FAULT DIAGNOSIS OF ENGINE KNOCKING USING DEEP LEARNING NEURAL NETWORKS WITH ACOUSTIC INPUT PROCESSING

by

Muzammil Ahmed Shaik

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Electrical and Computer Engineering

Department of Electrical and Computer Engineering

Hammond, Indiana

December 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Lizhe Tan, Chair

Department of Electrical and Computer Engineering

Dr. Quamar Niyaz

Department of Electrical and Computer Engineering

Dr. Khair Al Shamaileh

Department of Electrical and Computer Engineering

Approved by:

Dr. Lizhe Tan

ACKNOWLEDGMENTS

I want to take this opportunity to thank Dr. Lizhe Tan, chair of the committee, and Dr. Colin P. Elkin for their constant encouragement and counsel throughout the exploration cycle. Dr. Colin P. Elkin has offered me multiple areas to begin my research, as well as the chance to seek other projects without objection, and has passed me over to our chair, Dr. Lizhe Tan, to continue with the ideal project. With his scientific achievements and expertise, Dr. Lizhe Tan has guided me to the completion of the project by offering informative discussions about it.

I would like to thank Dr. Quamar Niyaz, a committee member who was always available to assist me in any situation, for his helpful career advice and suggestions in his area of expertise. I appreciate Dr. Khari Al Shamaileh's recommendations for my research work, as well as his advice that will help me not only in my profession but also in the future.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CNN	Convolutional neural network
FFT	Fast Fourier transform
FT	Fourier transform
STFT	Short-time Fourier transform
MFCCs	Mel-frequency cepstral coefficients
FDD	Fault detection and diagnosis
VHM	Vehicle health management
ANN	Artificial Neural Networks
ML	Machine learning
DL	Deep learning
GA	Genetic algorithms
WT	Wavelet transform
SVM	Supported vector machine
CSV	Comma separated value
KLT	Karhunen-Loeve transform
DCT	Discrete cosine transform
NLP	Natural language processing
GDM	Gradient descent method

ABSTRACT

The engine is the heart of the vehicle; any problems with this component will cause significant damage and may even result in the car being junked. The engine repair cost is enormous, and there is no guarantee that the existing engine will be repaired or replaced. Fault diagnosis in engines is critical; there have been numerous techniques and tools used for fault diagnosis in this revolutionary world, which require some extra cost to detect and still cannot detect faults such as knocking. The engine can have several problems but knocking is the major issue that blows up the engine and results in the breakdown of the vehicle. Our research focuses on this key issue which not only costs thousands of dollars but also results in waste. According to experts, at a very early stage, knocking can be detected by human senses, either visually or audibly. The most noticeable feature in detecting engine faults is the knocking sound. Artificial intelligence deep learning neural networks are well known for their ability to simulate humans; we can utilize this domain to train the networks on sound to detect engine knocking. Many neural networks have been designed for various purposes, one of which is classification. The best widely used and reliable network is the convolution neural network (CNN) which takes input as images and classifies them respectively. Engine sounds have been collected from Google's Machine Perception research. Our research shows that a prominent feature in building these networks is data. Understanding data and making the most of it is central to data science. A better model is created by meaningful data, not just by designing a complex network. We have used a new algorithmic method of extracting sound and feeding it into all variants of CNN, which we call dependent vehicle sound extraction, in which we use fast Fourier transform (FFT), short-time Fourier transform (STFT), and Mel-frequency cepstral coefficients (MFCCs) for processing input sound signals. We validated the utilization of deep learning networks with a unique dependent vehicle feature extraction technique to detect engine knocking with accurate classification.

1. INTRODUCTION

1.1 Literature Review

The process of finding a flaw in any machine before it leads to a breakdown is known as fault detection. Fault detection and classification are crucial elements of fault diagnosis. Before determining which type of fault exists, we must first determine whether a fault exists. Fault detection can be viewed as a binary classification problem to some extent [1]. Fault detection is an extensive field that can be applied to any industry. For instance, ArcelorMittal, a top steel tycoon, demonstrates the importance and use of fault detection in rolling machines using artificial intelligence concepts in an interview with Carlos Alba (chief digital officer at ArcelorMittal) and Peter D'haese (chief digital officer at ArcelorMittal Flat Europe) [2].

Classification is the finest method for detecting faults. There is currently a sizable collection of meta-classifiers built around specialized algorithms to match classification schemes [3]. C-MAPSS is a program that simulates a large commercial turbofan engine. The software is written in MATLAB and Simulink. This software generates final simulated results in labeling format, which can be used to classify various faults in an aircraft engine using deep learning classification algorithms [4]. Knock is a phenomenon in which the leftover "end gases" explode unexpectedly instead of consuming flawlessly in an extending wavefront across the chamber. Within the cylinder, increased temperature, stresses, and audible reverberations embrace the subsequent shock wave. These effects can be very damaging at high speeds and loads, yet, when not essential, knocking can be disturbing to the driver and is related to higher NOx emissions and lower force creation. [5]. Engine knocking can also be detected using fault detection and diagnosis (FDD) deep learning classification algorithms. Top-notch automotive industries have demonstrated their methods for detecting faults in vehicle engines, and some of them are still in development:

a. Infosys: Vehicle Maintenance Workbench, it is an integrated platform based on AI and ML optimization algorithms for fleet maintenance efficiency and safety [6].
b. Intuceo: Predictive Maintenance Solutions, it gathers data from every in-vehicle sensor and uses machine-learning algorithms to offer preventative maintenance options [7].

c. Questar: AI-based Predictive Analytics to keep fleets on the road, it offers a vehicle health management (VHM) platform that uses in-vehicle data collection and AI methods to produce early warnings of probable vehicle faults [8].
d. IBM: Connected Vehicle Predictive Maintenance Solution, artificial intelligence investigates and follows up on the information from the sensors and cameras by giving suggestions to the driver [9].
e. BMW: Predictive Maintenance Control Measures, this innovation uses sensors to filter the ongoing status/health of the parts, data analytics and other machine learning algorithms to forecast failures before they occur [10].
f. Ford: Connected Vehicle Data Collaboration, with drivers' permission, it gives third-party businesses safe and secure access to vehicle-generated data to develop innovative, specialized goods and services like usage-based insurance, predictive maintenance, and smart roadside recovery [11].
g. Hyundai Motor Group: Sound-based Fault Diagnosis and Predictive Maintenance, devised a creative method that allows artificial intelligence to pick up on automotive sounds, enabling it to identify damaged components. [12].

These industries focus on sensor data and rely on complex, time-consuming machine learning and deep learning architectures. Organizations such as Intuceo, Ford, and Hyundai Motor Group are working on this area of research and are still in the development stage. The estimation of combustion noise based on in-cylinder pressure measurements is used to identify engine knocking and this noise is detected by sensors [13]. Using these sensors there is wavelet transform method to detect knock [14]-[16]. Another research uses a microphone to capture engine knock on a specific vehicle and used that vehicle to capture knock [17]. Even though mechanical advances in the exact inspection of knock vibrancy are expanding because of refinements in signal handling, automakers are working intensely to enhance knock recognition. However, no method can completely switch the human hearing in recognizing among severe knocks and mechanistic blows. Conceivably innovative auditory sensors can develop knock detection significantly [18]. After the assortment of information from sensors, numerous strategies like Artificial neural networks (ANN), machine learning (ML), deep learning (DL), genetic algorithms (GA), wavelet transform (WT), Fuzzy logic, supported vector machine (SVM), and some statistical methods are utilized to identify knock which again depends on sensors [19]-[21]. Our research focuses on data and uses unique

dependent vehicle feature extraction techniques to transform the input signal to a meaningful form that can be pipelined to a deep learning network which is simple in structure and produces high accuracy in fault detection. Fast Fourier transform, short-term Fourier transform, and Mel-frequency cepstral coefficients techniques have been deployed in our research to transform raw input sound to feasible input to a deep learning network. We have used Artificial neural networks, 1D convolution neural network, and 2D convolution neural network to obtain a highly accurate fault-diagnosing engine system.

1.2 Research and Motivation

Sound can be utilized to identify engine knocking; currently, sensor data is the only way to do so. Formula 1 is the top level of competition for single-seater formula racing vehicles in worldwide competitions amongst the automotive companies, that use cutting-edge technology to demonstrate their prowess in the field. The Mercedes car at the 2016 Formula 1 Malaysian Grand Prix suffered an engine knock, and it has been reported that the engine's big-end bearing failed without being detected by the existing sensor architecture [22]. Our research emphasizes sound data that can accurately detect knocking and would be valuable in the automotive industry for fault detection and diagnosis. Building complex deep learning architectures will not result in an efficient method of detecting faults; instead, focus on data.

1.3 Organization of Thesis

The structure of this thesis is as follows. The dataset utilized in the experiment and the idea of engine knock detection using audio signal as input are introduced in Chapter 2. Three techniques for preparing input sound are described in Chapter 3. The widely utilized artificial neural network algorithm is used in Chapter 4. The outcomes of each combination are evaluated in Chapter 5, which presents several fault detection algorithms based on two-dimensional convolutional neural networks combined with two audio signal processing techniques. The outcomes of engine knock detection using a one-dimensional convolutional neural network are examined and shown in Chapter 6. The observations and future work are presented in Chapter 7.

1.4 Contribution of Thesis

a. In this thesis work, the artificial neural network method is investigated and applied in engine knock detection as the baseline.
b. Throughout the evaluation of deep learning neural networks with input processing for engine knock diagnosis with multiple audio labels, the most accurate classification method (2D STFT-CNN and 2D MFCC-CNN) is validated.
c. The feasibility of engine knock classification using a one-dimensional convolutional neural network is explored in order to reduce computational complexity.

2. ENGINE DATASET

An artificial neural network method is built on data. A good audio collection is required to develop effective ANN and deep learning (DL) algorithms for engine knock detection. Collecting engine sounds from various vehicles would take years to complete one set, which could be imbalanced due to a lack of knocking samples. There are few resources for collecting engine sound samples, and no automotive industry has an open-source collection of sounds from their vehicles to detect knock because they do not focus on sound. However, Google provides us with YouTube's car engine sound videos, timeframe, and the id to download the audio sample.

2.1 Audio Set Dataset

The Audio Set is a large dataset of manually annotated audio events that aims to bridge the data availability gap between image and audio research. They have collected information from individual person who have labelled audio files to investigate the presence of definite acoustic modules in 10-second cuts of YouTube tapes using a wisely coordinated characterized phenomenologically of 632 sound classes thoroughly categorized by the human synthesis and research. Labeling segments are proposed using object, framework, and subject matter evaluation searches. The Audio Set YouTube Corpus comprises named YouTube sections, organized as a comma separated value (CSV) document including YouTube identifiers, begin time, end time, and at least one label. Audio set portions are each 10 seconds in length except for unseen video which exceeds the duration of the actual video. Each audio file conveys at least one class name [23].

```
# YTID, start_seconds, end_seconds, positive_labels

-0RWZT-miFs, 420.000, 430.000, "/m/03v3yw,/m/0k4j"
```

Figure 2-1 CSV document of Audio Set [23].

Annotators confirmed the presence of sound classes /m/03v3yw ("Keys jangling") and /m/0k4j ("Car") in the YouTube video -0RWZT-miFs for the 10-second chunk from t=420 sec to t=430 sec which is illustrated in Figure 2-1.

2.2 The Audio Set Ontology of Engine

The sound of a machine generates mechanical energy. Combustion engines use fuel to generate heat, which in turn generates force. Electric motors are devices that convert electrical energy into mechanical motion. Pneumatic motors and clockwork motors are two other types of engines.

The Audio Set Ontology demonstrated in Figure 2-2 is a structured set of audio event categories. The provided categories are an extensive collection of terminology that may be utilized to identify audio events occurring in real-time soundtracks. A specific acoustic tape falls into the category that best fits the notion or comprehension that the listener has after hearing it. The hierarchical structure makes it as simple as feasible for individuals to choose the finest, truly precise groups for a particular set of auditory files [24].

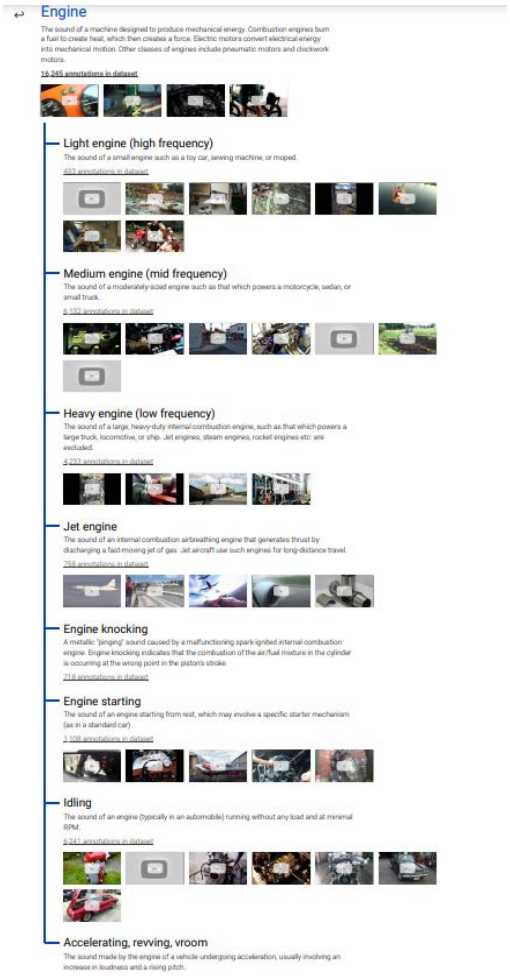

Figure 2-2 Engine Audio Set ontology [24].

2.3 The Audio Set Dataset Download

Engine related classes were selected from the engine's ontology. We made use of a toolkit for downloading the audio samples of engine knocking, engine starting, idling, and accelerating were used to detect knocking.

Using the toolkit, we have selected the engine classes to download the subset of Audio Set. The YouTube IDs that contain labels connected with the specified class is found by parsing the CSV files published for the dataset. URLs are created with the YouTube IDs using a variety of Python programs. Using the automatically created URLs and associated timestamps for each video, ten-second audio snippets are obtained. On the user's computer, clips are kept locally or on cloud for further use [25].

