# Large-Scale Acoustic Automobile Fault Detection: Diagnosing Engines Through Sound

Dennis Fedorishin
dcfedori@buffalo.edu
University at Buffalo
Buffalo, New York, USA

Justas Birgiolas
ACV Auctions, Inc.
Ronin Institute
Buffalo, New York, USA

Deen Dayal Mohan
University at Buffalo
Buffalo, New York, USA

Livio Forte III
ACV Auctions, Inc.
Buffalo, New York, USA

Philip Schneider
ACV Auctions, Inc.
Buffalo, New York, USA

Srirangaraj Setlur
University at Buffalo
Buffalo, New York, USA

Venu Govindaraju
University at Buffalo
Buffalo, New York, USA

## ABSTRACT

In this paper we present *AMPNet*, an acoustic abnormality detection model deployed at ACV Auctions to automatically identify engine faults of vehicles listed on the ACV Auctions platform. We investigate the problem of engine fault detection and discuss our approach of deep-learning based audio classification on a large-scale automobile dataset collected at ACV Auctions. Specifically, we discuss our data collection pipeline and its challenges, dataset preprocessing and training procedures, and deployment of our trained models into a production setting. We perform empirical evaluations of *AMPNet* and demonstrate that our framework is able to successfully capture various engine anomalies agnostic of vehicle type. Finally we demonstrate the effectiveness and impact of *AMPNet* in the real world, specifically showing a 20.85% reduction in vehicle arbitrations on ACV Auctions' live auction platform.

## CCS CONCEPTS

• **Hardware** → *Failure prediction*; • **Computing methodologies** → **Neural networks**.

## KEYWORDS

Engine fault detection, classification, multi-modal feature fusion, audio, vibration

**ACM Reference Format:**
Dennis Fedorishin, Justas Birgiolas, Deen Dayal Mohan, Livio Forte III, Philip Schneider, Srirangaraj Setlur, and Venu Govindaraju. 2022. Large-Scale Acoustic Automobile Fault Detection: Diagnosing Engines Through Sound. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539066


*KDD '22, August 14–18, 2022, Washington, DC, USA*

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539066

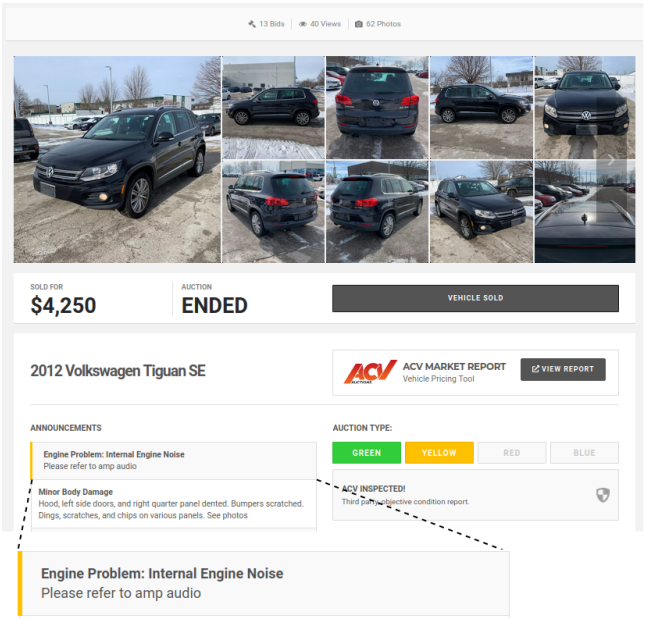


**Figure 1: An example condition report of a vehicle sold on the ACV Auctions platform. After a vehicle condition inspector created the report, AMPNet detected internal engine noise in the vehicle's engine audio recording. The report was then updated to reflect the serious engine issue to potential buyers.**

## 1 INTRODUCTION

Automobile condition is a critical factor in vehicle purchasing decisions. To assess used vehicle value, a buyer would benefit immensely from a detailed and accurate assessment of the vehicle's condition. This is especially true when evaluating vehicles listed on online auction platforms, where physical inspections are limited or impractical.

ACV Auctions is an online wholesale automotive marketplace that specializes in providing a detailed and transparent assessment of vehicles listed on its auction platform. A detailed and honest

portrayal of a vehicle's condition is likely to give buyers greater confidence in buying a vehicle sight unseen.

The typical vehicle's lifespan on the ACV Auctions platform is as follows: A user (seller) decides to list a vehicle on the auction platform. A trained vehicle condition inspector (VCI) from ACV Auctions comes to the vehicle to generate a condition report (CR), which is a unique aggregation of data regarding the vehicle's current state. Later, when the vehicle's auction launches, this condition report is presented to buyers interested in bidding for the vehicle.

A vehicle's condition report is a comprehensive review of all of its characteristics, damages, and faults. It includes over 40 photos of the exterior, interior, undercarriage, and engine bay, in addition to metadata such as the VIN number, odometer reading, engine type, fuel type, and other such information. A key component of the condition report (which this paper is based upon) is the Audio Motor Profile [4] (AMP™), which is ACV Auctions' process of recording a live audio and accelerometer vibration sample of a vehicle's engine in its operating condition. The AMP™ is a 30 second audio and vibration recording using a microphone and accelerometer placed on the engine. The VCI starts the recorder placed on the engine block, then enters the vehicle, turns it on, lets it idle for a period of time, then depresses the accelerator twice, and finally shuts the vehicle off. The audio recording portrays the sound emitted from the engine in each state of operation to allow buyers to assess the condition of the vehicle's engine.

When a VCI is creating a condition report, collecting metadata, and recording the AMP™, they also record any damage or abnormalities, including cosmetic and mechanical issues. However, some abnormalities that may be present in a vehicle are more easily discernible than others. For example, physical damage such as dents and scratches on a vehicle's exterior are relatively easy to identify and disclose while mechanical faults within the engine can be more difficult to detect. While these mechanical faults are typically not visible, they are often audible. If an engine has an internal problem, it will often emit an anomalous noise that can be discerned by comparing it to a set of reference non-faulty engine recordings. For example, a high pitch squealing noise can indicate excessive wear on a serpentine belt or bearing.

After a VCI records an AMP™, they list any abnormalities that they hear from the operation of the engine. However, many engine faults are very subtle issues that are often only discernible by automobile engine experts. As a result, some engine faults of an anomalous vehicle may be missed by the VCI who created the condition report and the faulty vehicle may be subsequently launched on the auction platform with the *appearance* of it being a non-faulty, problem-free vehicle.

In such cases, buyers may unknowingly buy these anomalous vehicles. If the buyer finds an undisclosed issue after they receive the vehicle, they are eligible to file for an *arbitration*, which is a method of resolving a dispute over a misadvertised vehicle. ACV Auctions, according to their arbitration policy, will cover the costs of certain engine issues with vehicles that are not considered normal wear-and-tear components. If a buyer proceeds with filing an arbitration claim, they obtain a repair quote from an independent automotive mechanic that discloses the exact fault with the vehicle and its estimated repair cost. ACV Auctions will then cover this repair cost on behalf of the buyer of the misadvertised vehicle.

However, this arbitration process is avoidable for both parties. If the vehicle was listed with disclosures of all of its anomalies and faults, the buyer would not file an arbitration as all issues are disclosed to the buyer at the time of the auction. Having a system that can more accurately detect engine faults and disclose them to buyers before vehicles are sold can directly reduce the amount of arbitration claims that are processed.

