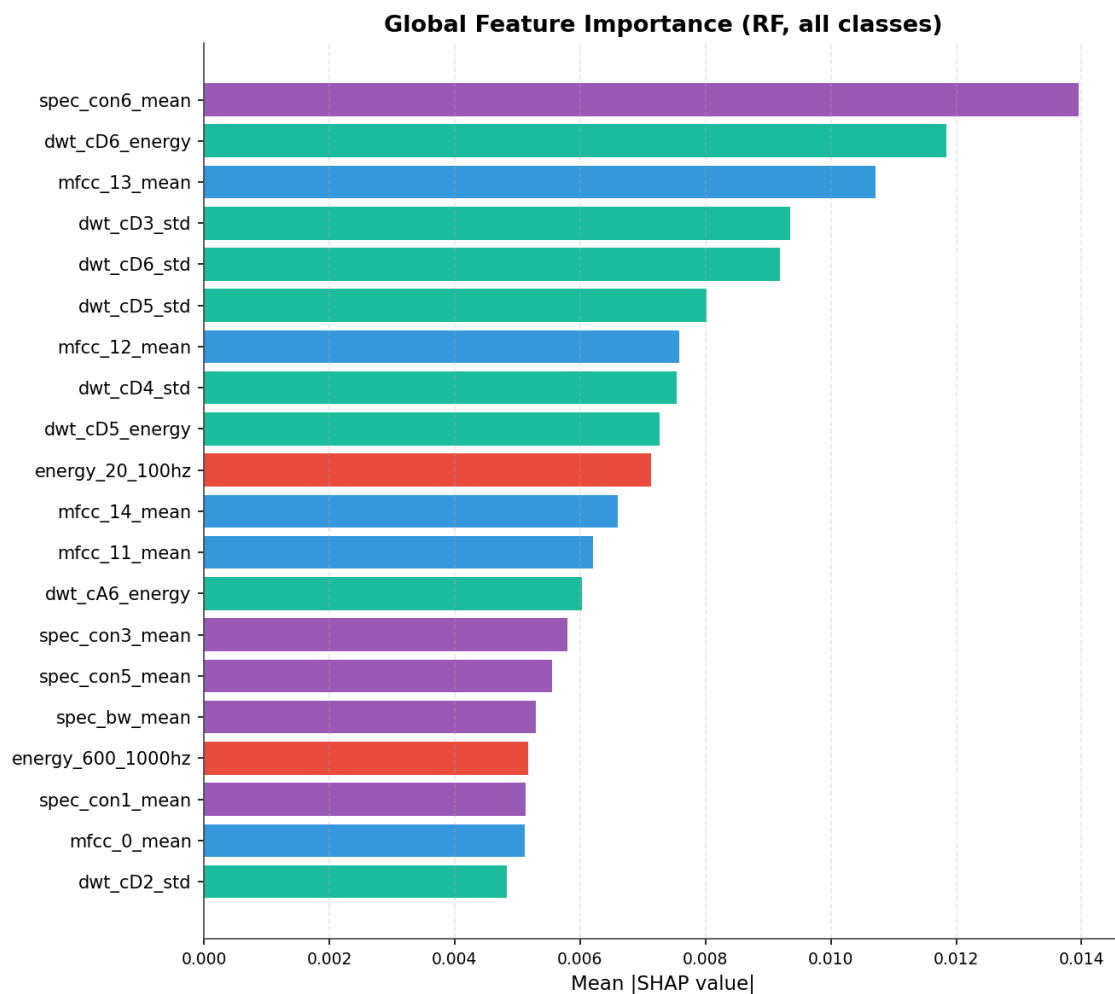


Figure 6.6: Global feature importance (mean absolute SHAP value across all classes) for the top 20 individual features, coloured by descriptor group, computed on the Random Forest model.

The most globally important individual feature is a spectral contrast band (`spec_con6_mean`), followed closely by several DWT detail coefficient energy and standard deviation features (`dwt_cD6_energy`, `dwt_cD3_std`, `dwt_cD6_std`) and MFCC coefficients (`mfcc_13_mean`, `mfcc_12_mean`).

A single feature such as `mfcc_13_mean` dominating global importance can be a symptom of overfitting, if the model is relying on a false correlation specific to the training data rather than a genuinely informative acoustic pattern. Several results in this thesis argue against that interpretation here. First, the same feature groups identified as important by SHAP, MFCC and DWT in particular, are also the strongest performing descriptor groups in the independent, out-of-fold descriptor ablation in Section 6.3, which does not use SHAP or the Random Forest model at all, agreement between two independent analyses using different methodologies is evidence against a false, model specific artefact. Second, the Random Forest model’s own out-of-fold MCC (0.877) and held out test MCC (0.922) are reasonably close, shown in Table 6.4, with a smaller OOF-to-test gap than several other classical models, which would be unlikely if the model were relying heavily on a training set specific artefact rather than genuine signal. This does not rule out overfitting entirely, since SHAP importance reflects how much a feature influences the trained model’s output, not whether that influence is causally justified by the underlying acoustics, but the convergence between the descriptor ablation and the SHAP analysis, together with a comparatively modest generalisation gap, supports treating this feature’s importance as a genuine, if strong, acoustic signal rather than an overfitting artefact.