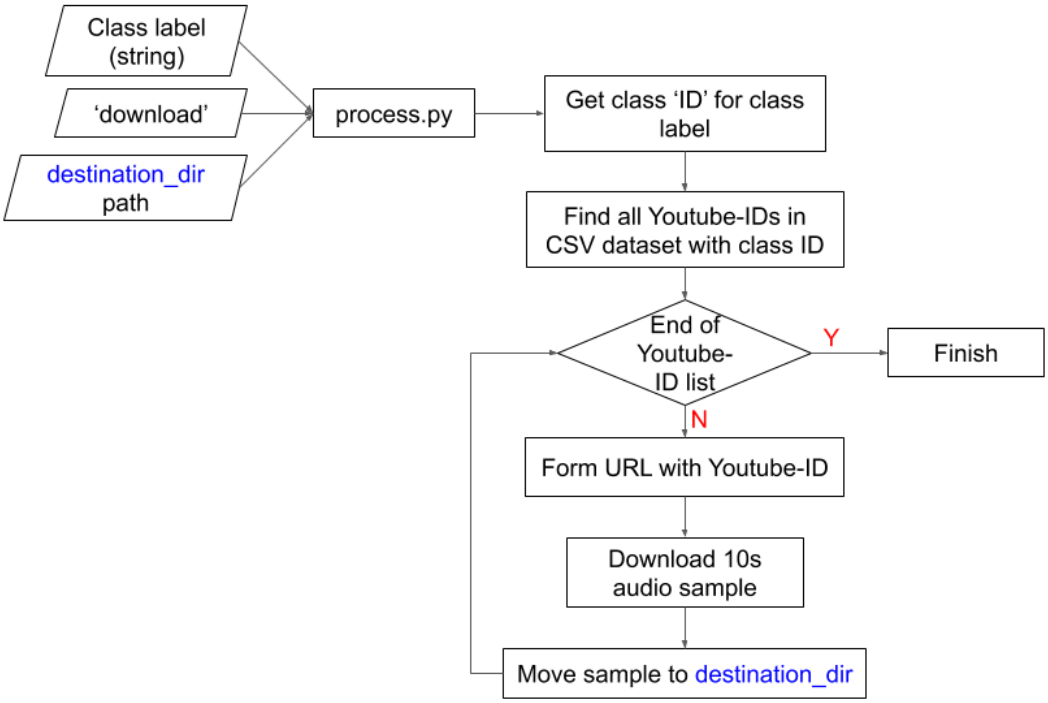

Figure 2-3 Flow chart of downloading a class from Audio Set [25].

Figure 2-3 demonstrates how to obtain engine classes from Audio Set without having to gather all the samples for each class repeatedly, which would take a considerable amount of time.

2.4 Improvised Toolkit for Downloading

The repetitive audio recordings in the toolkit that we downloaded for each class were a significant problem. It was challenging to download the files because each class took more than 4 hours to download and required the user to add input to an already downloaded or existing file. The existing code was modified, that makes sure duplicates are eliminated from the ontology file. As a result, the user is no longer required to submit information for every duplicate or existing file.

It had a section in the toolkit's utils file where it reads the CSV file provided by Audio Set and overwrites the CSV file. We removed the overwriting section from the code, which solved the user input problem, but the same file was downloaded twice. We then checked the CSV file, and it contained duplicates, which we removed and pipelined to the toolkit. Finally, the issue with user input was resolved, and we were able to efficiently download all the classes of engine.

Table 2-1 Description of engine audio set

S.no	Engine Audio Class	Audio Samples	Training	Testing	Total
1	Acceleration	100	16000 X 2049	4000 X 2049	20000 X 2049
2	Idle	100	16000 X 2049	4000 X 2049	20000 X 2049
3	Start	100	16000 X 2049	4000 X 2049	20000 X 2049
4	Knock	100	16000 X 2049	4000 X 2049	20000 X 2049
Total	4 Classes	400	64000 X 2049	16000 X 2049	80000 X 2049

We used 400 different audio samples from Google's audio set, which we preprocessed using our novel method of preprocessing the engine's audio samples, yielding 80,000 rows and 2049 columns.

3. SOUND PREPROCESSING TECHNIQUES

Python is a programming language that provides multiple libraries for audio processing, and it is rapidly growing as all multimedia trends. All Python libraries are open source and used for scientific and technical computing [26]. Librosa is an audio and music signal analysis library, it has been developed with easy to understand, use, parameters according to the terms used in sound analysis, retains backward compatibility against existing reference implementation, and keeps on evolving to resolve new issues [27]. We have used the Librosa library to process raw audio into deep-learning models.

3.1 Dependent Vehicle Feature Extraction

Librosa library reads the raw audio sample, and we extract the file's sample rate and data. Utilizing the MinMax scaler, the data is normalized before being subjected to random segmentation and packing. Next, we separate the data into training and testing sets for the deep learning network. Figure 3-1 portrays the methodology to create a training and testing test.

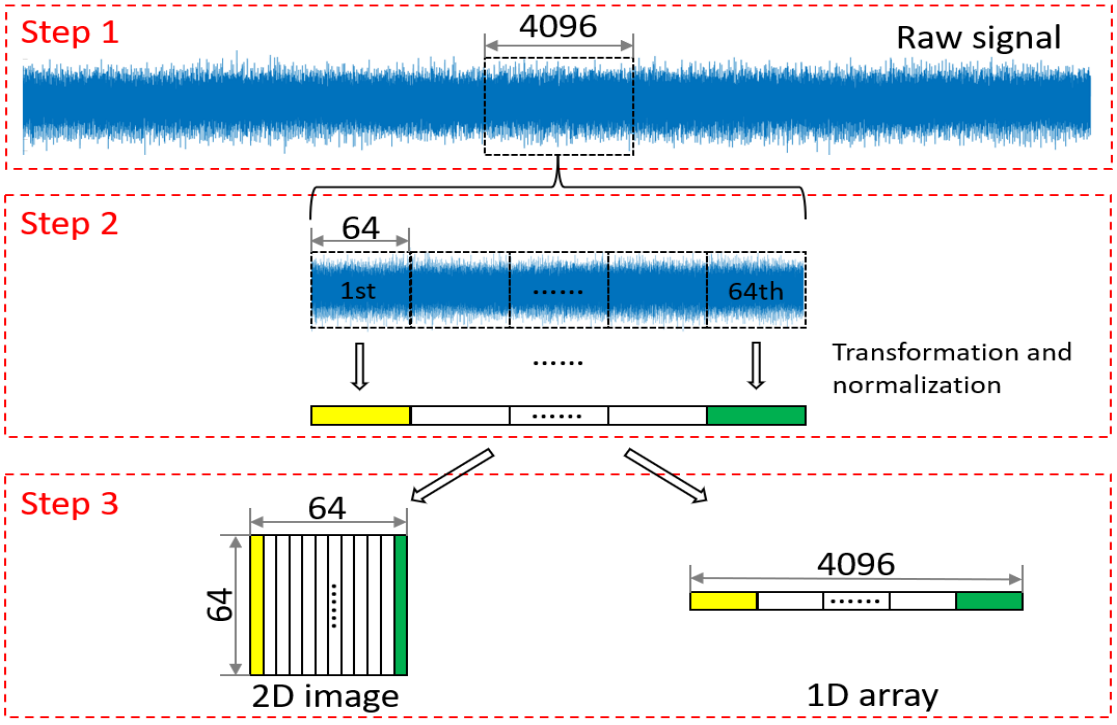

Figure 3-1 Preprocessing of audio using dependent vehicle feature extraction technique [28].

Step 1: Audio data is normalized, and we take n random points to pick k length signal from the audio data for segmentation.

Step 2: Almost every other frame is subsequently divided into Y bits of data records following acoustic fragmentation.

Step 3: In favor of a simple deep learning network, data blocks are stored in a list, whereas for a CNN network, data blocks are stored in a 2-dimensional array or a data frame. CNN has made a mark for itself in classification, having been used not only for images but also for audio and other data types. A fully connected CNN feature fusion model that combined audio and video results in accurate results, allowing us to be confident in CNN for our knocking diagnosis [29].

3.2 Sound Processing Methods

The sound handling strategy is significant in examining information present in the sound as the raw signal would neither lead us to any pattern recognition nor could be plotted to envision signals. Signal processing applications frequently require time domain information as well as the frequency content of the signal to analyze audio. Many applications for audio signal processing can be found in digital speech and audio, digital and cellular telephones, automobile controls, communications, biomedical imaging, image/video processing, and multimedia [30].

3.2.1 Fast Fourier Transform

The fast Fourier transform (FFT) is a computer algorithm that computes the discrete Fourier transform much faster than other algorithms. This algorithm is also useful to calculate power spectra. To understand FFT we must know the discrete Fourier transform, it is an algorithm that converts time-domain signal samples to frequency-domain components. This algorithm also establishes a connection between the time domain and frequency domain representations [31].

$$X(k) = \sum_{n=0}^{N-1} x(n) \, . \, e^{-2\pi kn/N} \;\; k = 0,1,2, \dots, N-1 \qquad (3.1)$$

Equation (3.1) shows $X(k)$ is the discrete Fourier transform of signal $x(n)$. The recurrent waves, that are challenging to identify from the initial spectrum, can easily be extracted and analyzed using this technique. FFT has fewer computations when compared with discrete Fourier transform. Using this method, we can concentrate on each signal in the audio files. FFT is much faster than algorithms as it computes transformation for the whole signal of 1024 points 100 times

faster than other algorithms. This algorithm involves fewer computational addition, subtraction, and multiplication operations.

In python, we can compute one-dimensional FFT using the NumPy library which has a function fft which uses the fastest algorithm to compute the Fourier transform of the audio signal. To obtain a frequency domain representation of the audio, engine audio samples were transformed using FFT in steps involved for dependent vehicle feature extraction technique. The sound files' totality of periodic and aperiodic impulses was converted from time to frequency. This method has been used in our investigation to draw attention towards the knocking signal, which is vividly observable in the frequency domain.

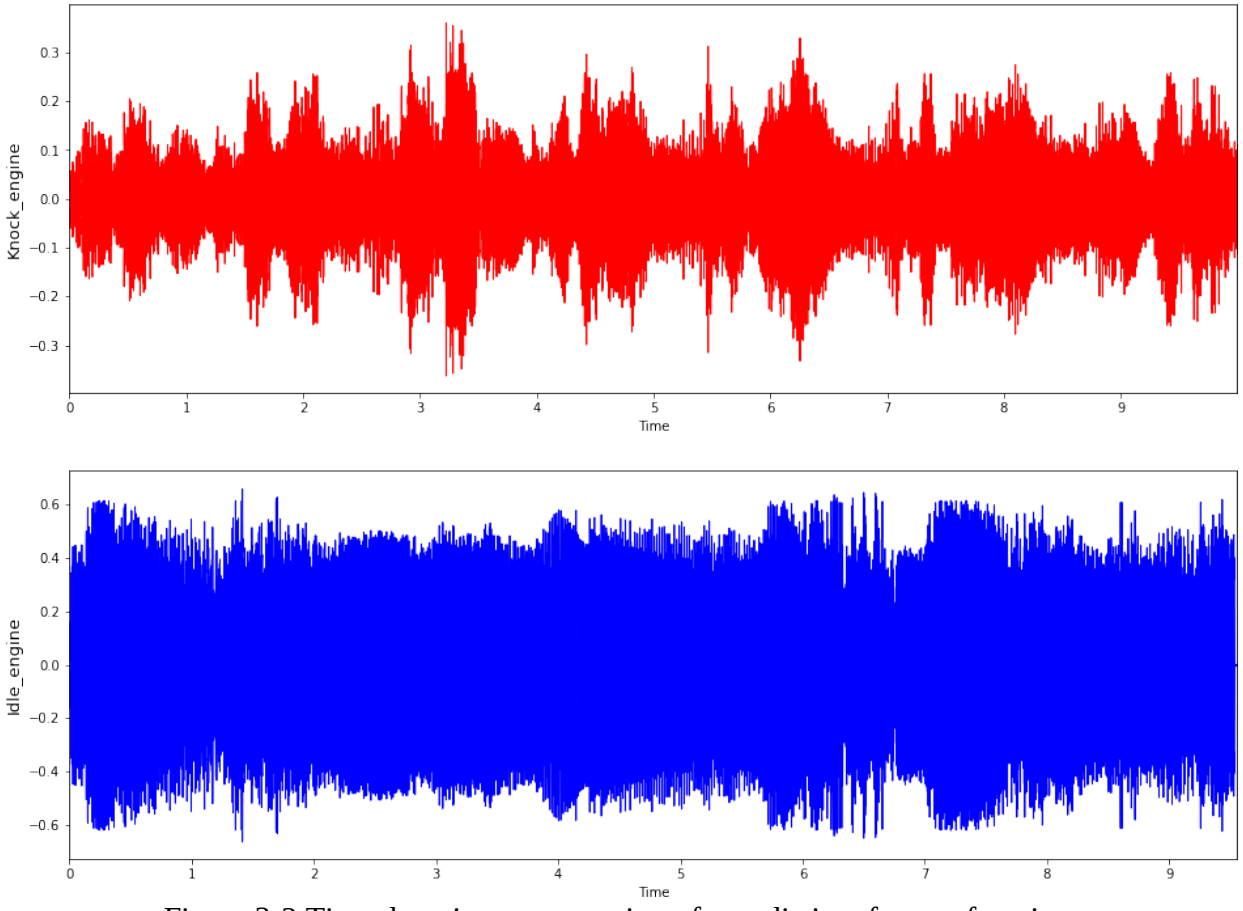

Figure 3-2 Time domain representation of two distinct forms of engine.

Figure 3-2 depicts two different audio files of engine datasets on the time domain, the signal in red being the engine knocking clip and the other being the idle engine.

Figure 3-3 clips are from the same categories as in Figure 3-2 and are a frequency domain representation of audio files.