In this paper, we seek to answer the following question: *Can we effectively diagnose faulty automobile engines in an automated fashion at scale?* Harnessing this ability would allow for the creation of more accurate condition reports, removing the need for humans to label engine faults, and reducing the amount of engine arbitration claims. It may also be applicable for a variety of other use cases outside of the automotive auction domain such as automated machine condition monitoring and engine repair estimates.

Research in automatic engine fault detection has been extensively explored along multiple directions. Early works such as [1, 2, 20] use spectral analyses, sound pressure levels, and frequency intensities to determine engine noises through audio recordings. Other approaches include using SVM classifiers [17, 31] and decision trees [28]. Recent works use deep neural networks for classifying engine faults [19, 25, 26].

As an alternative to using audio, some works investigate engine fault detection through vibration signals. In [28], engine faults are classified using accelerometer-recorded vibration signals. [13] uses vibration signals to specifically detect engine misfires. [31] extends these studies and performs fusion of signals from multiple vibration sensors mounted on different portions of an engine to improve detection performance. [3, 7, 12] compare the acoustic and vibration modalities and quantify their performance on detecting specific engine faults.

However, many of these works focus on detecting specific engine faults on a test bed of a single engine in a controlled testing environment [17, 19, 28]. Others perform fault detection on small-scale datasets of a specific type, for example only using diesel engines [3, 26, 30]. While these methods prove that certain engine faults can be detected through audio and vibration signals, they do not scale to a wide range of vehicles and engine types.

In this paper, we describe a production-deployed system that automatically detects and flags generic engine faults at scale. We introduce **A**udio **M**otor **P**rofile **Net**work (AMPNet), a novel multi-label classification network that predicts engine faults from the fusion of features extracted from engine audio, accelerometer-recorded vibration, and tabular metadata of vehicles. We discuss our pipeline for using AMPNet for detecting undisclosed engine faults after the creation of automobile condition reports on the ACV Auctions platform. Our experimental results show that we are able to accurately detect multiple engine faults across a large-scale dataset comprised of vehicles sold in the United States. We further discuss our process of deploying AMPNet into a production setting to automatically flag vehicles with engine faults and the subsequent reduction of engine-related arbitration claims across the auction platform. Our contributions in this paper are summarized are as follows:

- To the best of our knowledge, this is the first large-scale study investigating the detection of engine faults across various models of vehicles in the United States.
- We discuss our pipeline for collecting multi-modal data for engine fault detection across a large collection of vehicle and engine types.
- We discuss the challenges of acoustic engine fault detection in terms of data collection and applying deep learning models to the task. Specifically, we discuss the difficulty in sourcing quality labels for supervised learning and the unconstrained nature of the recorded data that we collect.
- We introduce our architecture design, AMPNet, which performs engine fault detection through the use of multi-modal feature fusion of audio, accelerometer-recorded vibration, and tabular metadata of individual vehicles.
- We conduct extensive experiments with AMPNet on ACV Auction's engine fault dataset to show the efficacy of our model. Further, we investigate AMPNet's effect on engine-related arbitration claims after deploying it on the ACV Auctions live auction platform.


**Figure 2: AMP™ device used to record audio for the creation of a vehicle's condition report and subsequently used for the creation of our engine fault dataset.**

## 2 RELATED WORKS

### 2.1 Audio Classification / Audio Tagging

Audio classification is a fundamental task in acoustic signal recognition with the goal of predicting the category of a certain audio sample. Audio tagging is a similar task of predicting the presence or absence of certain audio tags or events. Audio classification/tagging applications range from large-scale audio events [10] to classifying acoustic scenes [18], urban sounds [5, 22], and others. Early works in audio classification used hand-crafted features such as Mel-frequency cepstrum coefficients (MFCC) with Hidden Markov models. [8]. With the advent of convolutional neural networks (CNNs), researchers have used CNNs to learn features from Mel-spectrograms and other hand-crafted representations to perform classification [6, 21]. Further, researchers have investigated classification using features learned from the raw audio waveform using 1D CNNs [16, 23]. Others [9, 15, 32] have found that the fusion of features learned from multiple representations of audio (namely the raw waveform and Mel-spectrogram) improve audio classification performance across a variety of tasks. Recently, researchers have used the Transformer [29] architecture to create purely self-attention based classification networks that achieve state-of-the-art performance [11]. As discussed further in Section 4.1, we frame the acoustic engine fault detection task as a multi-label audio classification task. We draw upon these works to construct our audio classification networks.

## 3 DATA COLLECTION

In this section, we discuss our methodology for collecting a large scale dataset for engine fault detection. To train a model to detect engine faults across a wide variety of vehicles, we needed a training dataset that covered many makes and models of vehicles with a variety of engine faults. Our data was collected with the collaboration of over 1,000 VCIs spread across the United States. Before listing a vehicle on the ACV Auctions platform, a VCI creates a condition report of the vehicle. We subsequently collect these condition reports across all vehicles to use as training and evaluation data. From each instance of a vehicle's condition report, we utilize three key pieces of information for detecting engine faults:

*3.0.1 Engine audio.* All engine audio recordings are created using ACV Auctions AMP™ device, as shown in Figure 2. The device is an Apple iPhone X paired with a Zoom IQ6 stereo microphone. The iPhone and microphone are mounted inside a plastic enclosure with a rubber base to dampen any unwanted rattling of the device and prevent the device from sliding or falling during the recording process. The audio is recorded in two channels using a sampling rate of $44.1kHz$. A properly recorded audio recording has a duration between 25-35 seconds and starts with the vehicle turned off, followed by the engine start, a period of time where the vehicle has a sustained idle, then two depressions of the accelerator where the engine is allowed to fully reach idle between subsequent revs. Finally, the vehicle is turned off to complete the recording. This sequence provides insight into every operating point of an engine and encompasses certain faults that may be only present in particular conditions (e.g. during startup or in a certain RPM range).

*3.0.2 Accelerometer vibration.* Alongside recording the engine audio, we record the accelerometer vibration signal as well to provide another modality that can be used to predict engine faults. It is well known in the automotive field that various engine faults can present themselves as abnormal vibrations generated by an engine [3, 7, 12, 13, 28, 31]. A second modality such as vibration can also be used in the case of a failed or poor recording of audio, and vice-versa. The vibration signal is recorded using the built-in accelerometer on the iPhone X and is recorded using the accelerometer's maximum sampling rate of $100Hz$. The vibration captures a three-channel signal that represents the three spatial dimensions. The X axis is represented across the left and right side of the phone, the Y axis is across the top and bottom of the phone, and the Z axis is through the screen and back of the phone. To keep consistency between the

accelerometer and audio recordings, we record the audio and vibration simultaneously so that they are temporally consistent. This temporal consistency may be useful in extracting complementary features from both modalities or uncovering certain correlations of each modality at different operating points of the engine to improve fault detection performance.

*3.0.3 Tabular metadata.* Every inspected vehicle has an associated set of tabular metadata that describes the vehicle and its characteristics, for example, the year, make, model, and engine type. It has been observed that certain vehicles or engine types have "common problems". An automotive expert can often estimate which faults are most likely to be present based on just basic vehicle information. Tabular metadata in conjunction with audio and vibration can help uncover common faults among certain makes and models of vehicles. In addition, some faults may present themselves as similar signals in the audio and vibration modality, however they may actually be different faults based upon the type of engine that produces the noise. We utilize a combination of numerical and categorical metadata that ranges from generic vehicle information to specific engine-related properties and diagnostics. General vehicle information contains the vehicle model, year of production (age of vehicle), odometer, drivetrain (front, rear, or all-wheel-drive), and transmission type (manual or automatic transmission). Engine-specific metadata include engine displacement, properties, fuel type, and a list of active on-board diagnostic (OBD) codes.