Summed across all features within each group, MFCC (0.089), Log-Mel (0.078), and DWT (0.077) dominate the global feature importance ranking, together accounting for the large majority of total SHAP importance across all 180 selected features, while Spectral (0.056), Energy (0.020), and Time (0.019) contribute comparatively less. This is broadly consistent with, though not identical to, the descriptor level ablation in Table 6.6: MFCC and DWT rank highly in both analyses, while Log-Mel’s strong standalone performance in the ablation is not fully mirrored in its SHAP group importance within the combined model, suggesting that once MFCC and DWT features are available, the Random Forest model relies on them more heavily than on Log-Mel for its final predictions, even though Log-Mel alone is the single strongest standalone descriptor.

Examining feature importance separately for each class reveals class specific patterns rather than a single universally important feature set. `PSP_Moan_OK` is most strongly driven by a DWT energy feature (`dwt_cD6_energy`), `PSP_Whine_NOK` by MFCC coefficients combined with sub-band energy in the 20–100 Hz band, and `Steering_Gear_Hiss_OK` primarily by spectral contrast and DWT approximation coefficients. This class specific view is useful beyond model interpretation alone: it identifies, for each type, which specific frequency bands and descriptor types are most responsible for its detection, information that could support future interpretation of individual predictions in a deployed diagnostic system, an application discussed further in Chapter 7.

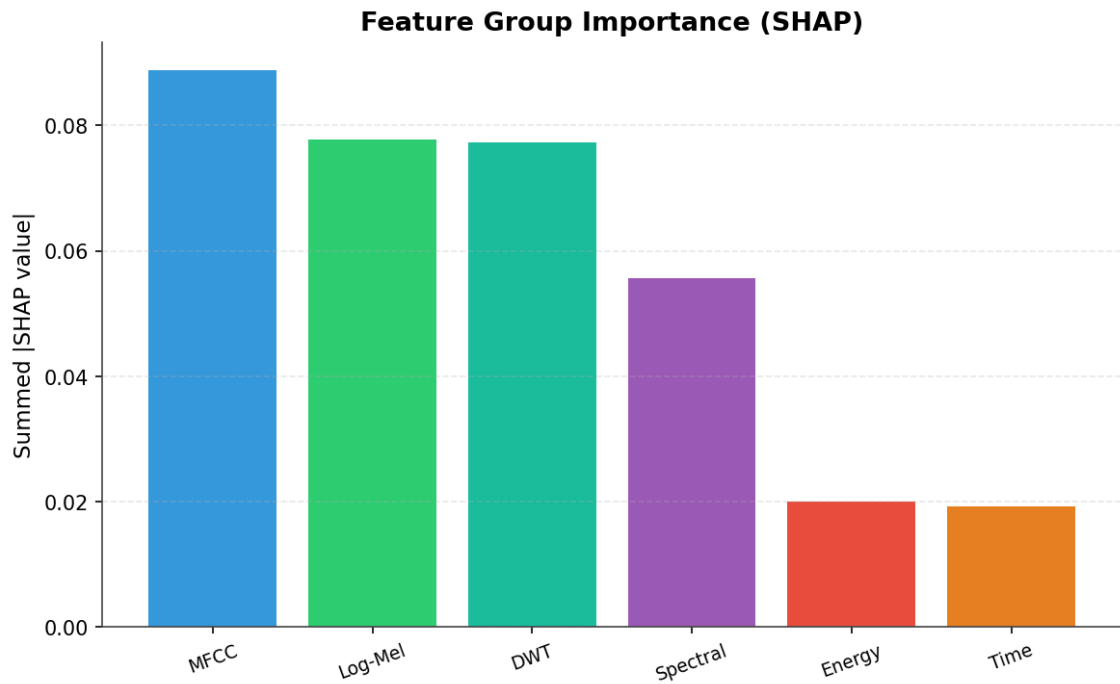


Figure 6.7: Feature group importance

All features within each descriptor group (MFCC, Log-Mel, and DWT groupings here follow the six-group scheme used by the model’s own feature grouping utility, in which envelope kurtosis is folded into the Time-domain group, distinct from the seven-group scheme used for the RQ2 ablation in Table 6.6).