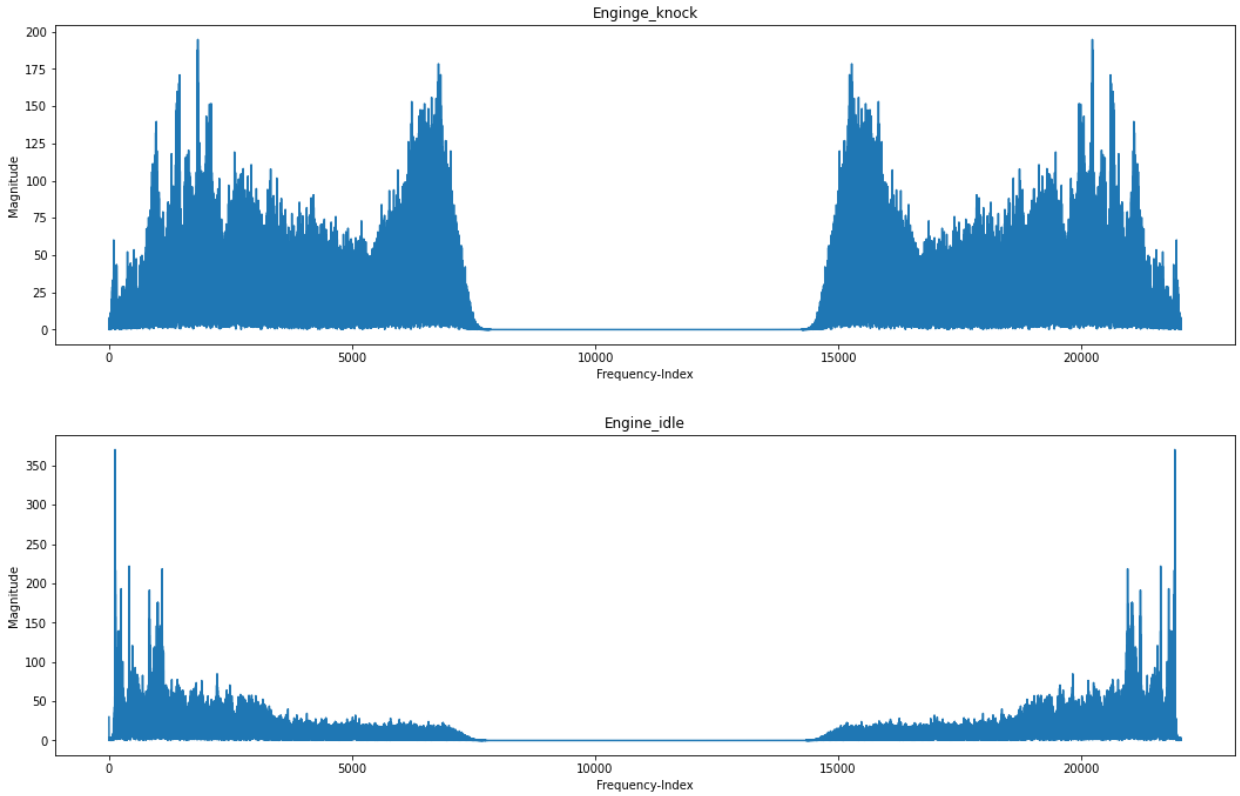

Figure 3-3 Frequency domain representation of two distinct forms of engine.

3.2.2 Mel Frequency Cepstral Coefficients

MFCC is a widely used technique in speech analysis. It is based on the known variation of the crucial frequency bandwidth of the human ear. De-connecting the result log energies of a channel bank made up of triangle filters which are straightly separated on the Mel frequency hierarchy yields the MFCC coefficients. As the most accurate estimate of the Karhunen-Loeve transform (KLT) currently available, the discrete cosine transform (DCT) implementation identified as distributed DCT (DCT - II) is typically used to separate the speech (KLT). A melodic cepstral acoustic vector is represented by MFCC data sets. The acoustic vectors are feature vectors that can be applied. A derivation on the MFCC acoustic vectors can be used to acquire more precise speech features. Speech signals do not scale linearly. As a result, for each tone with an actual frequency

at the sampling rate of 22 kHz. An idiosyncratic tone is computed in Hz at a range known as the 'Mel' scale.

$$Mel(f) = 2595 \ln\left(1 + \frac{f}{700}\right) \tag{3.2}$$

The Equation (3.2) converts frequency to Mel scale measurement where f is the frequency in Hz. MFCC coefficients are a bunch of DCT decor-related objects calculated via a modification of the logarithmically compacted channel result of vibrance got from a conceptually separated three-sided channel bank that conducts the DFT discourse acoustic.

$$C_m = \sqrt{\frac{2}{M}} \sum_{l=0}^{Q-1} \log[e(l+1)].\cos[m.\left(\frac{2l+1}{2}\right).\pi/Q] \tag{3.2}$$

Equation (3.2) calculates number of MFCC coefficients C_m where *m=0,1, 2,… ,R-1* , and R is the chosen amount of MFCCs. Filters on the Mel-frequency scale have been placed in the MFCC's filter bank structure. Because the neighboring filters consist of a corresponding region, they include additional interconnected data when compared to the filters extended. The correlation between filter powers varies (not holding to a first-order Markov correlation). Because of the non-constant relationship together with the filter bank productions, utilizing a DCT to the whole log-energy vector is improper. It is suggested to apply DCT in a disseminated way to obey the Markov fundamentals [32]. Engine knocking is a sound produced by a machine and we applied MFCC over the samples to differentiate engine sounds by extraction coefficients using the dependent vehicle feature extraction method. Librosa library has been used to convert audio samples into MFCC coefficients. By offering arguments to set the number of frames, hop length, number of MFCCs, and other parameters, the librosa. feature. mfcc method makes acquiring MFCCs easier.

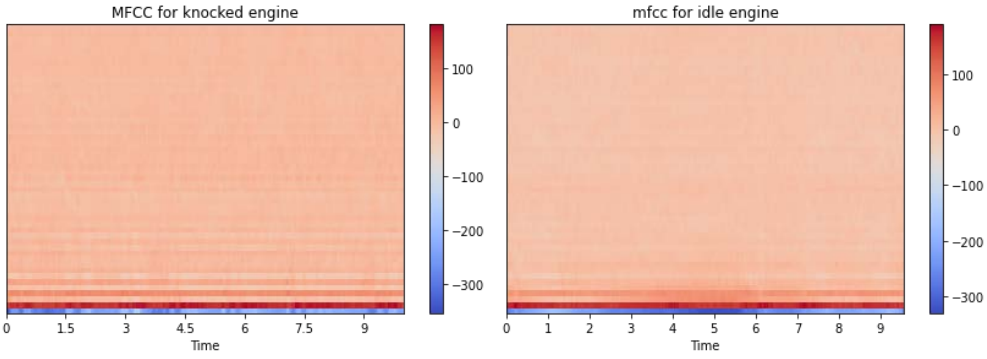

Figure 3-4 MFCC for knoked and idle engine.

The MFCC spectrogram for 50 coefficients for knocked and idle engine audio samples is shown in Figure 3-2. There are some differences in above images where the high dimensional model resulted in pattern recognition but at a high computational cost.

3.2.3 Short-time Fourier Transform

The traditional Fourier transform is the foundation for the short-time Fourier transform (STFT). The reason for developing this transformation was to enhance the Fourier transform (FT), which can manage the recurrence space, by duplicating a windowing capability with limited time before the Fourier change. The STFT could guarantee permission to recurrence space data while retaining adequate time-domain material. The Fourier change of the subsequent sign is utilized as the windowing capability sliding along the sign set, permitting the restriction and time-fluctuating recurrence area investigation to be comprehended [33]. The data to be transformed could be divided into frames in the discrete-time case (which usually have a fixed overlap). Each frame is Fourier transformed, and the resulting complex result is added to a matrix, which records the magnitude and phase for each time and frequency point. The STFT is defined in Equation (3.6) as shown below.

$$X(k) = \frac{1}{W}\sum_{n=0}^{W-1}[x(n)w(n)]e^{-j2\pi kn/W} \quad k = 0,1,\ldots,W-1 \tag{3.6}$$

where $x(n)$ is the original signal, $w(n)$ is the window function, which can reduce the leakage of the spectrum of the transformation. When compared to FFT, STFT has more information for audio samples and ensures that repeated information is not considered. STFT can be used to extract engine noises because it does away with FFT's restrictions. The addition of the window function allowed the STFT to accurately reflect the properties of engine sound signals in both the time and frequency domains, allowing it to fully show the dynamic process of engine sound and distinguish between distinct engine sound signals.

In STFT, the resolution of time and frequency can be traded off. To put it another way, a narrow-width window produces a better resolution in the time domain but a poorer resolution in the frequency domain, and vice versa. The spectrogram, which is an intensity representation of STFT magnitude over time, is frequently used to visualize STFT. Figure 3-4 displays two spectrograms showcasing various time-frequency resolutions.

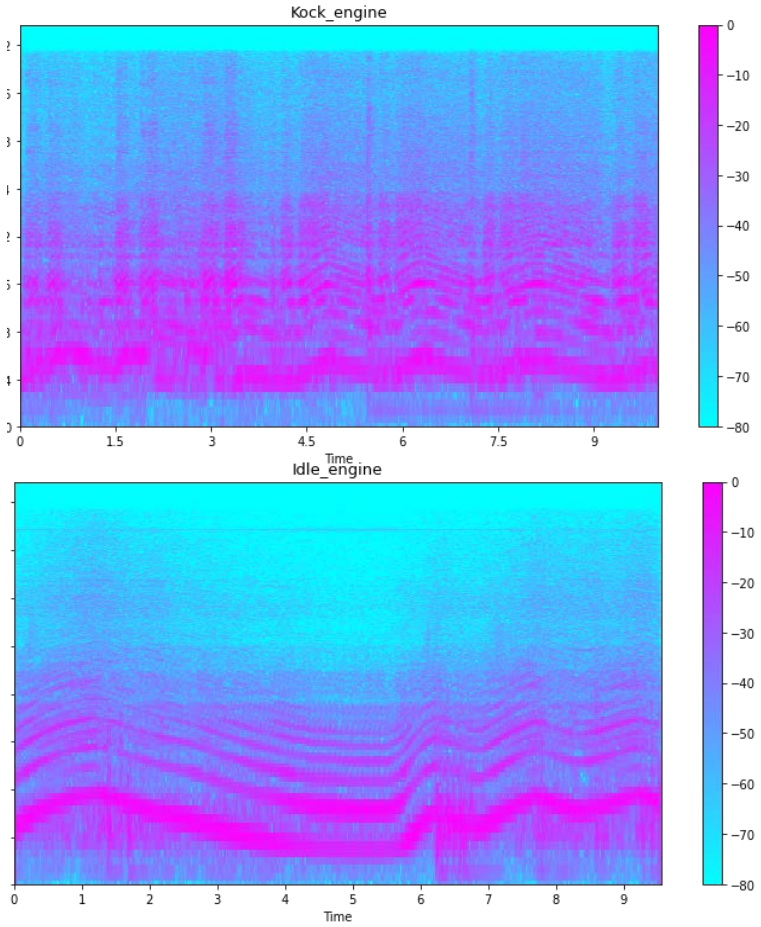

Figure 3-5 STFT for knocked and idle engine.

We used STFT to extract time and frequency features from our audio data using a sliding window that allows us to jump or skip data points from the sample. The Librosa library includes an STFT feature extraction function with various parameters for extracting audio according to the requirements.

4. DEEP LEARNING NEURAL NETWORKS PERFORMANCE

Deep learning (DL) remains exceptionally vital around our daily lives. This area of research has already had a significant influence in fields, for example, malignant growth identification, accuracy medication, self-driving vehicles, prescient gauging, and discourse acknowledgment. Conventional understanding, categorization, and design establishing techniques use meticulously handcrafted feature extractors that remain incomprehensible for enormous datasets. DL can likewise conquer the restrictions of prior superficial algorithms that forestalled effective preparation and deliberations of various leveled portrayals of multi-layered training information much of the time, contingent upon the issue's intricacy. Deep neural network (DNN) uses many levels of components that are extremely augmented in terms of procedures and designs [34].

Deep neural networks' success is based on advances in fast and large-scale computations. These structures typically demand additional reckoning capacity as well as preparation information than traditional approaches. Central processing units are not ideal for these complex algorithms. As an alternative, matrix-boosted processors, primarily conventional-end graphics managing elements and product-limited multilingual electric circuit like commercial tensor processing units, are commonly used. Smaller models are required for applications with limited high-end resources, like cellular phones or listening services. While many recent works focus on neural network generalization, solidity, or preparing with small budgets, the issue could be reasonable to explore choices for the necessities of constant sound signal handling [35].

4.1 Artificial Neural Network (ANN)

An ANN is a particular kind of neural network which prepares using procedures to study interpretations from datasets deprived of the need for manually designing feature extractors. It is well known under the category of multilayer perceptron (MLP). In contrast to the shallow learning model, which has fewer layers of units, deep learning has a higher or deeper number of processing layers. More complicated and continuous tasks can now be efficiently represented because of the transition from shallow to deep learning systems [36].

The idea of an ANN is first introduced in the context of biology, where neural networks are crucial to the functioning of the human body. Neural networks assist in the functioning of the

human body. A neural network is nothing more than a web of millions of millions of interconnected neurons. The human body performs all parallel processing with the assistance of these interconnected neurons, making it the ideal example of parallel processing. A neuron is a specific type of biological cell that uses electrical and chemical changes to transmit information from one neuron to another [37].

Keras is Powerful and simple to use, it is a free open-source Python framework for creating and analyzing deep learning models. It enables you to create and train neural network models and is a component of the TensorFlow library. Using data, we can construct an ANN, specify the model, compile it, fit it, and forecast the outcomes [38].

Even though there are other deep learning libraries like PyTorch and MXnet, we still favor using Keras because of its straightforward framework for building, compiling, and running models [39]. The effectiveness of the method employed to categorize and recognize bird sounds was the deciding factor in the selection of ANN [40]. To improvise the results produced by ANN can be modified using hyper parameter tuning, there are number of arguments and parameters present in ANN that can be modified according to the results [41]. The technique of spectrum detection involves comparing a signal's spectrum against a fixed or adaptive threshold (automatic). There is a sound at that frequency if the spectrum is greater than the threshold value for that frequency, which can be accurately detected using deep-learning neural networks [42].

Predictive Analysis ANN can be used for both supervised and unsupervised training by preprocessing using normalization, fast Fourier transformation, Short-Time Fourier Transformation, Mel Filter bank, and Mel Frequency Cepstral Coefficient [43]. Deep Neural Network with hyperparameter tuning with hyper parameters improves the model furthermore [44]. Our problem focuses on supervised training, and within this training, we have a classification problem. Humans labeled the audio samples provided by Audio Set, and we built a classification ANN model to accurately classify engine sounds.

$$ReLU(z) = \max\{0, z\} \tag{4.1}$$

Equation (4.1) is a ReLU activation function used in hidden layers, which chooses max between 0 and *z*.

$$Multiclass\ Classification: g(x)_j = \frac{e^{(w^j_{K+1}\, x + b^j_{K+1})}}{\sum_{k=1}^{N} e^{(w^j_{k+1} x + b^k_{k+1})}} \tag{4.2}$$

Equation (4.2) gives a number between 0 and 1 for each class, SoftMax function is used for multi-variate classification.