Active OBD codes on a vehicle indicate that the vehicle's own diagnostic systems found a vehicle fault, some of which relate to mechanical engine faults. While some engine faults can be directly determined from these codes, many faults are not detectable by on-board sensors and are instead detectable by listening to abnormal noises. Fusing this information can improve detection performance of faults that are both detectable and undetectable by the vehicle's own diagnostic system. Further, fusing features from audio, vibration, and metadata can uncover interdependencies between active OBD codes and captured anomalous signals.

The final metadata component we use is the number of incomplete OBD monitors present on the vehicle. In an attempt to obfuscate certain engine faults, some vehicle sellers will erase diagnostic codes and monitors from a vehicle. Although the actual fault is not fixed, the vehicle's diagnostic system will not detect these faults for a period of time. After a sustained driving period, these fault codes will reappear and monitors will show a "complete" status. If a large portion of OBD monitors in a vehicle are incomplete, that vehicle is often biased towards having some active fault.

In Section 5, we experimentally show that the fusion of features from these three modalities improves the overall performance of AMPNet across all engine faults.

## 3.1 Engine Fault Classes

To perform large-scale engine fault detection, we curated a set of engine faults that are present across a large amount of vehicle models and engine types. Using hyper-granular engine faults limit their applicability across all vehicles. Diesel engines can exhibit specific faults (glow plug failure, for example) that do not occur with gasoline engines. Our goal is to create a single model that is able to perform general engine fault detection on any vehicle. As a result, we selected five generic engine faults that are found on vehicles agnostic of engine type and vehicle manufacturer:

- *Internal engine noise (IEN)*: Noises that originate from the internals of a vehicle's engine. The two main categories of internal engine noise are ticking and knocking, which are two similar sounds that present themselves as consistent tapping. Ticks are often quieter, soft taps that originate from the valvetrain of an engine. Ticks are often considered less severe while knocks are often deeper, louder sounds that originate from the lower internals of the engine and are almost always an indication of severe engine damage.
- *Rough running engine (RR)*: Instability in the operation of the engine. This fault encapsulates any abnormal vibrations that are emitted from the engine, often from unstable idles. A rough running engine may have an unstable idle when the engine is unable to maintain a stable rotation rate. In addition, vehicles where accelerations are delayed or slowed are also considered as having a rough running engine.
- *Timing chain issue (TC)*: A vehicle that has an issue related to its timing chain, often presenting itself as a stretched chain that rattles audibly during a vehicle start. It is important to note that while most vehicles have timing chains, some vehicles instead have timing belts, which do not exhibit these audible faults. However we still deem this as an important engine fault to predict as it is regarded as a serious fault that often precedes catastrophic engine damage. In addition, it is a commonly missed fault by inspectors.
- *Engine accessory issue (ACC)*: These faults are related to accessory components on the engine. For example, power steering pump whines, serpentine belt squeals, bearing damage, turbocharger issues, and any other anomalous components that are not internal to the engine block.
- *Exhaust noise (EXH)*: Vehicles that have a cracked or damaged exhaust system near the engine often exhibit a noise similar to a tapping noise that engine ticks exhibit. While exhaust noises are considered less severe faults, they are still a commonly missed fault that requires attention.

When a VCI is recording the condition report of a vehicle, they label the corresponding vehicle's engine according to these five engine faults after they record the audio and vibration samples. We use the collected data with the five engine fault labels to construct training, validation, and evaluation datasets for the task of engine fault detection. In our experiments and discussion, we abbreviate these engine faults as IEN, RR, TC, ACC, and EXH, respectively.

## 3.2 Data Challenges

There are several inherent challenges with the data we collect for engine fault detection. Although VCIs are trained in the specific data collection process, there are still several variations that may occur between recorded samples. For example, when recording the audio and vibration, a recording should contain all of the engine states: the start, idle, two revs, and shut off. As VCIs manually perform these steps, the recordings may have slight inconsistencies. For example, a recording may only have one rev, or the engine may not reach an idle level before a subsequent rev. There is also no constraint to the duration and time of each engine state. For

example, a vehicle's first acceleration may be 10 seconds or 20 seconds into the recording.

There is also no guarantee of the orientation of the recording device in relation to the vehicle. We capture tri-axial (x-y-z cardinal directions) accelerometer data which has a correspondence to the recording device's orientation. However, given that the device may be oriented in any position on any vehicle, we do not know a given accelerometer recording's orientation in relation to its vehicle.

The other challenge with the collected data is in regards to the engine fault labels. As previously described, the arbitration process occurs when a vehicle is mistakenly labeled as having a non-faulty engine, even though it in fact does have a fault. For these arbitrated vehicles, we are able to retroactively re-label the engine fault labels of the vehicle according to the reason behind its arbitration.

However, only a subset of arbitration-eligible vehicles are in fact arbitrated. This is due to the fact that arbitration claims are *voluntarily* submitted by vehicle buyers. Often a buyer will not notice a fault themselves, or instead fix the fault without submitting an arbitration claim. Therefore, there is a subset of faulty vehicles that were never arbitrated. In these vehicles, their engine fault labels are considered incorrect. As a result, there is an inherent label noise in our dataset, where a small set of clean vehicles will have some unlabeled engine fault. Further, there is also a small degree of inter-class label confusion, as some faults are difficult to distinguish from one another on various vehicles. We show in Section 5 that even with the presence of this inherent label noise, we are still able to train an accurate classifier for engine faults.

# 4 APPROACH

## 4.1 Problem Formulation

We formulate the task of automobile fault detection as a multi-label classification task. Given a representation of a vehicle $x$, we make a prediction $\hat{y} \in \{0,1\}^c$ where $\hat{y}$ is a vector denoting probabilities of $c$ different engine faults. Let $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^N$ denote a training dataset of size $N$ comprised of vehicle representations $x_{i...N}$ with their respective engine fault labels $y_{i...N}$. The task of multi-label classification is to train a classifier $f(x) = \hat{y}$ using $\mathcal{D}_t$ and a multi-label classification training loss $\mathcal{L}$ such that the aggregate loss $\sum_{i=1}^N \mathcal{L}(y_i, \hat{y}_i)$ is minimized, with the goal of $f(x)$ becoming an accurate classifier of engine faults.

We represent a vehicle as the composition of an audio and vibration signal, and tabular metadata. Therefore for every vehicle representation $x_i$, we have $x_{i_a} \in \mathbb{R}^{n_a}$, $x_{i_v} \in \mathbb{R}^{n_v}$, and $x_{i_t} \in \mathcal{T}^{n_t}$, where $x_{i_a}, x_{i_v}, x_{i_t}$ denote the audio signal, vibration signal, and tabular metadata respectively. $\mathcal{T}$ denotes the set of tabular metadata used, described in section 3.0.3. $n_a$, $n_v$, and $n_t$ denote the respective lengths of the audio and vibration signals, and the number of tabular metadata entries. The classification of a vehicle $x_i$'s engine faults can now be rewritten as $f(x_{i_a}, x_{i_v}, x_{i_t}) = \hat{y}_i$. We describe the construction of the classifier in detail in Section 4.3.