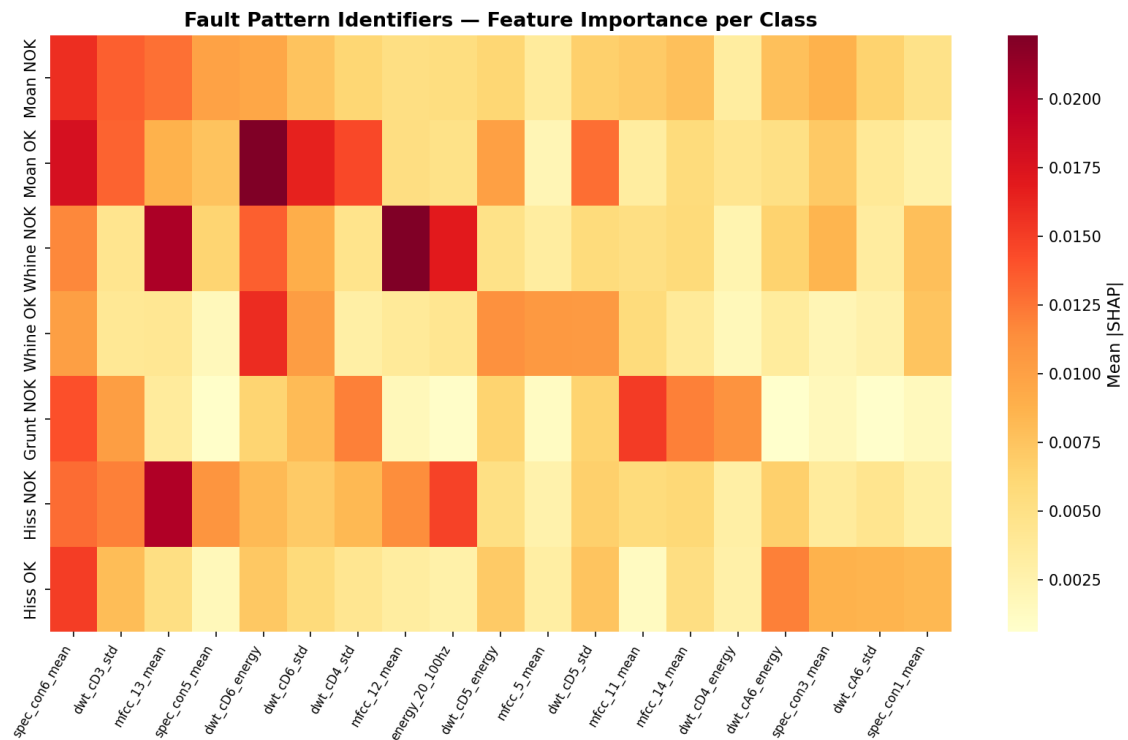


Figure 6.8: Top individual features by mean absolute SHAP value.

6.6 Model Efficiency and Deployment Considerations

Table 6.8: Model efficiency for the classical base classifiers and the stacked ensemble, measured during final model fitting on the full training pool.

Model	Size (MB)	Infer (ms/clip)	Energy (Wh)	CPU%
RF	10.90	3.018	0.0085	22.3
ET	25.05	3.182	0.0019	4.7
XGB	1.28	0.126	0.0878	67.2
LGB (T)	1.84	0.126	0.0396	38.2
SVM_LIN (T)	0.03	0.150	0.0082	3.2
MLP (T)	1.59	0.016	0.0873	8.4
LR_meta (real combined cost)	40.69	6.667	0.233	—