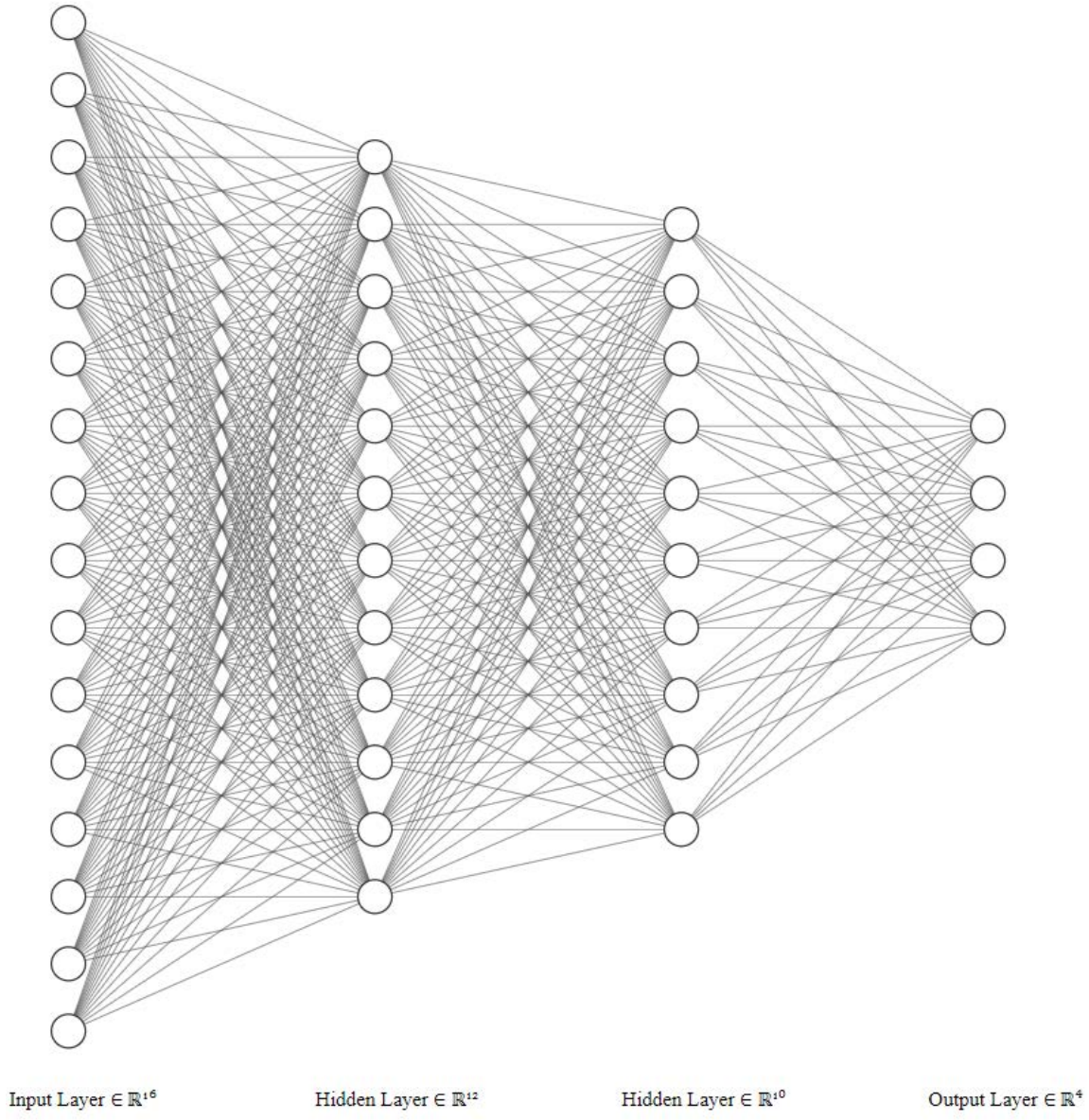

Figure 4-1 ANN architecture for multi-class.

Figure 4-1 depicts the classification architecture used in our research. It has one input layer, two hidden layers, and one output layer with four neurons because we are dealing with four different types of audio files. Activation functions in ANN are an essential component of neural network design. The activation function used in the hidden layer determines how well the network

model learns the training dataset. The type of predictions that the model can make will be determined by the activation function used in the output layer. In our architecture we made use of ReLu activation function in hidden layers and SoftMax function for the output layer.

4.2 Experiment Results

Engine knocking, engine idle, engine accelerating, and engine starting are the four categories into which engine audio clips from the Audio Set dataset are divided. Before diving into the results, let's talk about the several feature extraction strategies that were used to fit the data into binary classification and multi-class classification models. We employed simple ANN to classify these audio clips using the dependent vehicle feature extraction technique. Figure 4-2 displays the FFT feature extraction utilized for engine knocking and engine idling.

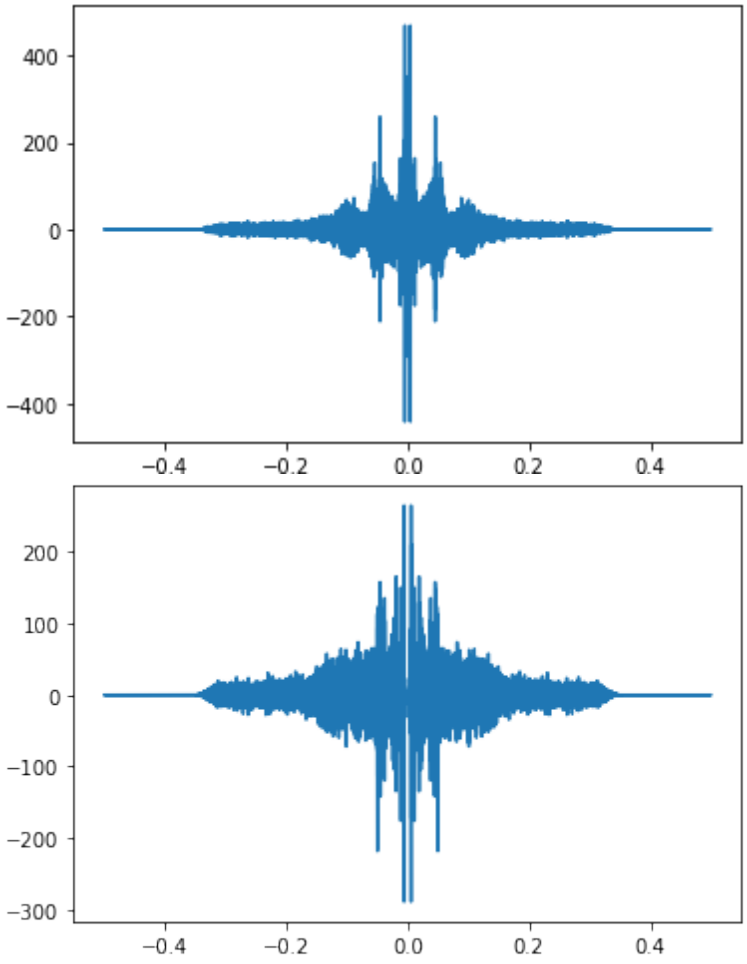

Figure 4-1 FFT of engine knocking and engine idle.

We used these audio files for FFT binary classification and multi-class classifications Figure 4-3, shows FFT for engine starting and engine accelerating. We started with binary classification because the results were noteworthy, and then we switched to multi-class classification.

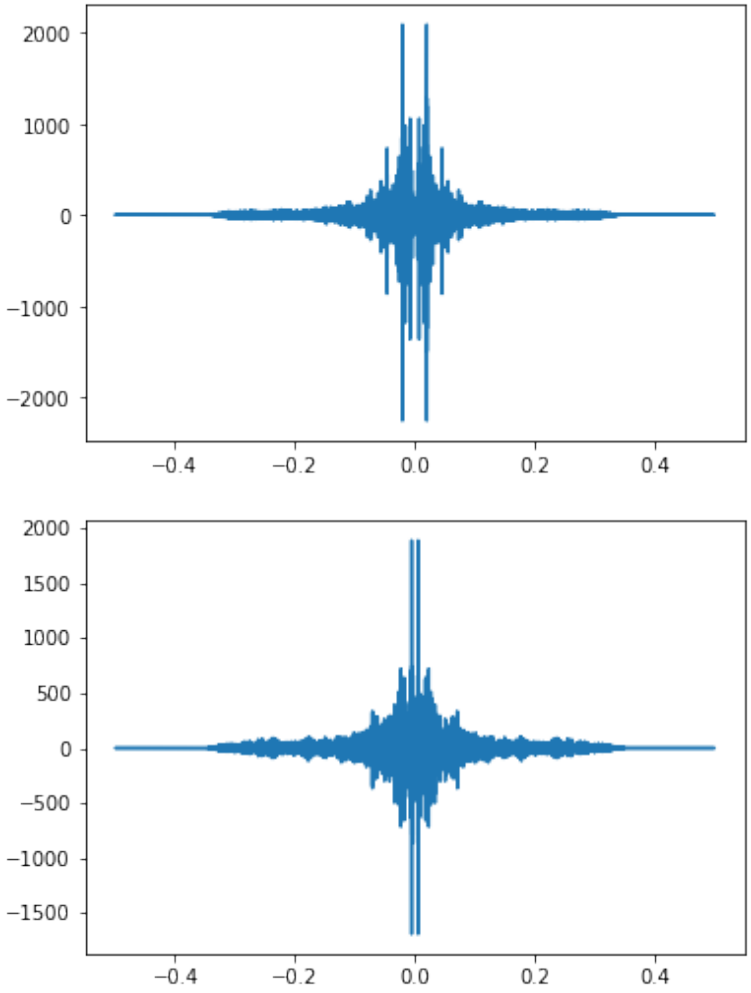

Figure 4-3 FFT of engine accelerating and engine starting.

We used MFCC to extract audio features from all four classes of engine audio files after applying FFT. When comparing the MFCC feature extraction technique against numerous models, including binary and multi-class classification, FFT came out on top. FFT analyzes a signal to determine its frequency content. MFFCs are perceptually motivating characteristics that mimic how humans perceive pitch. FFT has a linear resolution and evenly spaced bins.

Figure 4-4 depicts the MFCC feature extracted from the engine starting and accelerating at 50 MFCC'S, whereas Figure 4-5 displays MFCC extraction of engine idle and knock with similar MFCC coefficients as Figure 4-4.

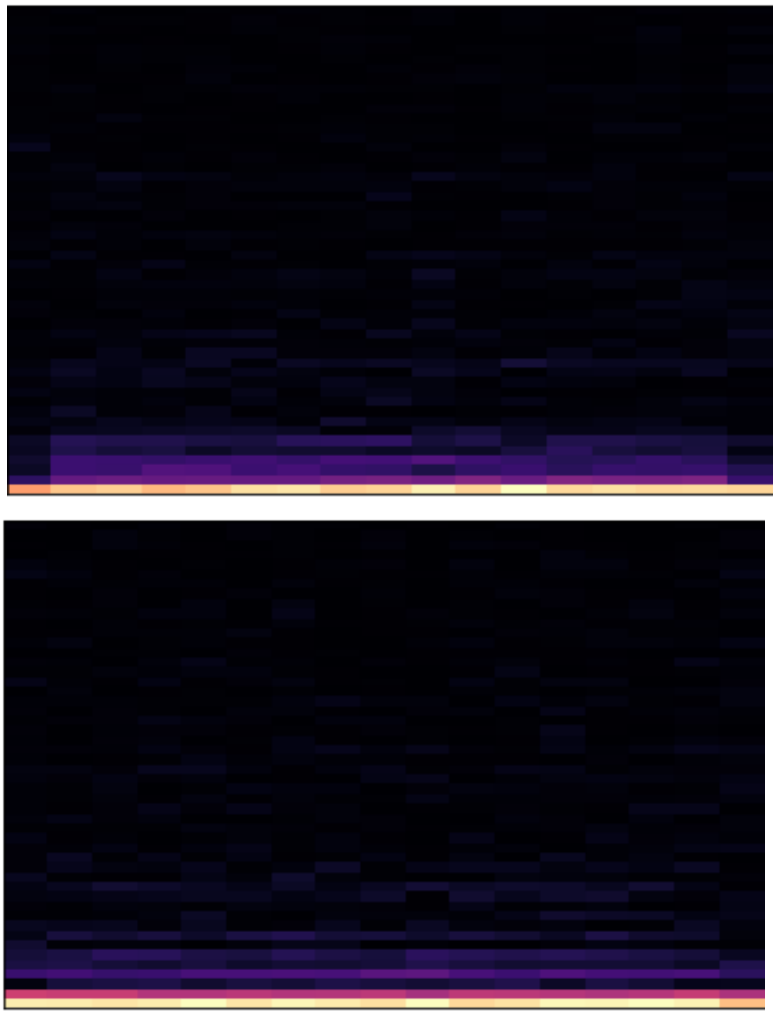

Figure 4-4 MFCC of engine starting and engine accelerating.

MFCCs are a condensed representation of the spectrum of an audio signal, where a waveform is represented by the sum of a potentially infinite number of sinusoids. When a cepstral coefficient is positive, most of the spectral energy is concentrated in low-frequency regions. If, on the other hand, a cepstral coefficient is negative, it indicates that the vast majority of the ghastly energy is assembled at increased frequencies.

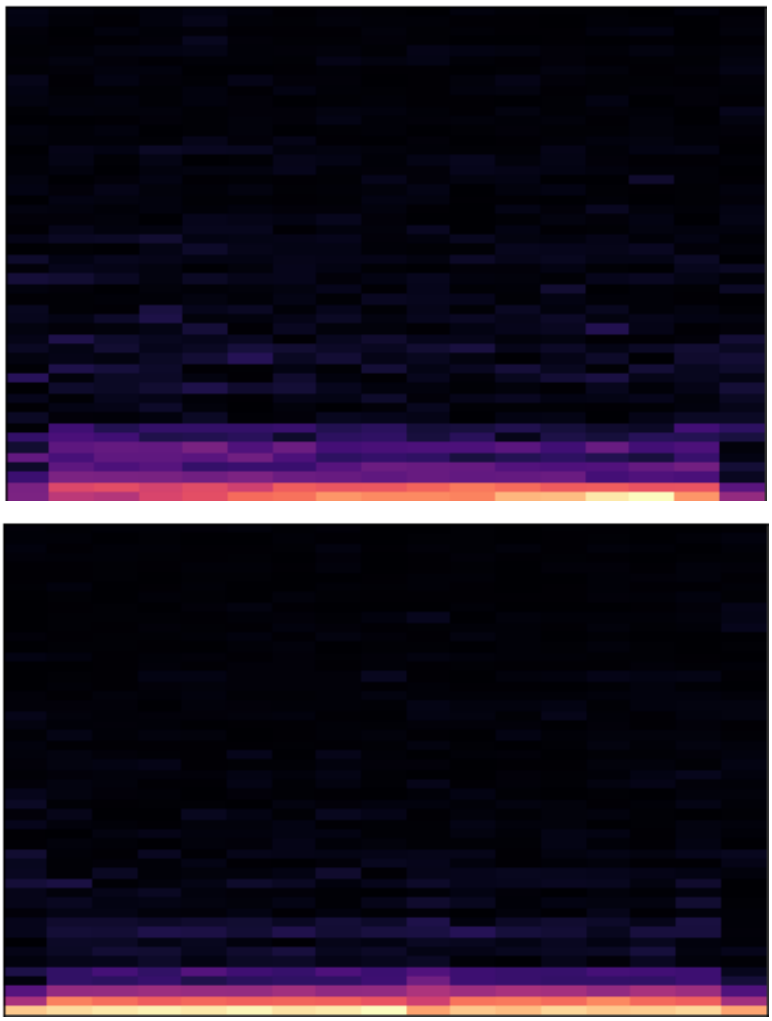

Figure 4-5 MFCC of engine knocking and engine idle.