## 4.2 Dataset

*4.2.1 Dataset splits.* To evaluate the performance of AMPNet, we create three large-scale splits of our collected data into train, validation, and evaluation sets. Each of the three sets are split with the intent of having a natural distribution of types of vehicles sold in the United States. To achieve a natural distribution, we split each dataset according to time periods of vehicle sales. The validation and evaluation datasets contain all vehicles that were sold on the platform in two separate time periods. The train set contains a subset of all vehicles sold, excluding the time periods in the validation and evaluation sets. Table 1 shows the number of positive cases of the five engine fault classes in the three datasets, and in addition the number of samples that are considered non-faulty. The train dataset contains 45,275 vehicles across 846 different models. The validation and evaluation set have 59,150 and 52,440 vehicles across 942 and 946 different models, respectively.

**Table 1: Overview of engine fault class distribution across the train, validation, and evaluation datasets.**

| Class Counts | Train | Validation | Evaluation |
|---|---|---|---|
| IEN | 16,295 | 3,357 | 3,142 |
| RR | 3,979 | 2,259 | 2,228 |
| TC | 1,902 | 1,123 | 1,046 |
| ACC | 16,668 | 15,291 | 14,602 |
| EXH | 18,126 | 8,117 | 7,611 |
| No Faults (Negative) | 11,426 | 37,206 | 31,711 |

*4.2.2 Dataset preprocessing.* The collected audio recordings are two-channel signals between 25-35 seconds in duration. The vibration recordings are three-channel signals that have the same duration as their audio counterpart. We crop all audio and vibration signals to 30 seconds, with zero padding for samples shorter than 30 seconds. When processing the signals, we utilize audio in mono format with a sample rate of 22,050$Hz$ and 100$Hz$ for vibration. After the waveform signals are preprocessed, we generate spectrogram representations of each. For audio, we generate a log-scaled Mel-spectrogram using an FFT window of 1024 units, a stride of 512 units, and 256 Mel-frequency bins. For vibration, we generate a linearly-scaled spectrogram representation of each channel. The spectrograms are finally passed through log compression. Note that we do not use Mel-scale for vibration as it is designed to mimic human's non-linear perception of sound. At the very low captured frequencies of the vibration signals, the Mel-scale carries no significant meaning. We instead use a simple linear scale of 128 frequency bins, an FFT window of 256 units and a stride of 32 units.

The set of metadata for a particular vehicle contains 5 numerical entries and 6 categorical text entries, which are listed in Section 3.0.3. The 6 text entries are passed through a tokenization operation for each value independently, which count the occurrence of all possibilities of each entry. The tokenized vectors of each categorical entry are concatenated together along with the numerical entries, which is denoted by $x_{i_t}$. The vector $x_{i_t}$ is then element-wise normalized such that each value has a zero mean and unit standard deviation, calculated from the statistics across the training dataset.

For the formulation of AMPNet in Section 4.3, we define the normalized waveforms of audio and vibration of a vehicle $x_i$ as $x_{i_a}$ and $x_{i_v}$ respectively. We present the spectrogram constructions of the audio and vibration as functions defined as $\phi_a(x_{i_a})$ and $\phi_v(x_{i_v})$ respectively. We also present the metadata tokenization and normalization as a function defined as $\phi_t(x_{i_t})$.

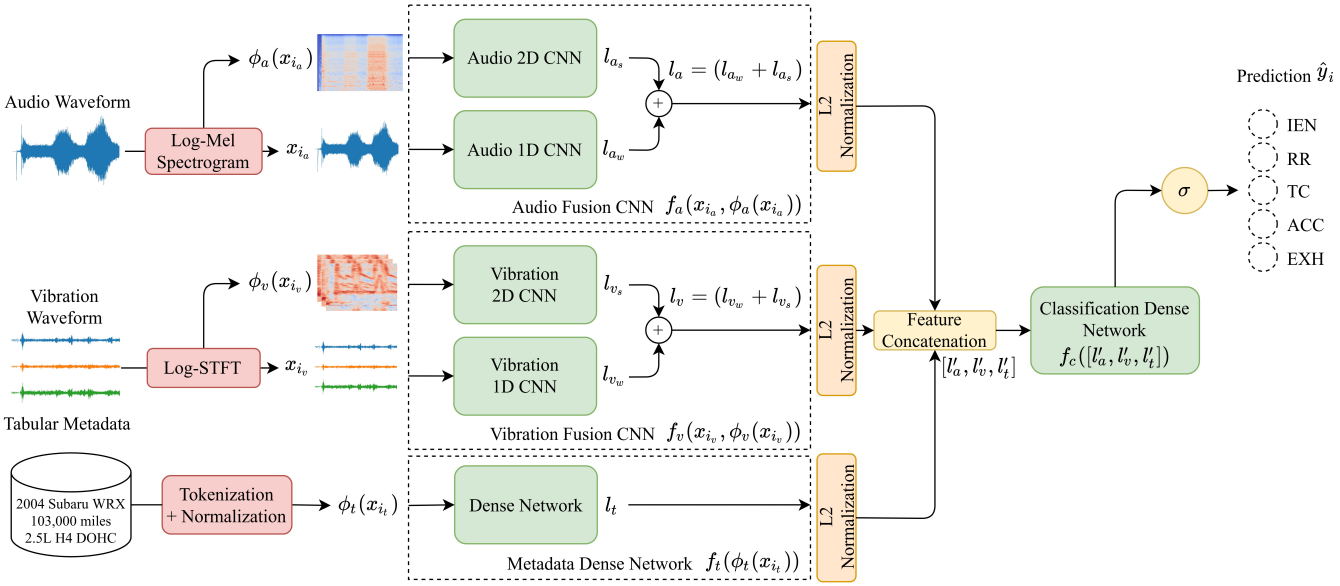


**Figure 3: Overview of *AMPNet*. Features from the audio, vibration, and metadata modalities are extracted independently, then concatenated and fused for final multi-label classification of the five engine faults.**

## 4.3 Proposed Model

Our model performs multi-label engine fault classification through the multi-modal feature fusion of features extracted from audio, vibration, and metadata. Figure 3 depicts the design of our model. We build upon works in literature that have shown that the fusion of multiple representations of the same signal improves classification performance on a variety of tasks [9, 15, 32]. For both the audio and vibration modalities, we extract features from their waveform and spectrogram representations simultaneously. The waveform and spectrogram features from each independent modality are initially fused, then the multi-modal features of the audio, vibration, and metadata are concatenated together for final classification.

The audio fusion CNN, $f_a$, extracts features from the audio waveform $x_{i_a}$ and Mel-spectrogram $\phi_a(x_{i_a})$ using two separate CNN networks. The network for the spectrogram is comprised of repeating blocks of a 2D convolution layer, batch normalization, LeakyReLU non-linear activation, and a max pooling layer. This block is repeated four times and the resulting feature response is global-average-pooled into a feature vector of 1024 units, denoted by $l_{a_s}$. Similarly, the waveform network is comprised of the same blocks, except that 1D convolutions are used. We also replace the first layer of the waveform network with learnable parameterized Sinc filters [24], which has been shown to be useful in multiple audio understanding tasks [9, 24]. The resulting feature vector of 1024 units from the waveform network is denoted as $l_{a_w}$. Once features from the audio waveform and spectrogram representations are extracted, they are fused together using element-wise summation, denoted by $l_a = (l_{a_w} + l_{a_s})$.