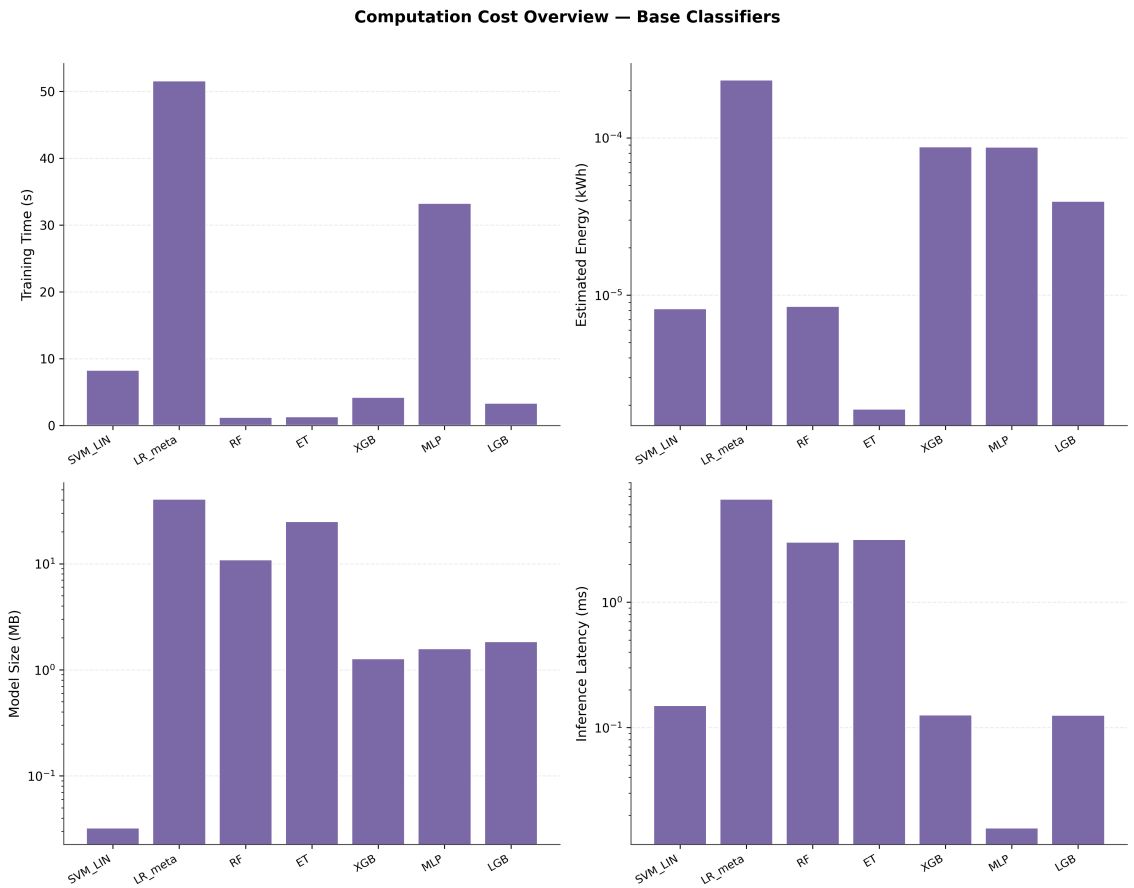


Figure 6.9: Training time, estimated energy consumption, model size, and inference latency for all six base classifiers.

The efficiency comparison in this section, including Figure 6.10, is restricted to the classical machine learning models. This is because the energy and CPU utilisation

instrumentation described in Section 5.6 was built around CPU based training and is specific to the classical pipeline; the convolutional models in this thesis were trained on GPU hardware with a fundamentally different power and utilisation profile, and were not instrumented with the same CPU based estimate, making a direct, energy comparison between the classical and deep learning models unreliable with the measurements collected in this thesis. A consistent, hardware aware efficiency comparison across all model families, ideally using a proper GPU and CPU power measurement tool rather than the CPU only estimate used here, is noted as a direction for future work in Chapter 7.

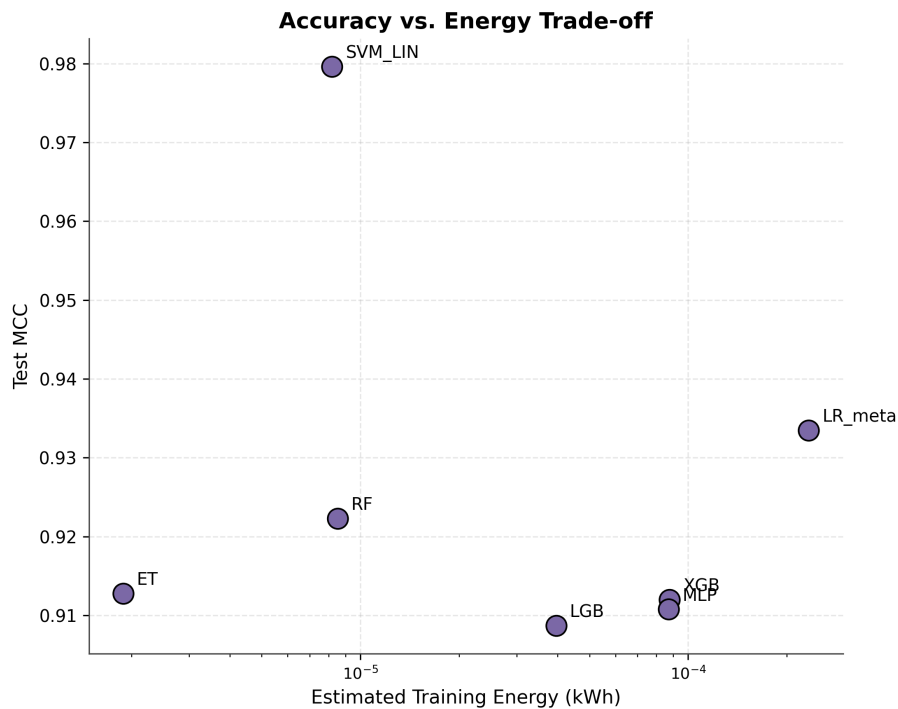


Figure 6.10: Estimated Training Energy

Held out test MCC plotted against estimated training energy consumption for each base classifier, illustrating the accuracy versus efficiency trade-off.