4.2.1 Results for ANN using FFT feature extraction

Figure 4-3 depicts FFT for engine starting and accelerating; we used these audio clips for FFT binary and multi-class classifications. We started with 17 epochs for binary classification model and then moved on to 50 epochs for multi-class classification.

(1) ANN FOR ENGINE KNOCK VS ENGINE IDLE USING FFT

The average accuracy is 96.17% and it takes 1 mins and 25 sec to finish the training. Figures 4-4, 4-5, 4-6, 4-7, 4-8, and 4-9 show that the method has good classification accuracy.

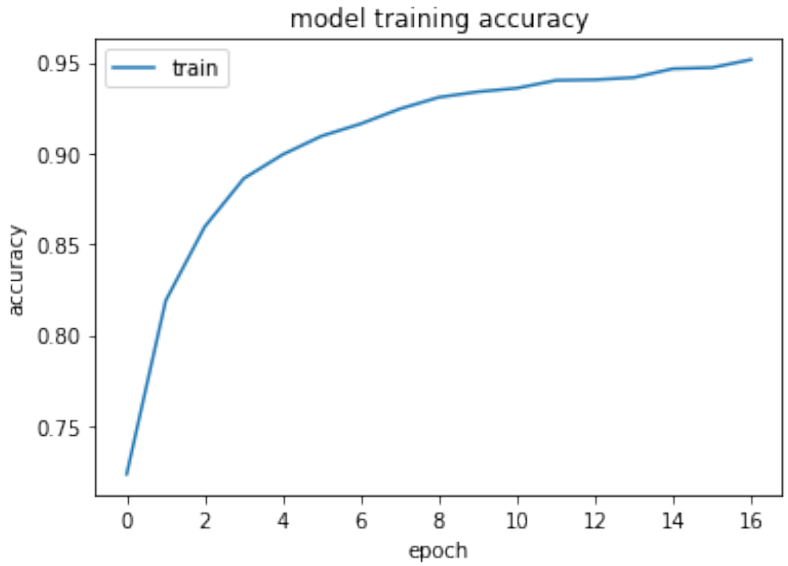

Figure 4-6 Training progress of ANN (knock vs idle).

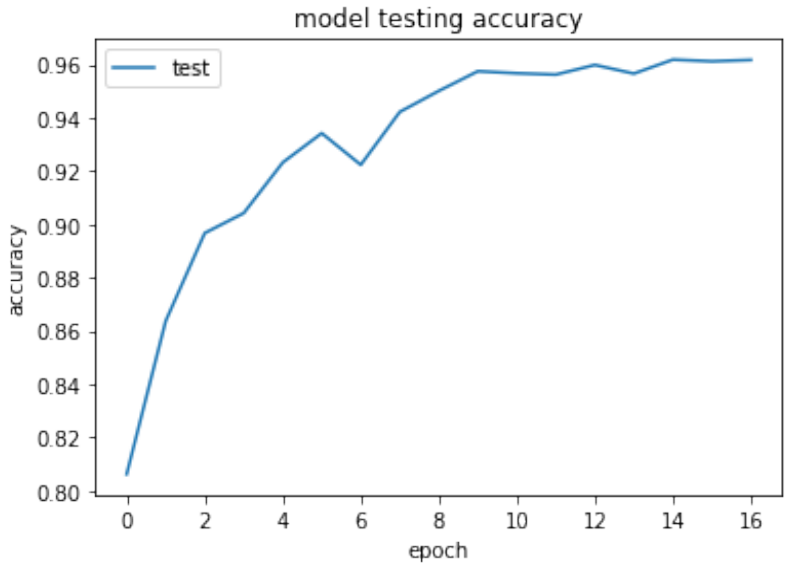

Figure 4-7 Testing accuracy of ANN (knock vs idle).

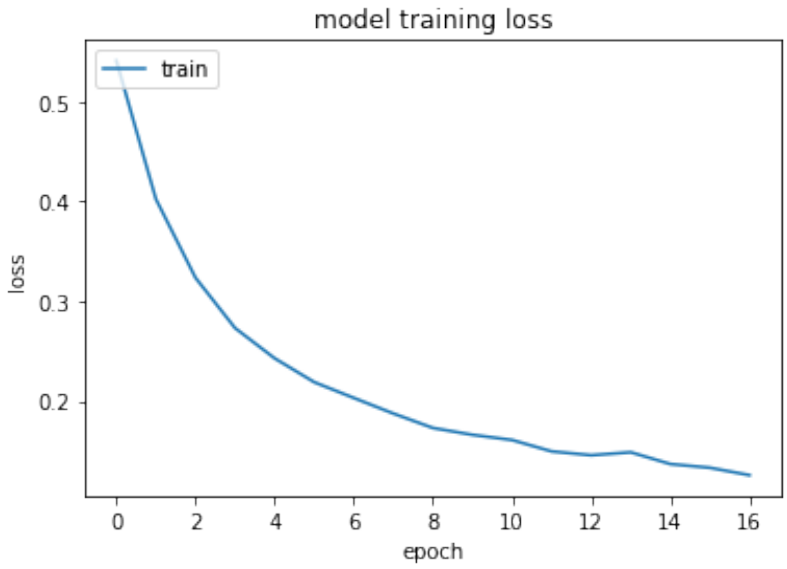

Figure 4-8 Training loss of ANN (knock vs idle).

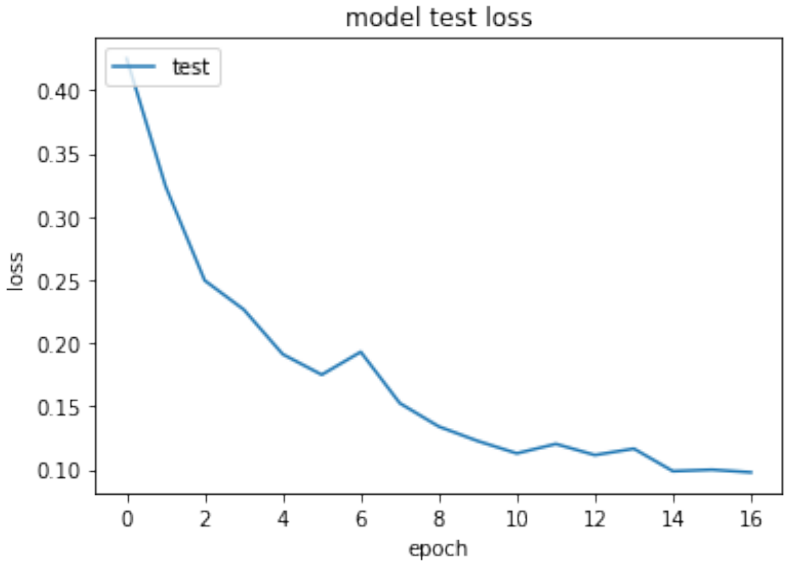

Figure 4-9 Testing loss of ANN (knock vs idle).

	precision	recall	f1-score	support
0	0.96	0.97	0.96	4065
1	0.96	0.96	0.96	3935
accuracy			0.96	8000
macro avg	0.96	0.96	0.96	8000
weighted avg	0.96	0.96	0.96	8000

Figure 4-10 Evaluation metrics of ANN (knock vs idle).

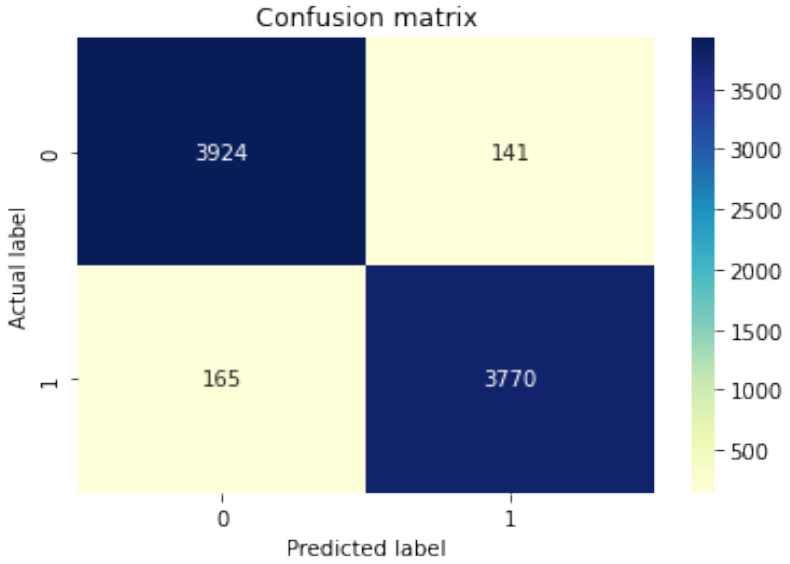

Figure 4-11 Confusion matrix for ANN (knock vs idle).

(2) ANN FOR ENGINE KNOCK VS ENGINE START USING FFT

The average accuracy is 93.16%. It takes 1 mins and 24 sec to finish the training. From the below Figures, we can see that the method has lower classification accuracy when compared with the above ANN model.

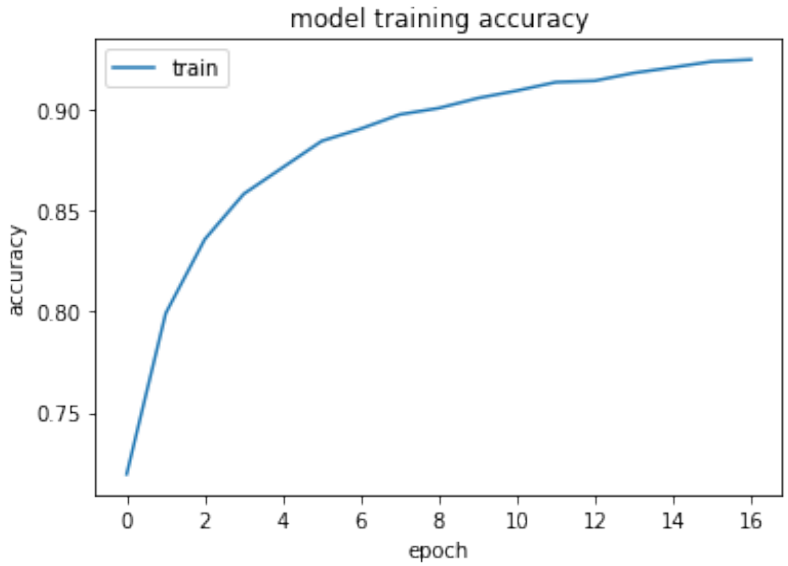

Figure 4-12 Training progress of ANN (knock vs start).

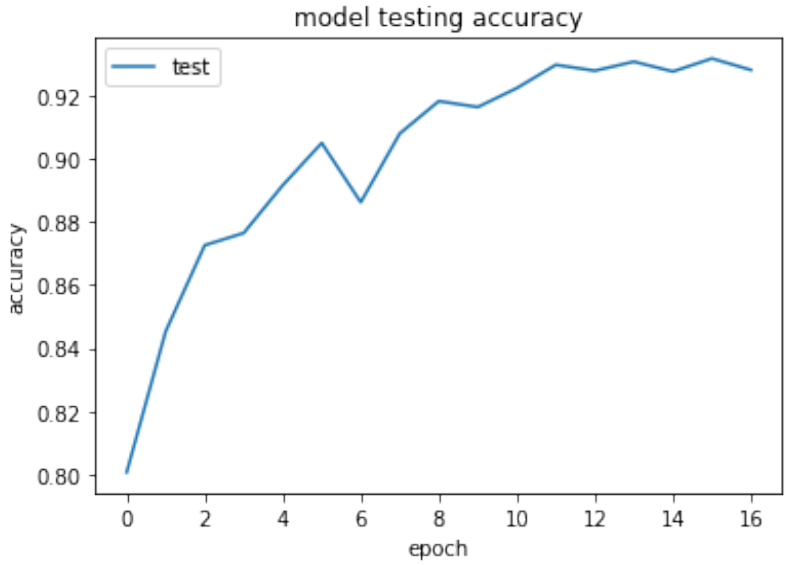

Figure 4-13 Testing accuracy of ANN (knock vs start).

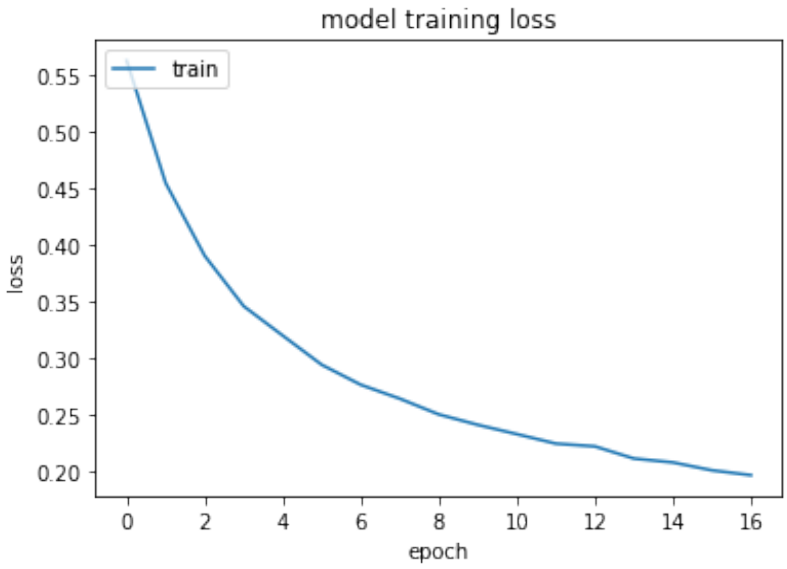

Figure 4-14 Training loss of ANN (knock vs start).