The construction of the vibration fusion CNN, $f_v$, follows a similar construction as $f_a$. Both the waveform and spectrogram networks are created with repeating blocks of 1D and 2D convolution layers, batch normalization, LeakyReLU activation, and max pooling, respectively. For the vibration waveform network, the first layer is a 1D convolution that takes the 3-channel waveform. The resulting 1024-unit feature vectors from both networks are element-wise summed, denoted by $l_v = (l_{v_w} + l_{v_s})$, where $l_{v_w}$ is the extracted features of the vibration waveform $x_{i_v}$ and $l_{v_s}$ is the extracted features of the vibration spectrogram $\phi_v(x_{i_v})$.

The metadata dense network, $f_t$, is a network constructed of linear layers to extract intermediate features from the tokenized metadata $\phi_t(x_{i_t})$. Specifically, $f_t$ is constructed of 2 repeating blocks of linear, LeakyReLU activation, batch normalization, and dropout layers. We denote the resulting metadata feature vector as $l_t$.

After obtaining $l_a$, $l_v$, and $l_t$, we $L_2$ normalize each vector, resulting in $l'_a$, $l'_v$, and $l'_t$. We perform $L_2$ normalization to scale each vector to the same range, which aids in the prevention of one modality's features overpowering another during fusion. We denote the $L_2$ normalization of a vector $z$ as $z' \rightarrow z/\|z\|_2$.

To perform the final classification, we construct a classification dense network, $f_c$, that takes the concatenated features from each modality $[l'_a, l'_v, l'_t]$ and outputs logits of each engine fault class. $f_c$ is constructed in a similar fashion as $f_t$, where it contains 2 repeating blocks of linear, LeakyReLU activation, batch normalization, and dropout layers followed by a final linear layer that outputs class-wise logits. Finally, the model outputs are passed through a sigmoid activation to project the outputs of $f_c$ into class-wise probabilities. We denote the sigmoid activation of a vector $z$ as $\sigma(z)$.

Combining each stage, we construct the final AMPNet model $f$:

$$f(x_{i_a}, x_{i_v}, x_{i_t}) = \sigma(f_c([l'_a, l'_v, l'_t])) \tag{1}$$

$$= \sigma(f_c[f_a(x_{i_a}, \phi_a(x_{i_a}))', f_v(x_{i_v}, \phi_v(x_{i_v}))', f_t(\phi_t(x_{i_t}))']) \tag{2}$$

**Table 2: Summary of engine fault detection performance of each component of AMPNet.**

| | Macro-Average | | Class-Wise ROC AUC | | | | | Class-Wise AP | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | mROC | mAP | IEN | RR | TC | ACC | EXH | IEN | RR | TC | ACC | EXH |
| Audio Only | 0.716 | 0.269 | 0.796 | 0.700 | 0.690 | 0.655 | 0.740 | 0.367 | 0.103 | 0.050 | 0.431 | 0.393 |
| Vibration Only | 0.627 | 0.188 | 0.600 | 0.734 | 0.603 | 0.591 | 0.608 | 0.101 | 0.254 | 0.031 | 0.349 | 0.204 |
| Audio+Vibration | 0.741 | 0.313 | 0.806 | 0.773 | 0.699 | 0.668 | 0.758 | 0.376 | 0.268 | 0.057 | 0.443 | 0.419 |
| Metadata Only | 0.806 | 0.336 | 0.749 | 0.864 | 0.948 | 0.703 | 0.767 | 0.148 | 0.288 | 0.490 | 0.429 | 0.326 |
| Audio+Vibration+Metadata | **0.844** | **0.454** | **0.853** | **0.877** | **0.951** | **0.729** | **0.812** | **0.410** | **0.383** | **0.515** | **0.489** | **0.476** |

We train the model using the training configuration described in Section 4.4 paired with binary cross-entropy loss:

$$\mathcal{L}(y_i, \hat{y}_i) = -\frac{1}{c}\sum_{i=1}^{c} y_i \log \hat{y}_i + (1 - y_i)\log\,(1 - \hat{y}_i) \tag{3}$$

Further details about the exact construction of the feature extractors and classification network are described in Appendix A.3.

## 4.4 Training Configuration

All models are trained with the SGD optimizer with a learning rate scheduled by the 1cycle policy described in [27]. The 1cycle policy starts the optimizer's learning rate at a small value, then anneals it to a maximum learning rate $\lambda_{max}$, and subsequently anneals it back to a small value over the entire training procedure. We also follow a learning rate range test introduced by [27] to automatically find the $\lambda_{max}$ parameter for the learning rate scheduler. Appendix A.1.1 describes in detail the range test that is used. During training, all models are trained for 20 epochs with a batch size of 16.

*4.4.1 Data augmentations.* During training, we perform data augmentations on both the audio and vibration modalities. For both audio and vibration, we perform random time shifting which randomly shifts the audio and vibration representations forwards and backwards along the time axis. The samples that are randomly shifted are rolled over. For example, if the representation is shifted $k$ samples forward, the last $k$ samples are rolled to the beginning of the representation. We perform time shifting on both the waveform and spectrogram representations of both audio and vibration. We also perform random shuffling of the channels of both the vibration waveform and spectrograms. The waveforms and spectrograms are shuffled independently, such that the x-y-z orientation of the waveforms may not necessarily align with the spectrograms for a given sample. Since we have no knowledge of the accelerometer orientation in relation to a vehicle for a given sample, we perform this augmentation to aid the model in becoming invariant to the orientation. Similarly we perform random time shifting to improve invariance towards the variations in the unconstrained nature of audio recordings, explained in Section 3.2.

## 4.5 Evaluation Protocol

The main goal of deploying AMPNet is to reduce engine-related arbitration claims across the auction platform. However, as explained in Section 3.2, only a subset of arbitration-eligible vehicles are in fact arbitrated. As a result, using the number of arbitrations caught in a historical month is a noisy and inaccurate measure of engine fault detection models. Instead, we use receiver operating characteristic area-under-curve (ROC AUC) and average precision (AP) scores as *proxies* for arbitrations. The stronger classifier (high ROC and AP scores) will inherently catch more arbitration-eligible vehicles. An arbitrated vehicle is equivalent to a vehicle with a positive engine fault class, which is inherently captured in the ROC and AP metrics. ROC AUC is defined as the calculated area under the ROC curve, which is a plotted curve of the true positive rate against the false positive rate of a binary classifier. AP is defined as the area under the precision-recall (PR) curve, which is created by plotting the precision versus recall of a binary classifier at various thresholds. In our experiments, we show ROC AUC and AP scores for each engine fault class. We further calculate the macro-averaged scores of ROC AUC and AP across each class. The calculated macro-averaged ROC AUC score of $c$ classes, denoted by $mROC$, is $1/c \sum_{i=1}^{c} ROC_{AUC}(c_i)$. Similarly the macro-averaged AP score, denoted by $mAP$ is $1/c \sum_{i=1}^{c} PR_{AUC}(c_i)$. We use these metrics to quantify engine fault detection performance.

# 5 EXPERIMENTS

Table 2 shows the engine fault detection performance when incrementally adding each component of our described model to investigate each component's respective contribution to performance. Each method depicted in Table 2 is constructed by removing the other respective feature extractors, while using the same dense classification network $f_c$. For example, the audio only model rewrites (2) to be $f(x_{i_a}) = \sigma(f_c([l'_a]))$. Similarly the vibration only and metadata only models are rewritten as $f(x_{i_v}) = \sigma(f_c([l'_v]))$ and $f(x_{i_t}) = \sigma(f_c([l'_t]))$ respectively. The audio and vibration fusion model is rewritten as $f(x_{i_a}, x_{i_v}) = \sigma(f_c([l'_a, l'_v]))$.