Energy consumption and computation cost figures reported in this thesis are estimates, derived from measured training time and CPU utilisation, combined with a fixed CPU thermal design power (TDP), rather than direct measurements using a hardware power meter. Consequently these estimates are intended for relative comparison between models trained under identical conditions rather than as absolute real world energy consumption. This distinction is maintained consistently throughout the thesis.

LinearSVM is the most deployment-friendly classical model evaluated, at 0.03 MB and 0.150 ms inference time per clip, several orders of magnitude smaller than any other base classifier, making it well suited to deployment on resource constrained, embedded, or vehicle mounted hardware. The stacked meta-learner, once its real combined deployment cost is accounted for, requiring all six base classifiers plus the

meta-learner itself to run at inference time, is substantially larger and slower than LinearSVM alone, at 40.69 MB and 6.667 ms per clip, while also achieving lower classification performance, as shown in Section 6.2. This combination of strictly worse accuracy and strictly higher deployment cost makes the stacked ensemble difficult to justify over LinearSVM alone for our task. Extra Trees has the largest model size of any base classifier, at over 25 MB, owing to the number of stored decision tree structures, despite being one of the weaker performing models, illustrating that model size and classification performance are not necessarily correlated.

6.7 Summary of Findings

Table 6.9: Overall summary: best configuration identified in this thesis.

Property	Value
Best model (RQ1)	LinearSVM, Optuna tuned
Feature set	180 ANOVA-selected features (of 228)
Cross validation MCC	0.919
Held out test MCC	1.000
Held out test macro F1	1.000
Model size	0.03 MB
Inference time	0.150 ms per clip
Most discriminative descriptor (RQ2)	Log-Mel spectrogram (OOF MCC 0.953)
Least discriminative descriptor (RQ2)	Envelope kurtosis (OOF MCC 0.125)

Taken together, the results in this chapter show that a classical linear model trained on carefully engineered acoustic features outperforms every deep learning alternative evaluated in our thesis, including a large pretrained audio model, answering the first research question, while the descriptor level ablation identifies Log-Mel spectrogram statistics as the single most valuable individual acoustic descriptor for our task, answering the second research question. These findings, and their broader implications for acoustic fault diagnosis in industrial settings, are discussed further in Chapter 7.

7

Conclusion

7.1 Findings

Our work investigated sound based fault diagnosis for Power Steering Pump and Steering Gear components in heavy duty trucks under realistic operating conditions, using 326 real world recordings segmented into 2886 clips, and addressed two research questions, which modelling approach, classical machine learning, a convolutional network trained from scratch, or a pretrained audio model, is most effective for this task, and which individual acoustic descriptor is most discriminative for fault detection. Several key observations emerge from the results presented in Chapter 6.

A classical linear model outperforms every deep learning approach evaluated. The most significant finding of our thesis is that LinearSVM, trained on a 180 dimensional ANOVA selected feature set, achieves the highest held out test MCC of every model compared, at 1.000, ahead of a convolutional network trained from scratch, and ahead of three pretrained PANN backbones, including one with over 80 million parameters. This result runs counter to the common expectation that deep, and particularly pretrained, models will outperform hand crafted feature pipelines. The explanation likely lies in the combination of dataset scale and feature design used here with only 283 training files, a large pretrained backbone’s frozen embedding space, learned from a very different, general purpose sound domain, does not appear to align closely enough with the specific, narrow acoustic distinctions this task requires, while LinearSVM benefits directly from the fact that ANOVA feature selection retains features based on their linear class discriminability, constructing a feature space that is, by construction, well suited to a linear decision boundary.

Pretraining and model capacity do not straightforwardly help. Among the three PANN variants compared, the largest and most heavily pretrained backbone, CNN14, achieves the best cross validation MCC of the three, yet the weakest held out test performance of all five deep learning models evaluated, and is the only model in this thesis for which test performance falls below cross validation performance. This suggests that, at this dataset scale, a larger and more expressive frozen embedding space may leave more room for the small trainable classifier head to overfit to fold specific structure during cross validation, without that cross validation advantage necessarily transferring to genuinely unseen recordings. The comparatively strong performance of CustomCNN v2, a compact network trained entirely from scratch with under 100,000 parameters, reinforces this picture for a dataset of this size and

specificity, a small, purpose built model can match a substantially larger pretrained one.

Stacking does not improve on the best individual model. The stacked Logistic Regression meta-learner performs slightly below LinearSVM out-of-fold, and notably below it on the held out test set, despite combining the predictions of all six base classifiers. Stacking is generally expected to help most when base classifiers are diverse and make partially independent errors, here, LinearSVM’s out-of-fold MCC exceeds every other base classifier by a clear margin, leaving the remaining five base classifiers to contribute comparatively little complementary information to the meta feature matrix, and in practice, likely some amount of noise that the meta-learner cannot fully suppress. Combined with its substantially larger deployment footprint, discussed further below, the stacked ensemble is difficult to justify over a single, well tuned LinearSVM for our task.