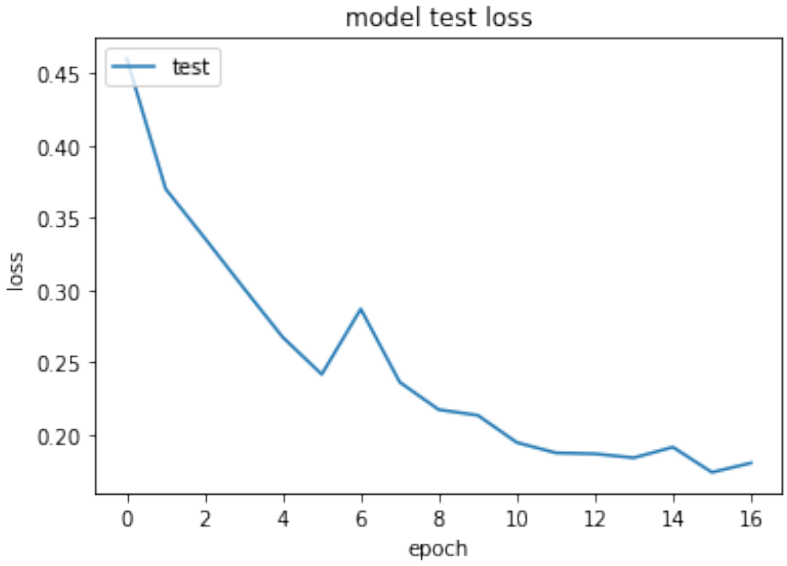

Figure 4-15 Testing loss of ANN (knock vs start).

	precision	recall	f1-score	support
0	0.93	0.92	0.93	3927
1	0.92	0.94	0.93	4073
accuracy			0.93	8000
macro avg	0.93	0.93	0.93	8000
weighted avg	0.93	0.93	0.93	8000

Figure 4-16 Evaluation metrics of ANN (knock vs start).

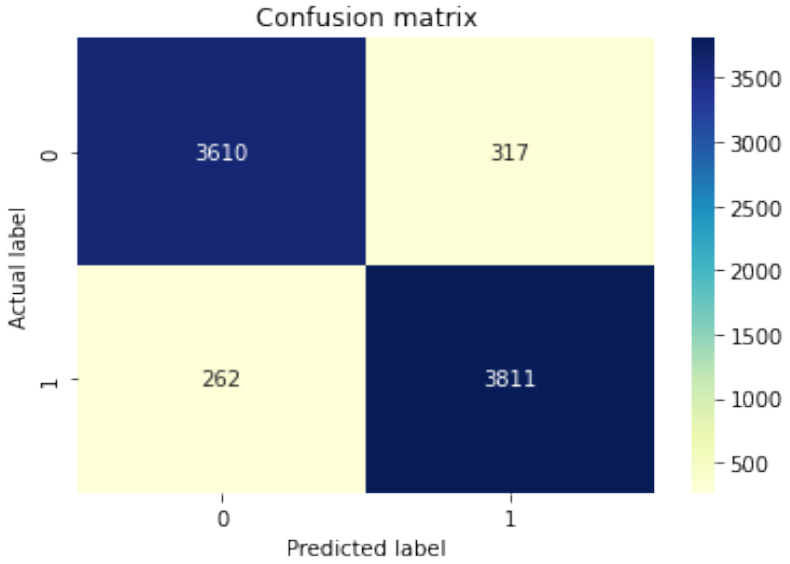

Figure 4-17 Confusion matrix for ANN (knock vs idle).

(3) ANN FOR ENGINE KNOCK VS ENGINE ACCELERATION USING FFT

The average accuracy is 97.00%; it takes 1 mins and 06 sec to finish the training. Figures below show that the method has better classification accuracy when compared with engine start and idle audio samples ANN model.

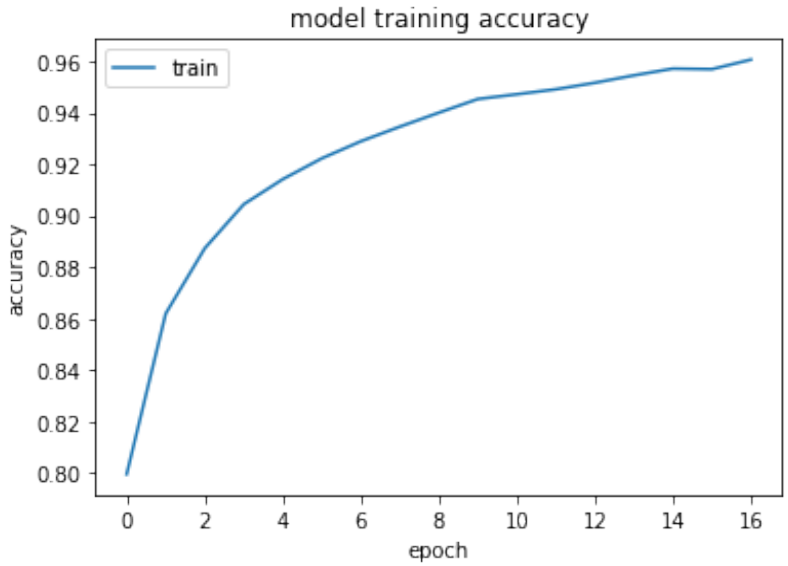

Figure 4-18 Training progress of ANN (knock vs acceleration).

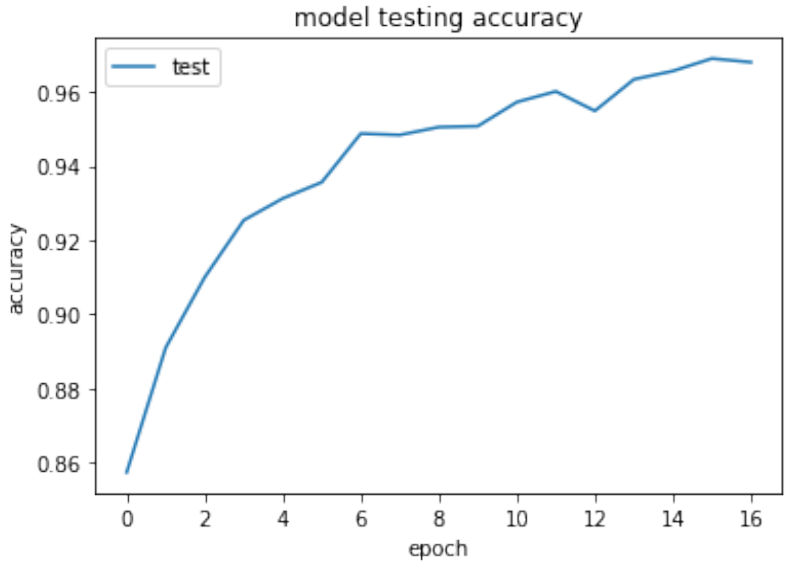

Figure 4-19 Testing accuracy of ANN (knock vs acceleration).

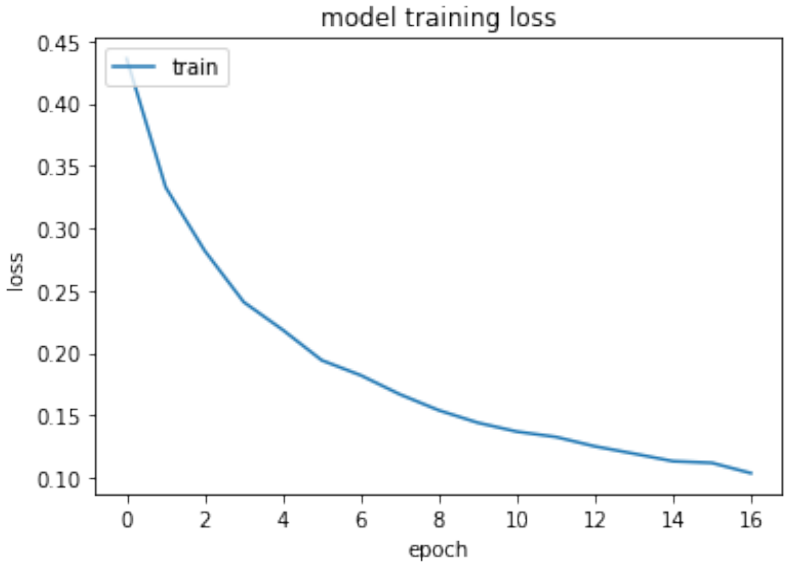

Figure 4-20 Training loss of ANN (knock vs acceleration).

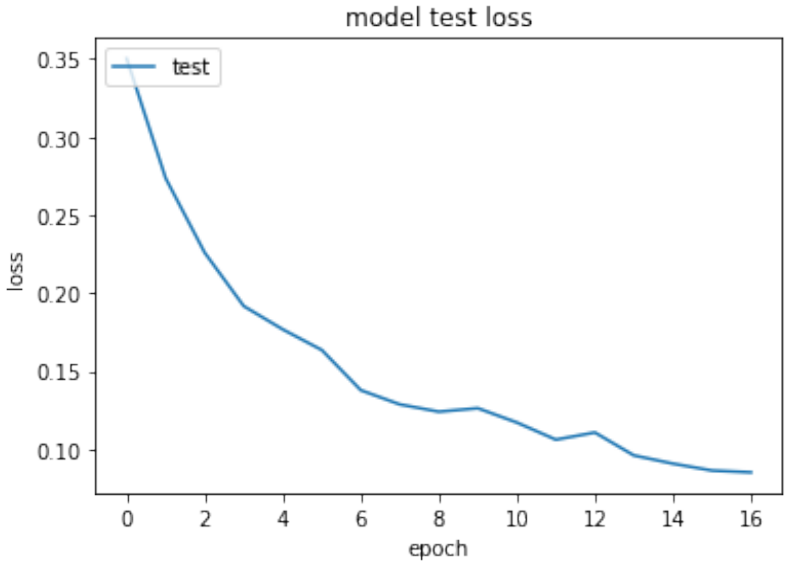

Figure 4-21 Testing loss of ANN (knock vs acceleration).

	precision	recall	f1-score	support
0	0.98	0.95	0.97	4053
1	0.95	0.98	0.97	3947
accuracy			0.97	8000
macro avg	0.97	0.97	0.97	8000
weighted avg	0.97	0.97	0.97	8000

Figure 4-22 Evaluation metrics of ANN (knock vs acceleration).

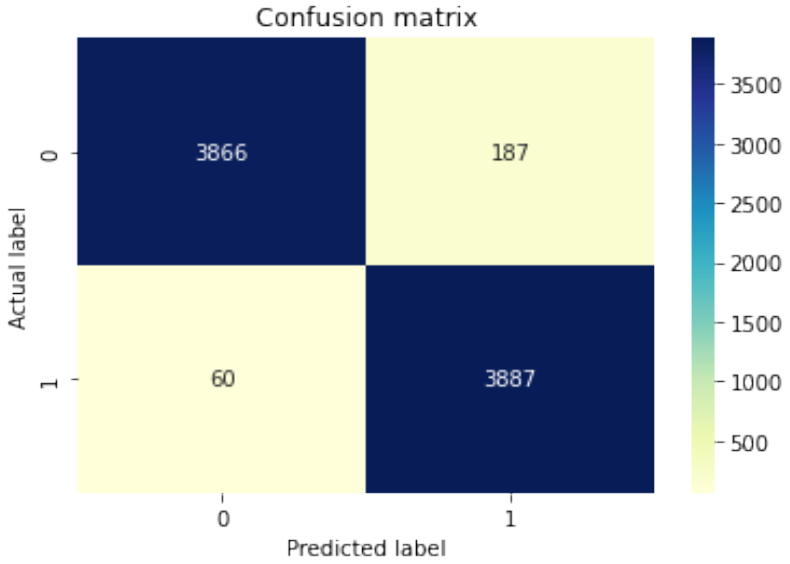

Figure 4-23 Confusion matrix for ANN (knock vs acceleration).

(4) ANN FOR MULTICLASS CLASSIFICATION USING FFT

The average accuracy is 92.00%. It takes 6 mins and 28 sec to finish the training. From Figures beneath exhibits that the method has optimal classification accuracy for all the engine audio samples ANN model.

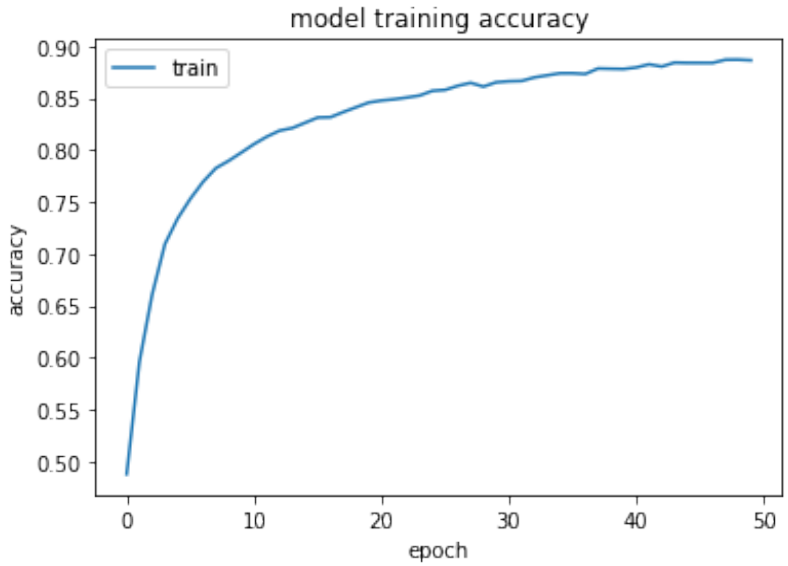

Figure 4-24 Training progress of ANN (multi-class classification).

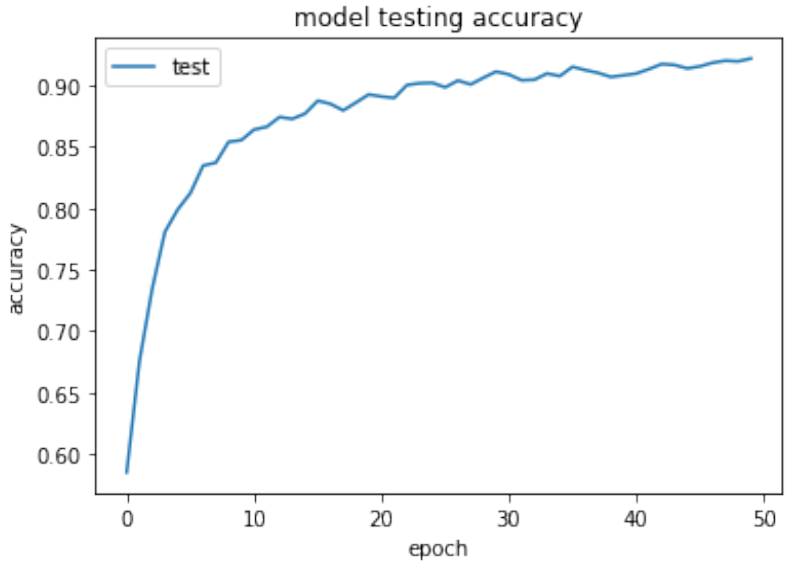

Figure 4-25 Testing accuracy of ANN (multi-class classification).