We see that the fusion of audio, vibration, and metadata features achieved a performance of 0.844 mROC and 0.454 mAP, significantly outperforming any individual components both in terms of mROC and mAP. Looking at each individual component's class-wise performance, we notice that certain modalities become strong classifiers on certain engine fault classes over others. For example, the audio modality is significantly better than any other single modality for capturing IEN, while the vibration modality outperforms audio in capturing RR. We infer that the disparity in performance is because IEN is often diagnosed through audible tapping, while RR often presents itself as a shaking and vibrating engine that isn't necessarily audible. When fusing audio and vibration features, we see that the IEN and RR performance outperforms any individual modality. It is interesting to note that although one modality is able to capture more significant features than the other for a specific engine fault, fusing them together still provides complementary features that improves detection performance.

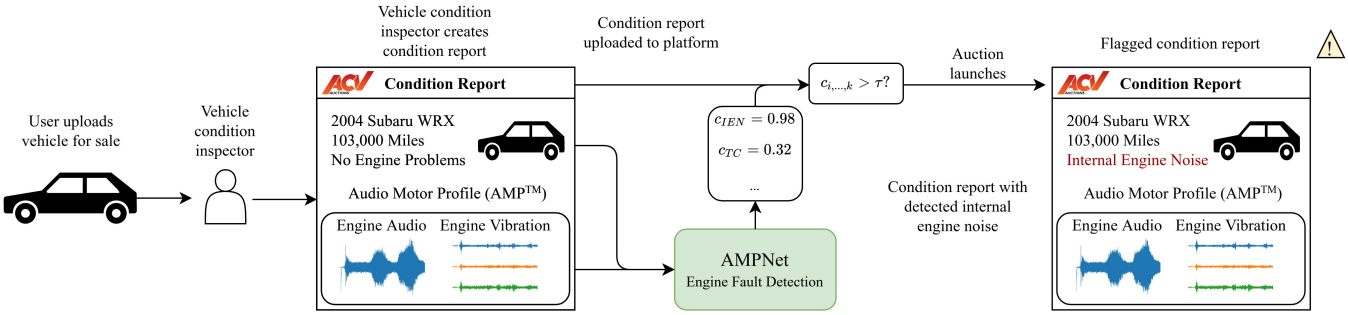


**Figure 4: Overview of our deployed pipeline. Vehicle condition reports are passed into AMPNet for a multi-label classification of possible engine faults. If certain classes of faults detected exceed a threshold $\tau$, the condition report is flagged with the fault and undergoes secondary human review before the vehicle auction is launched.**

**Table 3: Comparison of AMPNet audio classification network against several baseline models.**

| Model Comparison | mROC | mAP | # Params |
|---|---|---|---|
| CNN14 [15] | 0.713 | 0.265 | 79.7M |
| AST [11] | 0.662 | 0.227 | 87.4M |
| Wavegram-Logmel-CNN14 [15] | **0.718** | **0.269** | 80.2M |
| Our Model (Audio Only) | 0.716 | **0.269** | **2.4M** |

**Table 4: Relative vehicle engine arbitration amounts on the ACV Auctions platform of various time periods.**

| Time Period | AMPNet Status | Relative Arbitration Amount |
|---|---|---|
| Jan. 2021 - April 2021 | OFF | 0.00% |
| May 2021 - June 2021 | ON | **-20.85%** |
| July 2021 - Sept. 2021 | OFF | -1.15% |

We also see that the audio and vibration modalities perform poorly on the detection of TC, while the metadata information significantly outperforms them. We infer that because timing chain issues occur only at the vehicle start for a very short duration, they are difficult to detect in the recorded signals. Vehicle engines with timing chains also often have diagnostic sensors that can detect these faults, which are captured in the metadata of the vehicle. However, we see that the fusion of all three modalities still improves TC performance over metadata alone, meaning there are still complementary features being learned in the audio and vibration modalities. Further we see that adding metadata information to audio and vibration improves detection performance across all classes, from which we infer that the information captured in a vehicle's metadata helps uncover various biases towards each of the engine faults that significantly improve performance. While the training dataset discussed and used in this paper is a subset of all available training data, training on larger collections of vehicles further increase ROC and AP performance across all engine faults.

## 5.1 Audio Network Comparison

We also compare our waveform-spectrogram fusion network design against other well-known audio classification networks in literature to show its relative effectiveness for our task of engine fault detection. Specifically, we compare against the Audio Spectrogram Transformer [11] and CNN14 [15] that utilize the audio spectrogram. Additionally, we compare against Wavegram-Logmel-CNN14 [15], a network that similarly performs fusion of features from the waveform and spectrogram representations of audio. The networks in [11, 15] have previously shown state of the art performance on large scale audio classification datasets such as Audioset [10]. As shown in Table 3, we see that our audio classification network outperforms the various methods and is comparable to the Wavegram-Logmel-CNN14 network [15]. In addition, our audio classification network has significantly fewer parameters compared to the models introduced in [11, 15]. Given the strong performance of our audio classification network design, we follow the same network strategy for extracting features from the vibration modality. Further details about the training configurations of each of these comparisons are found in Appendix A.4.

## 6 DEPLOYMENT CASE STUDY

We conduct an online A/B test of AMPNet by deploying it into ACV Auctions' live auction platform. As depicted in Figure 4, we pass all recorded condition reports of vehicles through AMPNet, where AMPNet predicts whether the vehicle's engine has any engine faults that have not been previously disclosed by the inspector who created the condition report. For deployment, we tune class-wise thresholds, denoted by $\tau_{i,...,c}$, such that if a predicted engine fault exceeds its class threshold, the vehicle is subsequently flagged with the corresponding engine fault. If a secondary human reviewer agrees with AMPNet, these faults are presented on the condition report when the vehicle is launched on the auction platform, shown in Figure 1. We tune the thresholds to favor very precise predictions, at the expense of recall, as we want to avoid falsely labeling a vehicle as having a faulty engine when in fact it is clean. We hypothesized that having AMPNet actively detect engine faults across all vehicles will reduce engine-related arbitration claims, as a smaller amount of vehicle condition reports will have missed engine faults (i.e. reducing the number of arbitration-eligible vehicles). Table 4 shows the relative percent of arbitrations of given time periods where AMPNet is active and inactive. Activating AMPNet for two months,

May and June 2021, resulted in a 20.85% reduction in total engine arbitrations compared to the previous time period. After these two months, we disabled AMPNet and saw that engine arbitrations increased to about the same level as the first time period. From this test we infer that AMPNet is able to significantly reduce the number of vehicle engine arbitrations across the auction platform.

## 7 CONCLUSIONS

We presented *AMPNet*, a large-scale engine fault detection pipeline for the automatic detection of vehicle engine faults. We described our process for collecting data from condition reports of vehicles recorded across the United States that is used to train and evaluate AMPNet. We presented the construction, training, and evaluation process of our models and further experimentally demonstrated that AMPNet is able to accurately capture engine faults agnostic of the type of vehicle. We further investigated the uses and quantified individual performance of multiple modalities of information for engine fault detection, specifically audio and vibration signals, and tabular metadata. We finally show the effects of deploying AMPNet into the ACV Auctions live auction platform, showing a significant 20.85% decline in engine-related arbitration claims across the platform. With AMPNet, we have the ability to significantly assist human inspectors to detect and list all engine faults of a vehicle and simultaneously improve the quality and consistency of condition reports of vehicles sold on the auction platform. With this work we show that automatic engine diagnosis is possible at scale, and we believe this work is a step in the direction of improving machine condition monitoring techniques.