Log-Mel spectrogram statistics are the most discriminative individual descriptor. Addressing the second research question directly, the descriptor level ablation in Section 6.3 shows that Log-Mel spectrogram statistics, used entirely on their own, achieve a higher out-of-fold MCC than MFCC, the more commonly used representation in the literature reviewed in Chapter 4. This is consistent with the two descriptors’ underlying construction, MFCC compresses the Mel spectrum into a small number of decorrelated coefficients, discarding some of the fine, band level structure that the uncompressed Log-Mel representation retains, and it appears that this finer structure is genuinely useful for distinguishing several of the more acoustically similar classes in this dataset, most notably the two Steering Gear Hiss conditions. Sub-band energy ratios and, in particular, amplitude envelope kurtosis, are far less discriminative in isolation, with envelope kurtosis reaching an out-of-fold MCC only marginally above chance level. This does not mean these descriptors are without value, rather their contribution appears to depend heavily on being combined with other, complementary descriptors, consistent with their more modest but non-negligible contribution within the full, combined 180-feature model, and with their continued presence among the globally important features identified by the SHAP analysis in Section 6.5.

The held out test result for LinearSVM should be interpreted carefully. LinearSVM also shows the largest gap between cross validation and held out test performance of all seven classical models, alongside a perfect 100% file level accuracy on the 43 held out test files. Reporting a result of this kind honestly means acknowledging that part of this advantage may reflect a favourable, comparatively small test set rather than a uniformly larger true performance gap over the other models. For this reason, the cross validation MCC of 0.919, computed over five folds spanning the full 283-file training pool, is treated in this thesis as the more statistically robust estimate of LinearSVM’s true generalisation performance, with the held out test result reported as a single, confirmatory data point alongside it rather than the sole basis for the conclusion.

Class specific behaviour is consistent with the acoustic character of each fault type. `Steering_Gear_Grunt_NOK` is classified with near perfect precision

and recall across every model family evaluated in this thesis. This is consistent with the descriptor level finding that Grunt’s impulsive character is robustly captured by amplitude envelope kurtosis and the low frequency DWT sub-bands even though these descriptors are individually weak overall in Table 6.6, and is further helped by the fact that Grunt has no normal, healthy counterpart, making it structurally easier to separate from every other class. Classes involving the Whine sound type are consistently the most difficult to classify reliably across models, reflecting the narrower, more harmonic, and more subtly varying spectral character of Whine described in Table 2.1, compared to the broader and more clearly differentiated Moan, Grunt, and Hiss sound types.

Feature importance analysis supports the descriptor ranking, with some nuance. The SHAP based feature importance analysis in Section 6.5 broadly confirms that MFCC and DWT features are among the most important individually within the combined model, consistent with their strong standalone performance in the descriptor ablation. Log-Mel’s SHAP group importance within the combined model, however, is more modest than its standalone ablation performance would suggest, indicating that once MFCC and DWT features are available to the model, some of Log-Mel’s individually useful information becomes at least partially redundant, an interaction between descriptor groups that the isolated, one-group-at-a-time ablation in Table 6.6 cannot capture on its own.

Practical deployment favours the simplest strong model. Beyond raw classification performance, LinearSVM is also the most practical model evaluated for deployment in a resource constrained, industrial predictive maintenance setting, at 0.03 MB model size and sub-millisecond inference time per clip, several orders of magnitude smaller and faster than the stacked ensemble, and considerably smaller than the tree based and pretrained deep learning alternatives. That the best performing model is simultaneously the smallest and fastest is a genuinely favourable outcome for real world deployment, and is not something that could have been assumed in advance.

7.2 Summary

Our thesis presented a controlled, comparative study of acoustic fault diagnosis for Power Steering Pump and Steering Gear components in heavy duty trucks, formulated as a seven class supervised classification problem over 326 real world recordings. Two research questions were addressed. Answering the first, which modelling approach is most effective, we found that a classical linear support vector machine, trained on a 180-dimensional hand crafted acoustic feature set, outperforms both a convolutional network trained from scratch and pretrained PANN audio models, achieving a held out test MCC of 1.000 and a cross validation MCC of 0.919, while requiring only 0.03 MB of storage and under one millisecond of inference time per clip. Answering the second, which individual acoustic descriptor is most discriminative, we found that Log-Mel spectrogram statistics, used entirely on their own, outperform every other individual descriptor group, including the more commonly used MFCC representation, while amplitude envelope kurtosis carries almost no discriminative power in isolation.