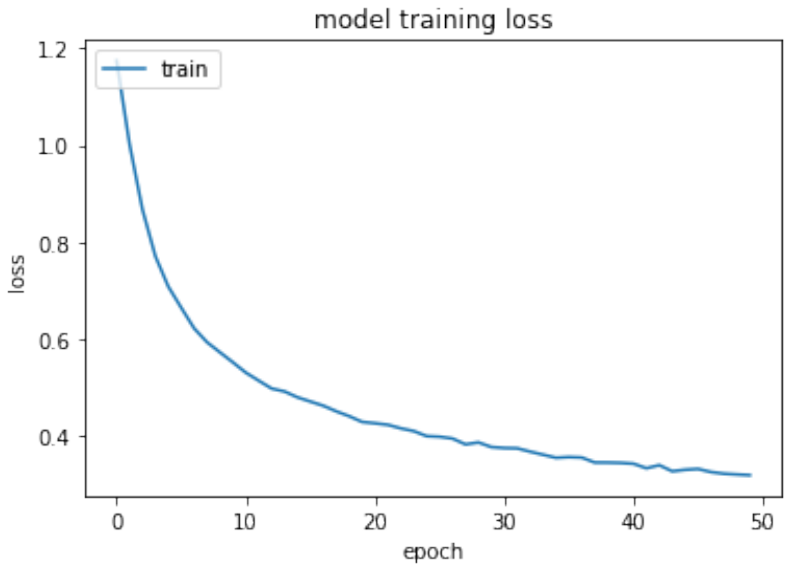

Figure 4-26 Training loss of ANN (multi-class classification).

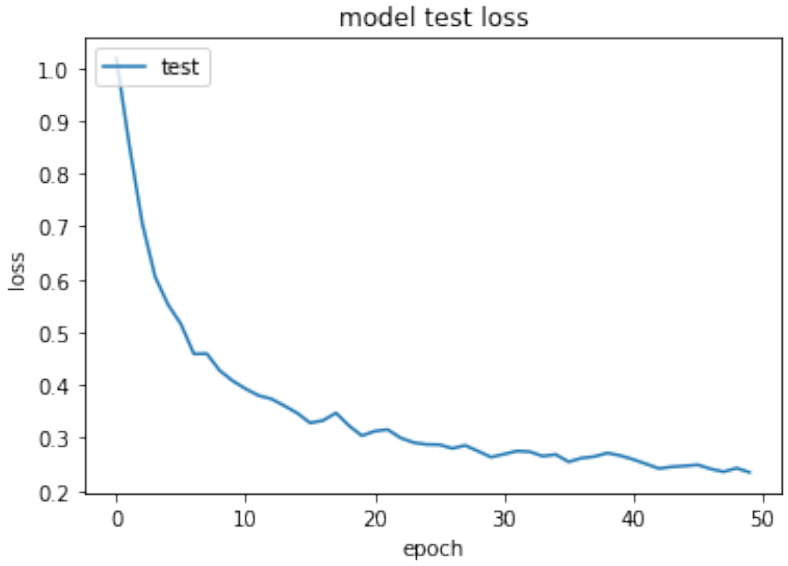

Figure 4-27 Testing loss of ANN (multi-class classification).

	precision	recall	f1-score	support
0	0.87	0.94	0.91	3986
1	0.98	0.92	0.95	4114
2	0.95	0.91	0.93	4014
3	0.88	0.90	0.89	3886
accuracy			0.92	16000
macro avg	0.92	0.92	0.92	16000
weighted avg	0.92	0.92	0.92	16000

Figure 4-28 Evaluation metrics of ANN (multi-class classification).

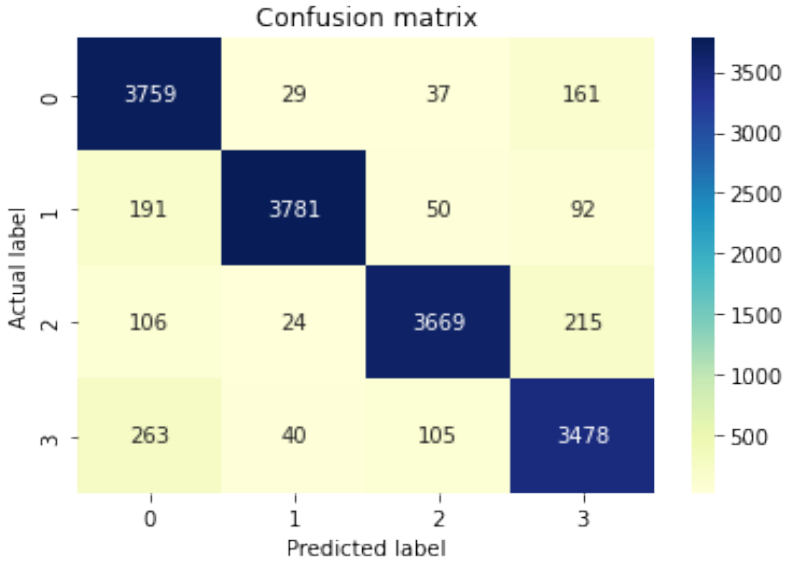

Figure 4-29 Confusion matrix of ANN (multi-class classification).

4.2.2 Results for ANN using MFCC feature extraction

We know that MFCC is appropriate for human speech, but we used it to the engine sound samples, which generated intriguing results. FFT has produced some fantastic results not only for binary classification, but also for multi-class classification.

(1) ANN FOR ENGINE KNOCK VS ENGINE IDLE USING MFCC

The average accuracy is 90.28% and it takes 3 mins and 22 sec to finish the training. Images below illustrates that the method has produced almost equal classification accuracy as FFT.

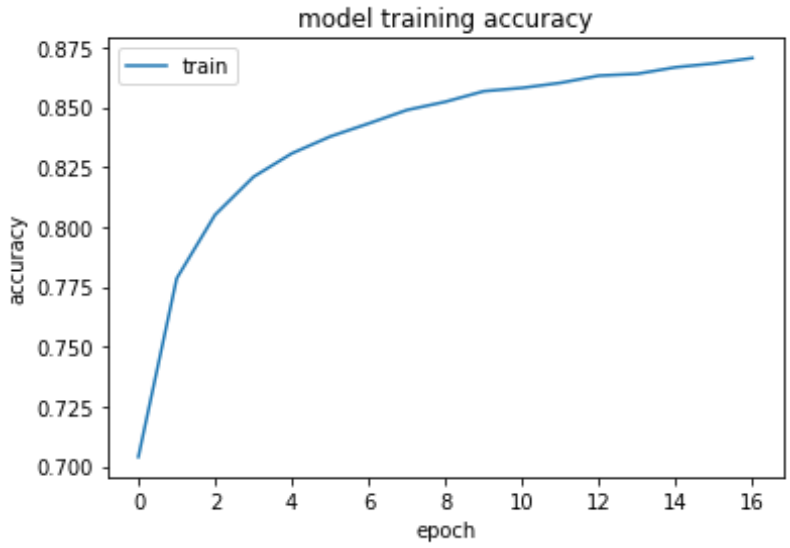

Figure 4-30 Training progress of ANN (knock vs idle).

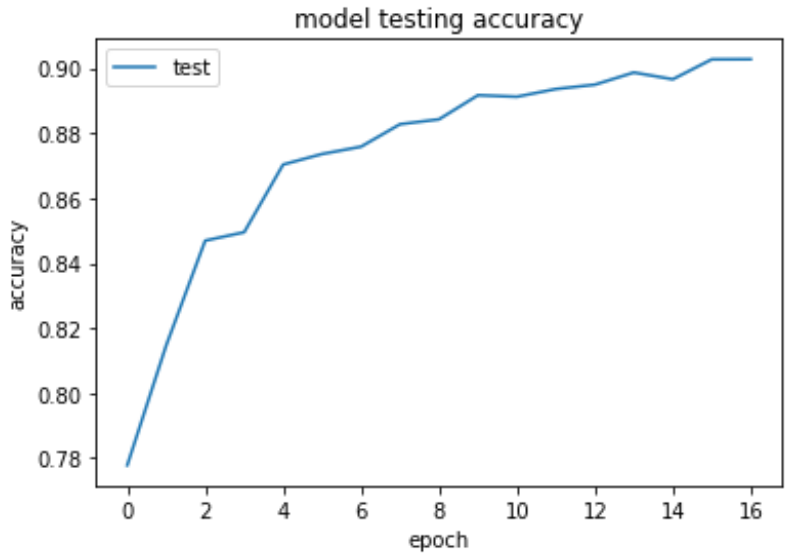

Figure 4-31 Testing accuracy of ANN (knock vs idle).

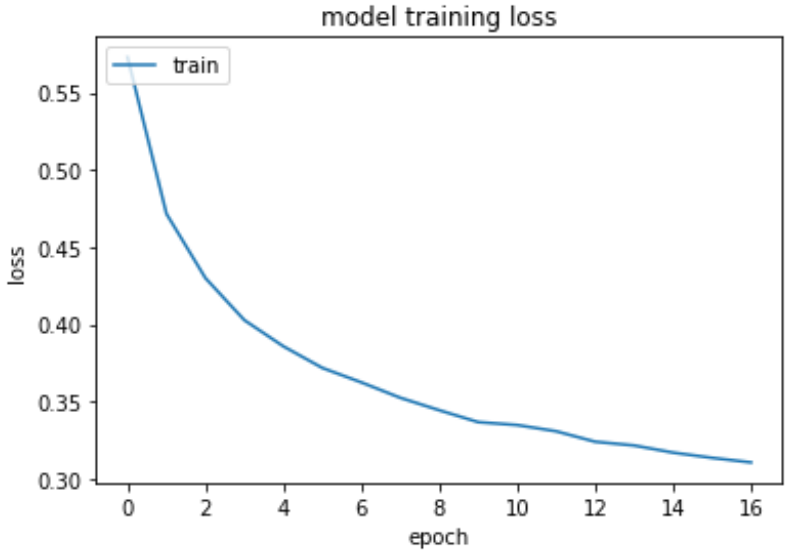

Figure 4-32 Training loss of ANN (knock vs idle).

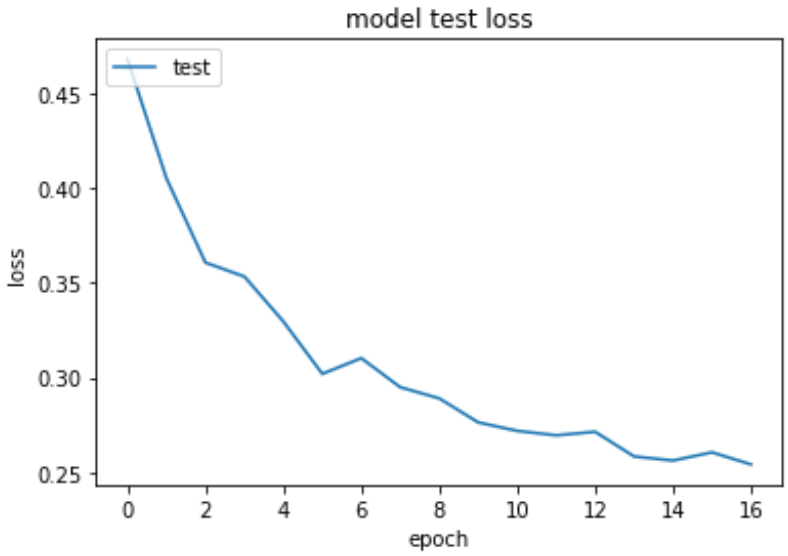

Figure 4-33 Testing loss of ANN (knock vs idle).

	precision	recall	f1-score	support
0	0.89	0.91	0.90	11964
1	0.91	0.89	0.90	12036
accuracy			0.90	24000
macro avg	0.90	0.90	0.90	24000
weighted avg	0.90	0.90	0.90	24000

Figure 4-34 Evaluation metrics of ANN (knock vs idle).

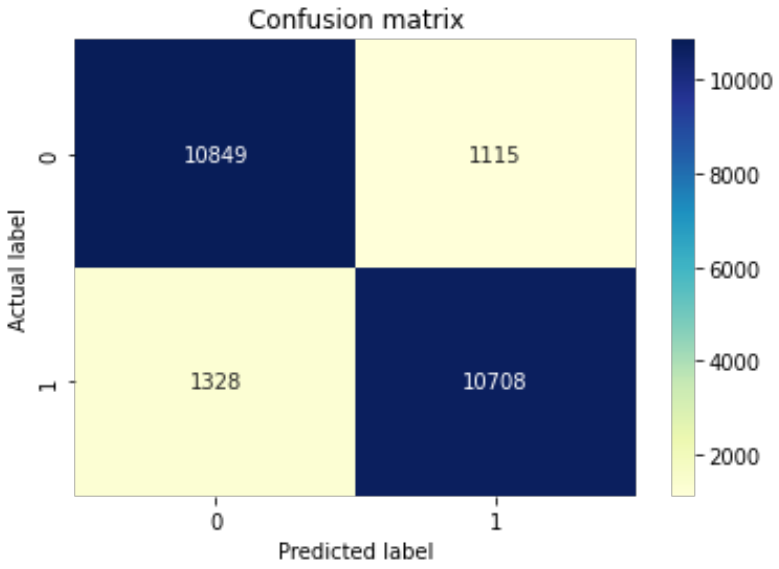

Figure 4-35 Confusion matrix for ANN (knock vs idle).

(2) ANN FOR ENGINE KNOCK VS ENGINE START USING MFCC

The average accuracy is 85.87%; it takes 2 mins and 45 sec to finish the training. Figures explains that the method has maintained good classification accuracy when compared with above ANN model it is lower by 5%.

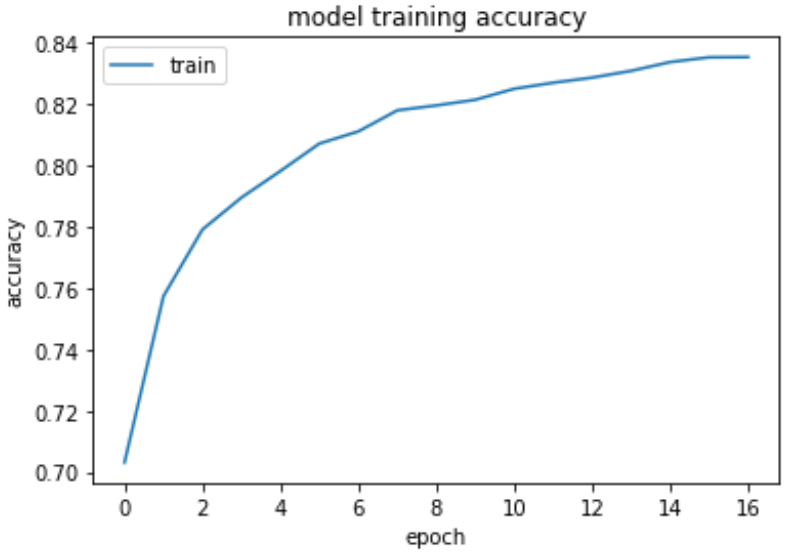

Figure 4-36 Training progress of ANN (knock vs start).