## ACKNOWLEDGMENTS

This work was supported by the Center for Identification Technology Research (CITeR) and the National Science Foundation (NSF) under grant #1822190.

## REFERENCES

[1] Wail M Adaileh. 2013. Engine fault diagnosis using acoustic signals. In *Applied Mechanics and Materials*, Vol. 295. Trans Tech Publ.

[2] Ali I Alahmer, Wail M Adaileh, and Mohammad A Al Zubi. 2014. Monitoring of a spark ignition engine malfunctions using acoustic signal technique. *International Journal of Vehicle Noise and Vibration* 10, 3 (2014), 201–213.

[3] Sabry Allam, Mohammed Abdo, and M Rabie. 2018. Diesel engine fault detection using vibration and acoustic emission signals. *Diesel Engine* (2018).

[4] Charlie Campanella, Keith Carolus, Reid Gershbein, Daniel Magnuszewski, Michael Pokora, Timothy Poulsen, Philip Schneider, and Dennis Christopher Fedorishin. 2020. Vehicle audio capture and diagnostics. US Patent App. 16/749,585.

[5] Mark Cartwright, Ana Elisa Mendez Mendez, Jason Cramer, Vincent Lostanlen, Graham Dove, Ho-Hsiang Wu, Justin Salamon, Oded Nov, and Juan Bello. 2019. SONYC Urban Sound Tagging (SONYC-UST): A Multilabel Dataset from an Urban Acoustic Sensor Network. In *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*. 35–39.

[6] Keunwoo Choi, George Fazekas, and Mark Sandler. 2016. Automatic tagging using deep convolutional neural networks. *arXiv preprint arXiv:1606.00298* (2016).

[7] Simone Delvecchio, Paolo Bonfiglio, and Francesco Pompoli. 2018. Vibro-acoustic condition monitoring of Internal Combustion Engines: A critical review of existing techniques. *Mechanical Systems and Signal Processing* 99 (2018), 661–683.

[8] A.J. Eronen, V.T. Peltonen, J.T. Tuomi, A.P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi. 2006. Audio-based context recognition. *IEEE Transactions on Audio, Speech, and Language Processing* 14, 1 (2006), 321–329. https://doi.org/10.1109/TSA.2005.854103

[9] Dennis Fedorishin, Nishant Sankaran, Deen D Mohan, Justas Birgiolas, Philip Schneider, Srirangaraj Setlur, and Venu Govindaraju. 2021. Waveforms and Spectrograms: Enhancing Acoustic Scene Classification Using Multimodal Feature Fusion. In *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*. Barcelona, Spain, 216–220.

[10] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 776–780.

[11] Yuan Gong, Yu-An Chung, and James Glass. 2021. AST: Audio Spectrogram Transformer. *arXiv preprint arXiv:2104.01778* (2021).

[12] Patricia Henriquez, Jesus B Alonso, Miguel A Ferrer, and Carlos M Travieso. 2013. Review of automatic fault diagnosis systems using audio and vibration signals. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44, 5 (2013), 642–652.

[13] Prathap V Jayasooriya, Geethal C Siriwardana, and Tharaka R Bandara. 2021. Vibration analysis to detect and locate engine misfires. In *2021 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, Vol. 4. IEEE, 237–243.

[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. 2020. PANNs: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), 2880–2894.

[16] Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. 2017. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *arXiv preprint arXiv:1703.01789* (2017).

[17] Steve Koshy Mathew and Yu Zhang. 2020. Acoustic-based engine fault diagnosis using WPT, PCA and Bayesian optimization. *Applied Sciences* 10, 19 (2020), 6890.

[18] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. 2018. A multi-device dataset for urban acoustic scene classification. *arXiv preprint arXiv:1807.09840* (2018).

[19] Ahmed F Mofleh, Ahmed N Shmroukh, and Nouby M Ghazaly. 2020. Fault detection and classification of spark ignition engine based on acoustic signals and artificial neural network. *International Journal of Mechanical and Production Engineering Research and Development* 10, 3 (2020), 5571–5578.

[20] Dayong Ning, Jiaoyi Hou, Yongjun Gong, Zengmeng Zhang, and Changle Sun. 2016. Auto-identification of engine fault acoustic signal through inverse trigonometric instantaneous frequency analysis. *Advances in Mechanical Engineering* 8, 3 (2016), 1687814016641840.

[21] Kamalesh Palanisamy, Dipika Singhania, and Angela Yao. 2020. Rethinking cnn models for audio classification. *arXiv preprint arXiv:2007.11154* (2020).

[22] Karol J Piczak. 2015. Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th international workshop on machine learning for signal processing (MLSP)*. IEEE, 1–6.

[23] Jordi Pons, Oriol Nieto, Matthew Prockup, Erik Schmidt, Andreas Ehmann, and Xavier Serra. 2017. End-to-end learning for music audio tagging at scale. *arXiv preprint arXiv:1711.02520* (2017).

[24] Mirco Ravanelli and Yoshua Bengio. 2018. Speaker recognition from raw waveform with sincnet. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 1021–1028.

[25] G Sakthivel, R Jagadeeshwaran, D SaravanaKumar, et al. 2020. Condition Monitoring of a IC Engine Fault Diagnosis using Machine Learning and Neural Network Techniques. In *2020 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS)*. IEEE, 183–189.

[26] Syed Maaz Shahid, Sunghoon Ko, and Sungoh Kwon. 2021. Real-time abnormality detection and classification in diesel engine operations with convolutional neural network. *Expert Systems with Applications* (2021), 116233.

[27] Leslie N Smith and Nicholay Topin. 2019. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, Vol. 11006. International Society for Optics and Photonics, 1100612.

[28] Jianfeng Tao, Chengjin Qin, Weixing Li, and Chengliang Liu. 2019. Intelligent fault diagnosis of diesel engines via extreme gradient boosting and high-accuracy time–frequency information of vibration signals. *Sensors* 19, 15 (2019), 3280.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[30] Wenkui Xi, Zhixiong Li, Zhe Tian, and Zhihe Duan. 2018. A feature extraction and visualization method for fault detection of marine diesel engines. *Measurement* 116 (2018), 429–437.

[31] Ruili Zeng, Lingling Zhang, Jianmin Mei, Hong Shen, and Huimin Zhao. 2017. Fault detection in an engine by fusing information from multivibration sensors. *International Journal of Distributed Sensor Networks* 13, 7 (2017), 1550147717719057.

[32] Boqing Zhu, Changjian Wang, Feng Liu, Jin Lei, Zhen Huang, Yuxing Peng, and Fei Li. 2018. Learning environmental sounds with multi-scale convolutional neural network. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

# A APPENDIX

## A.1 Training Configuration

*A.1.1 Learning rate range test.* We construct a modified version of the learning rate range test introduced in [27] where $n$ learning rate values $\lambda_1, \lambda_2, ..., \lambda_n$ are uniformly sampled and used within a single forward pass of the model to find the learning rate which produces the lowest batch-wise loss. The lowest batch-wise loss learning rate, $\lambda_k$, divided by a factor of 10 is selected as the optimal learning rate. We further run this test $m$ times and take the median $\lambda_k$ to account for any outliers in selected learning rates due to batch stochasticity. The final $\lambda_{max}$ is calculated from:

$$\lambda_{max} = \frac{median(\lambda_{k_1}, \lambda_{k_2}, ..., \lambda_{k_m})}{10} \tag{4}$$

We found that using the 1cycle learning rate policy paired with the above described learning rate range test performs well across our engine fault detection models. For example, the $\lambda_{max}$ calculated for the model described in Figure 3 is 0.027.