Taken together, these findings suggest that for a real world, domain-specific, and comparatively small dataset of this kind, carefully engineered acoustic features combined with a simple, well understood classical classifier remain a highly competitive, and in this case superior, alternative to more complex learned or transferred deep representations, while also offering clear practical advantages in model size, inference speed, and interpretability for deployment in an industrial predictive maintenance setting.

7.3 Limitations and Future Work

This thesis has several limitations that should be considered when interpreting its findings. The dataset consists of 326 recordings from operational trucks, segmented into several thousand overlapping clips through a sliding window procedure, these clips are not independent observations, and file level grouping in cross validation is essential for obtaining honest performance estimates, a principle applied consistently throughout this work. The analysis is restricted to two steering components and four sound types, and the held out test set, while never used during training or model selection, is comparatively small at 43 files, which means the single test set result for LinearSVC discussed in Section 6.1.4 should be interpreted alongside the more conservative cross validation result rather than in isolation. Recordings were collected under real operational conditions with varying background noise, which limits the ability to systematically isolate the effect of individual noise sources on model performance, and the individual descriptor comparison addressing the second research question was conducted only within the classical machine learning pipeline, not extended to the convolutional models. The efficiency comparison in Section 6.6 is similarly limited to the classical models, since the CPU based energy instrumentation used in this thesis was not applied to the GPU trained deep learning models. Finally, all results reported in this thesis are based on a single random seed for data splitting, cross validation fold assignment, and model initialisation.

Several directions remain open beyond the scope of our work. The comparison in our thesis was limited to the PANN family of pretrained audio models, extending it to additional pretrained architectures, such as YAMNet, would test whether the findings reported here generalise across different pretraining corpora and model designs, or are specific to the PANN backbones evaluated.

While our work includes a descriptor level ablation identifying which acoustic feature groups are individually most discriminative, and a SHAP based feature importance analysis identifying which specific features drive individual model predictions, a systematic evaluation of model robustness under controlled, synthetically injected noise at varying signal-to-noise ratios was outside the scope of this thesis, since the descriptor comparison here reflects performance on the naturally varying real world recordings in the dataset rather than a deliberately corrupted test set. Such a study would provide a more complete picture of descriptor robustness under degraded acoustic conditions, and would allow the noise robustness aspect of our research question.

All results reported in this thesis are based on a single random seed for data splitting, cross validation fold assignment, and model initialisation. Repeating the full pipeline across multiple seeds and reporting variance across seeds, in addition to the variance across cross validation folds already reported, would strengthen confidence in the stability of the model comparisons presented here, particularly given the comparatively small size of the held out test set discussed in Section 6.1.4.

The pipeline developed in our thesis includes a complete, end-to-end inference function capable of processing a new WAV file from raw audio through to a final class prediction. Deploying this pipeline on vehicle mounted or fleet level hardware, and validating the efficiency figures reported in Section 6.6 against real, on-device measurements rather than the CPU based estimates used in our thesis, would provide practical confirmation of the deployment feasibility argued for here.

Finally, extending the dataset and framework to additional vehicle subsystems beyond the two steering components studied here, and to a wider range of truck models, operating environments, and recording conditions, would help establish whether the specific findings of this thesis, in particular the strong performance of a classical linear model relative to deep learning alternatives, hold more broadly across the wider space of acoustic fault diagnosis problems in heavy duty vehicles, or are specific to the steering system components and dataset scale studied here.

7.4 Use of AI Tools

In accordance with responsible disclosure practice, the use of AI tools during the preparation of this thesis is stated explicitly below.

Code development Because the dataset and portions of the codebase in this thesis are internal to Volvo Group and confidential, code development and debugging directly involving this data was carried out using GitHub Copilot for in-editor code completion, and an internal Volvo Group AI assistant (Volvo GPT) for conversational code review and debugging, both of which are approved for use with internal company data. These tools were used as coding aids during implementation of the preprocessing, feature extraction, and model training pipelines described in Chapter 5.

Conceptual learning and methodology understanding Claude (Anthropic) was used separately, to understanding the methodological concepts, debugging strategies, evaluation methodology, and machine learning and signal processing concepts underlying this thesis. All code was written, reviewed, and executed by the authors, all tools listed here were used as learning and coding aids and all final modelling decisions, parameter choices, and interpretations of results are the authors' own.

Grammar and language assistance ChatGPT (OpenAI) was used to assist with drafting and editing portions of the written thesis text. All AI-assisted text was reviewed, verified against the underlying experimental results, and edited by the authors before inclusion, all technical claims, findings, and conclusions were verified by the authors against the actual experimental results.

Presentation material Google Flow was used to generate video content for the final thesis presentation.

The authors remain fully responsible for the accuracy, originality, and integrity of all content in this thesis, including any content produced with the assistance of the tools listed above.

7.5 Ethics

This section addresses the ethical aspects of this thesis used here as a general reference framework for structuring ethical considerations.