After each epoch of training, we validate the current model against the validation set. At the end of the training sequence, we evaluate the trained model on the evaluation set at the checkpoint where the model achieved the highest macro-averaged average precision score on the validation set.

*A.1.2 Data augmentations.* As previously mentioned in Section 4.4.1, we perform time shifting of the waveform and spectrogram representations of both the audio and vibration. For the audio and vibration waveform, we randomly time shift up to 95% of the size of each respective waveform. For the audio spectrogram, we select the shifting factor randomly from a normal distribution with a mean of 100 samples and standard devation of 400 samples. For the vibration spectrogram, the shifting factor is also sampled from a normal distribution with a mean of 10 samples and standard deviation of 40 samples. Note that each of the shifting factors are sampled independently such that the waveforms and spectrograms are shifted by varying degrees, meaning that they are no longer temporally aligned.

## A.2 Dataset Preprocessing

Table 5 describes the parameters of the audio and vibration spectrogram construction, in addition to the final shapes of each modality that AMPNet consumes. Both the audio and vibration waveforms are normalized to the range $(-1, 1)$. Similarly the audio and vibration spectrograms are normalized using Z-score normalization such that each spectrogram has zero mean and unit standard deviation.

## A.3 Proposed Model

Tables 6, 7, and 8 show the detailed construction of the spectrogram, waveform, and metadata feature extractor networks that are defined in Section 4.3, respectively. For the Sinc layer in the audio waveform network, we use the official implementation of SincNet [1] introduced by [24]. Further, Table 9 shows the detailed construction of the final classification network that takes the extracted multi-modal features and performs multi-label classification of the five engine faults. The complete AMPNet model, illustrated in Figure 3, has a total of 4.7 million parameters. The training time of AMPNet on an RTX6000 GPU for 20 epochs with a batch size of 16 takes 8 hours to complete.

[1] https://github.com/mravanelli/SincNet

**Table 5: Description of the inputs to AMPNet.**

| Modality | FFT Window | Stride | Frequency Bins | Shape |
|---|---|---|---|---|
| Audio Spectrogram | 1024 | 512 | 256 | [1, 256, 1292] |
| Vibration Spectrogram | 256 | 32 | 128 | [3, 128, 94] |
| Audio Waveform | | | | [1, 661500] |
| Vibration Waveform | | | | [3, 3000] |
| Tokenized Metadata | | | | [1, 82] |

## A.4 Audio Network Comparison

*A.4.1 CNN14 [15].* We use the official implementation of CNN14 [2]. For the audio spectrogram input, we follow the same procedure in Section 4.2.2. The network is trained using the training procedure described in [15], specifically using the Adam [14] optimizer with a learning rate of 0.001.

*A.4.2 AST [11].* We use the official implementation of the Audio Spectrogram Transformer (AST) [3]. We use a stride of 20 samples in both the time and frequency dimension for splitting the input spectrogram into patches. We construct the input spectrograms using the procedure in Section 4.2.2 and follow the training procedure of AST in [11].

*A.4.3 Wavegram-Logmel-CNN14 [15].* Similarly to CNN14, we use the official implementation of Wavegram-Logmel-CNN14[2] and the training procedure described in [15]. However, to use the default structure of the network, we utilize the spectrogram construction parameters used in [15]. Specifically, we use an FFT window size of 2048 samples, stride of 320 samples, and 128 Mel frequency bins.

Note that we experimented with multiple training configurations of different optimizers and learning rates, including the configuration used for AMPNet to find each network's best performance. The above described training configurations were found to perform the best for each respective network.

[2] https://github.com/qiuqiangkong/audioset_tagging_cnn
[3] https://github.com/YuanGongND/ast

**Table 6: Construction of the audio and vibration spectrogram feature extractor networks.**

| Component | Channel Input (Audio / Vibration) | Channel Response (Audio / Vibration) | Kernel Size | Padding | Stride | Input (Audio / Vibration) | Output (Audio / Vibration) |
|---|---|---|---|---|---|---|---|
| **Conv Block 1** | 1 / 3 | 32 | | | | $\phi_a(x_{i_a})/\phi_v(x_{i_v})$ | |
| 2D Convolution | | 32 | 3x3 | 1x1 | 1x1 | | |
| Batch Normalization | | 32 | | | | | |
| LeakyReLU Activation | | | | | | | |
| Max Pooling | | 32 | 2x2 | 0 | 2x2 | | |
| **Conv Block 2** | 32 | 64 | | | | | |
| **Conv Block 3** | 64 | 128 | | | | | |
| **Conv Block 4** | 128 | 256 | | | | | |
| **Conv Block 5** | 256 | 256 | | | | | |
| **Global Average Pooling** | | | | | | | $l_{a_s}/l_{v_s}$ |

**Table 7: Construction of the audio and vibration waveform feature extractor networks.**

| Component (Audio / Vibration) | Channel Input (Audio / Vibration) | Channel Response | Kernel Size (Audio / Vibration) | Padding | Stride (Audio / Vibration) | Input (Audio / Vibration) | Output (Audio / Vibration) |
|---|---|---|---|---|---|---|---|
| ***Conv Block 1*** | 1 / 3 | 32 | | | | $x_{i_a}/x_{i_v}$ | |
| Sinc Layer / 1D Convolution | 1 / 3 | 32 | 251 | 125 | 1 | | |
| Batch Normalization | 32 | 32 | | | | | |
| LeakyReLU Activation | | | | | | | |
| Max Pooling | 32 | 32 | 8 / 3 | 0 | 8 / 3 | | |
| **Conv Block 2** | 32 | 64 | | | | | |
| 1D Convolution | 32 | 64 | 7 | 3 | 1 | | |
| Batch Normalization | 64 | 64 | | | | | |
| LeakyReLU Activation | | | | | | | |
| Max Pooling | 64 | 64 | 8 / 3 | 0 | 8 / 3 | | |
| **Conv Block 3** | 64 | 128 | | | | | |
| **Conv Block 4** | 128 | 256 | | | | | |
| **Conv Block 5** | 256 | 256 | | | | | |
| **Global Average Pooling** | | | | | | | $l_{a_w}/l_{v_w}$ |

**Table 8: Construction of the tabular metadata feature extractor network.**

| Component | Feature Input | Feature Output | Dropout $p$ | Input | Output |
|---|---|---|---|---|---|
| **Dense Block 1** | 82 | 226 | | $\phi_t(x_{i_t})$ | |
| Linear Layer | 82 | 226 | | | |
| LeakyReLU Activation | | | | | |
| Batch Normalization | 226 | 226 | | | |
| Dropout | | | 0.3 | | |
| **Dense Block 2** | 226 | 36 | | | $l_t$ |

**Table 9: Construction of the final engine fault classification network.**

| Component | Feature Input | Feature Output | Dropout $p$ | Input | Output |
|---|---|---|---|---|---|
| **Dense Block 1** | 2084 | 512 | | $[l'_a, l'_v, l'_t]$ | |
| Linear Layer | 2084 | 512 | | | |
| LeakyReLU Activation | | | | | |
| Batch Normalization | 512 | 512 | | | |
| Dropout | | | 0.3 | | |
| **Dense Block 2** | 512 | 256 | | | |
| **Classification Linear Layer** | 256 | 5 | | | $\hat{y}_i$ |