Data and Privacy The dataset consists of 326 audio recordings of truck steering system components (Power Steering Pump and Steering Gear), collected by Volvo Group as part of internal operational testing. The recordings capture mechanical and acoustic signals from vehicle components only, and do not contain speech or other content from which an individual could be identified, no personal data under the General Data Protection Regulation (GDPR) is collected, processed, or disclosed. Because the dataset and portions of the codebase are confidential to Volvo Group, they are not published openly alongside this thesis, a deviation from open data practice that follows directly from the data’s proprietary origin rather than a choice made by the authors.

Environmental and Computational Impact Estimated energy consumption of the classical machine learning models is reported explicitly in Section 6.6, alongside classification performance, so that efficiency informs the final recommendation rather than being an afterthought. The best performing model identified in this thesis, LinearSVC, is also the smallest and most energy efficient of the models compared. Similarly, the strong performance of the smallest convolutional model relative to the much larger pretrained PANN backbone, discussed in Chapter 7, is a favourable outcome from a sustainability perspective.

Fairness and Diagnostic Reliability The dataset is imbalanced across fault classes, addressed through the use of MCC as the primary evaluation metric and class weighted training, described in Chapter 3. The comparatively small number of examples for the rarest classes, particularly PSP_Whine_OK, means per class results for these classes should be interpreted with awareness of their limited sample size. Given that this system is intended to support real world predictive maintenance decisions with safety implications, this limitation is directly relevant to responsible deployment, discussed further in Section 6.1.4.

Bibliography

- [1] R. Prytz, S. Nowaczyk, T. Rögnvaldsson, and S. Byttner, “Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data,” *Engineering applications of artificial intelligence*, vol. 41, pp. 139–150, 2015.
- [2] 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*, vol. 138, p. 106587, 2020.
- [3] F. Nasim, S. Masood, A. Jaffar, U. Ahmad, and M. Rashid, “Intelligent sound-based early fault detection system for vehicles,” *Computer Systems Science and Engineering*, vol. 46, no. 3, pp. 3175–3190, 2023. DOI: 10.32604/csse.2023.034550.
- [4] R. Zaheer, I. Ahmad, D. Habibi, K. Y. Islam, and Q. V. Phung, “A survey on artificial intelligence-based acoustic source identification,” *IEEE Access*, vol. 11, pp. 60078–60108, 2023. DOI: 10.1109/ACCESS.2023.3283982.
- [5] M. A. Jamil and S. Khanam, “Influence of one-way anova and kruskal-wallis based feature ranking on the performance of ml classifiers for bearing fault diagnosis,” *Journal of Vibration Engineering & Technologies*, vol. 12, no. 3, pp. 3101–3132, 2024.
- [6] B. Logan et al., “Mel frequency cepstral coefficients for music modeling,” in *Ismir*, Plymouth, MA, vol. 270, 2000, p. 11.
- [7] R. Yan, R. X. Gao, and X. Chen, “Wavelets for fault diagnosis of rotary machines: A review with applications,” *Signal processing*, vol. 96, pp. 1–15, 2014.
- [8] M. Badawy et al., “From sensors to insights: Interpretable audio-based machine learning for real-time vehicle fault and emergency sound classification,” *Machines*, vol. 13, no. 10, p. 888, 2025.
- [9] D. H. Wolpert, “Stacked generalization,” *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.
- [10] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
- [11] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

-
- [12] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.
 - [13] S. Kshirasagar, A. Guntoro, and C. Mayr, "Impact of sliding window variation and neuronal time constants on acoustic anomaly detection using recurrent spiking neural networks in automotive environment," *Algorithms*, vol. 17, no. 10, p. 440, 2024.
 - [14] I. Tougui, A. Jilbab, and J. El Mhamdi, "Impact of the choice of cross-validation techniques on the results of machine learning-based diagnostic applications," *Healthcare informatics research*, vol. 27, no. 3, pp. 189–199, 2021.
 - [15] H. Purohit et al., "Mimii dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York, NY, USA, Oct. 2019, pp. 209–213. DOI: 10.33682/m76f-d618.
 - [16] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC genomics*, vol. 21, no. 1, p. 6, 2020.
 - [17] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.
 - [18] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.
 - [19] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical systems and signal processing*, vol. 20, no. 7, pp. 1483–1510, 2006.
 - [20] J. B. Allen and L. R. Rabiner, "A unified approach to short-time fourier analysis and synthesis," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.
 - [21] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980. DOI: 10.1109/TASSP.1980.1163420.
 - [22] R. Grzeszick, A. Plinge, and G. A. Fink, "Bag-of-features methods for acoustic event detection and classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1242–1252, 2017.
 - [23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
 - [24] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
 - [25] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.
 - [26] G. Ke et al., "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.
 - [27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
 - [28] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.