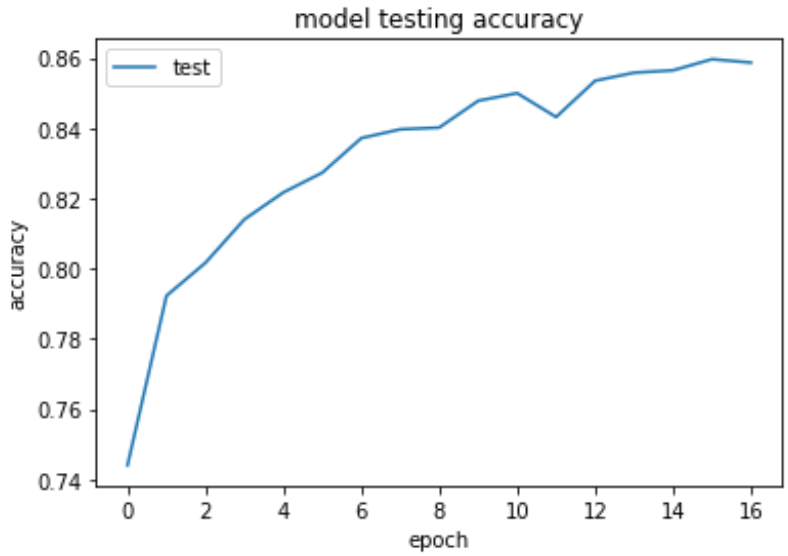

Figure 4-37 Testing accuracy of ANN (knock vs start).

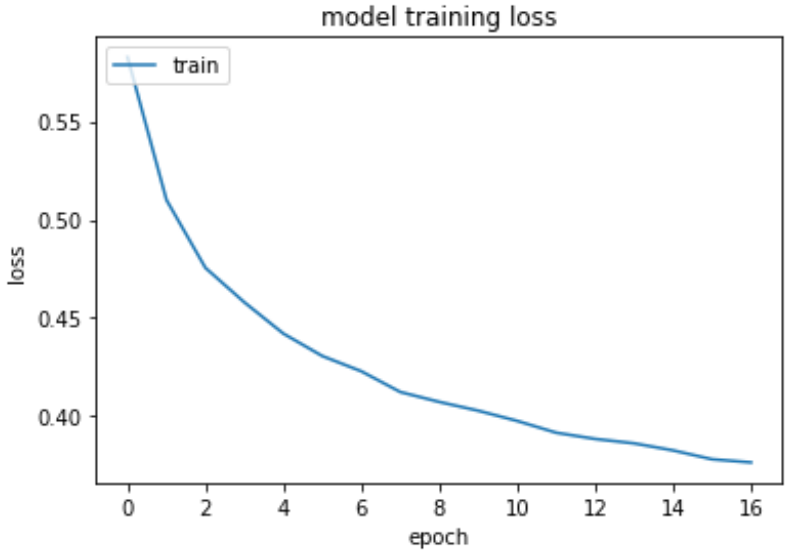

Figure 4-38 Training loss of ANN (knock vs start).

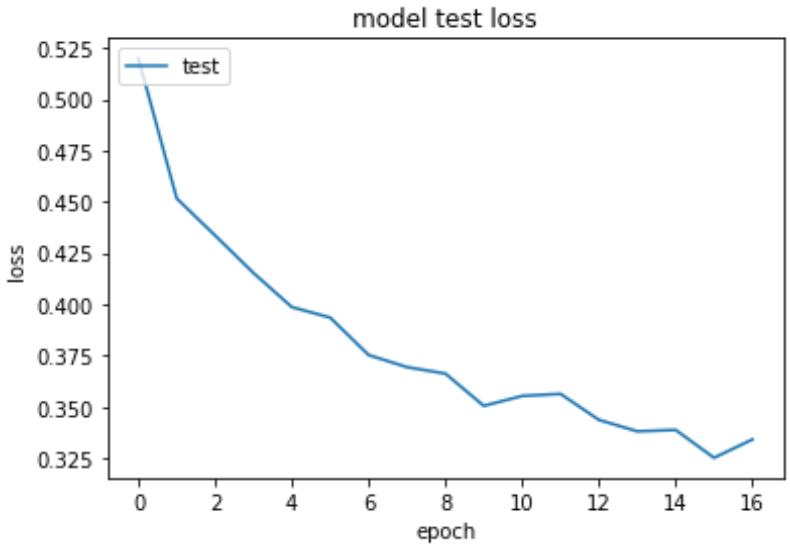

Figure 4-39 Testing loss of ANN (knock vs start).

	precision	recall	f1-score	support
0	0.78	0.93	0.85	12030
1	0.91	0.74	0.82	11970
accuracy			0.83	24000
macro avg	0.85	0.83	0.83	24000
weighted avg	0.85	0.83	0.83	24000

Figure 4-40 Evaluation metrics of ANN (knock vs start).

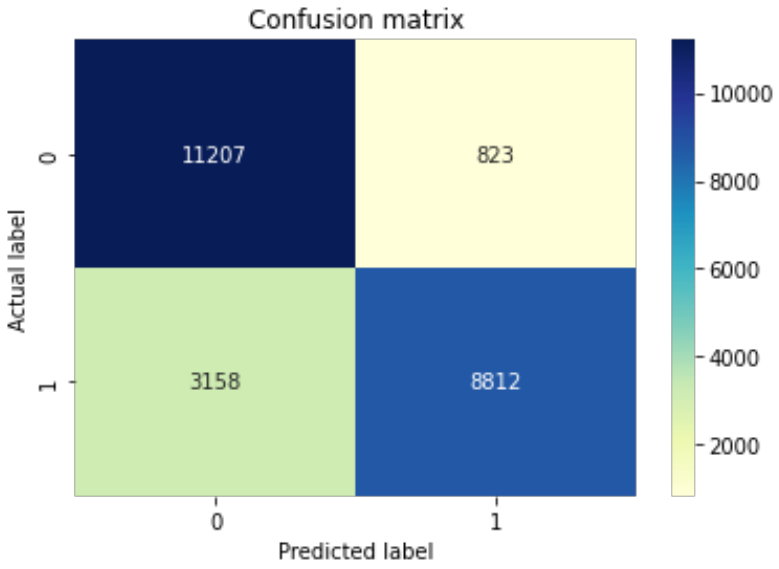

Figure 4-41 Confusion matrix for ANN (knock vs start).

(3) ANN FOR ENGINE KNOCK VS ENGINE ACCELERATION USING MFCC

The average accuracy is 90.14%. It takes 3 mins and 22 sec to finish the training. Explanation of training and testing resulted images are shown below and this method has better classification accuracy when compared with engine start and idle audio samples ANN model.

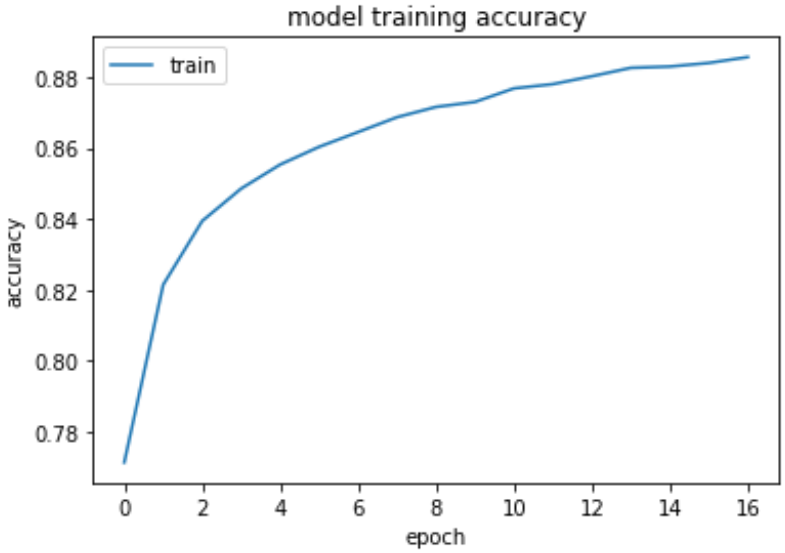

Figure 4-42 Training progress of ANN (knock vs acceleration).

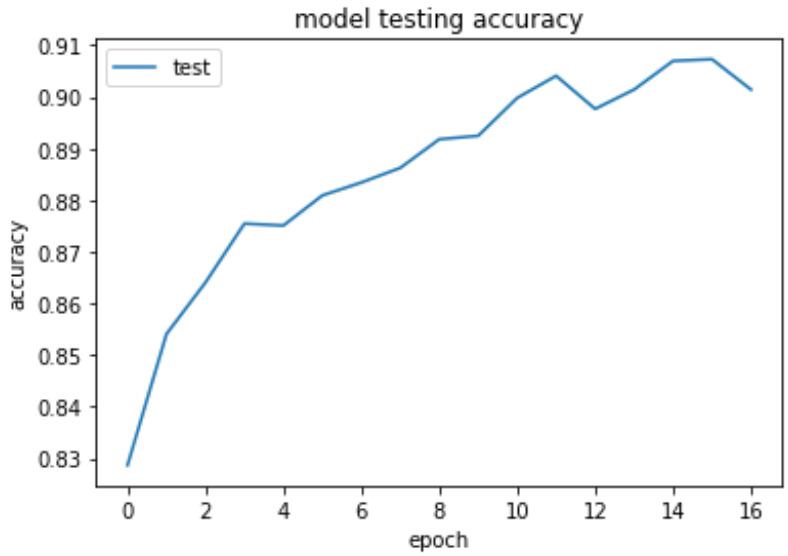

Figure 4-43 Testing accuracy of ANN (knock vs acceleration).

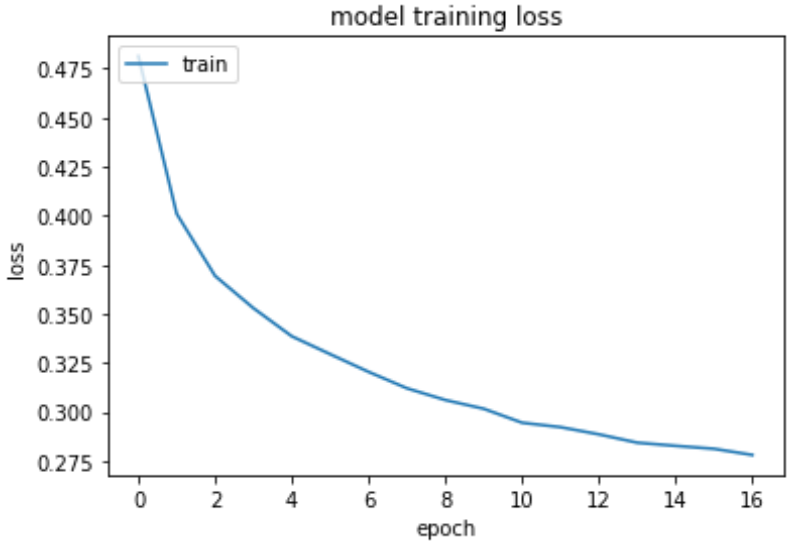

Figure 4-44 Training loss of ANN (knock vs acceleration).

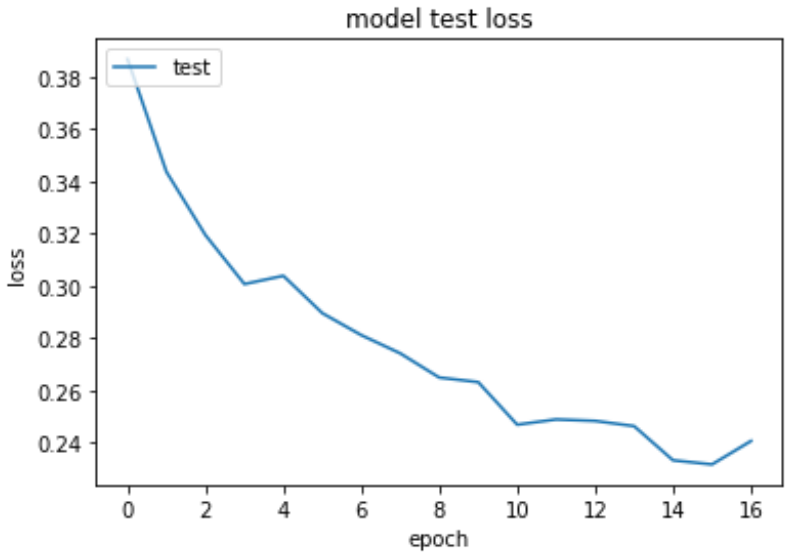

Figure 4-45 Testing loss of ANN (knock vs acceleration).

	precision	recall	f1-score	support
0	0.92	0.87	0.90	11913
1	0.88	0.93	0.91	12087
accuracy			0.90	24000
macro avg	0.90	0.90	0.90	24000
weighted avg	0.90	0.90	0.90	24000

Figure 4-46 Evaluation metrics of ANN (knock vs acceleration).

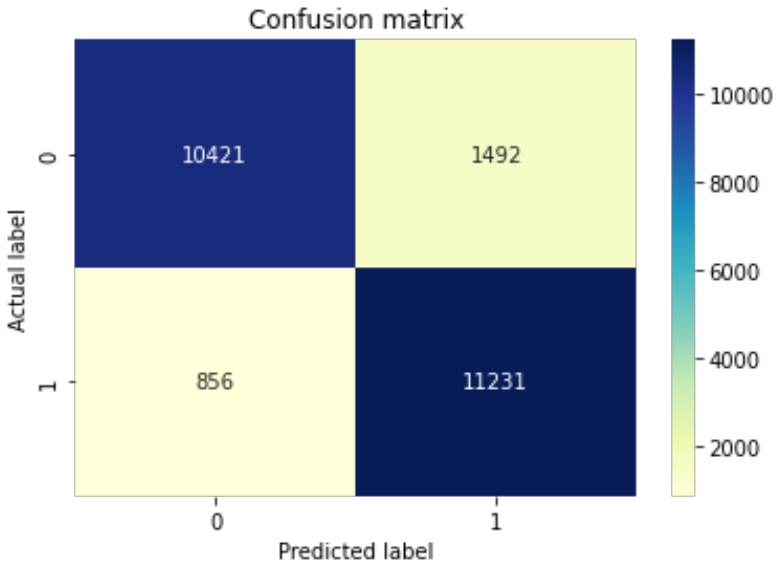

Figure 4-47 Confusion matrix for ANN (knock vs acceleration).

(4) ANN FOR MULTICLASS CLASSIFICATION USING MFCC

The average accuracy is 77.52% and it takes 17 mins and 22 sec to finish the training. From Figures 4-48 till 4-53 we can see that the method has not obtained suitable classification accuracy for all the engine audio samples ANN model.

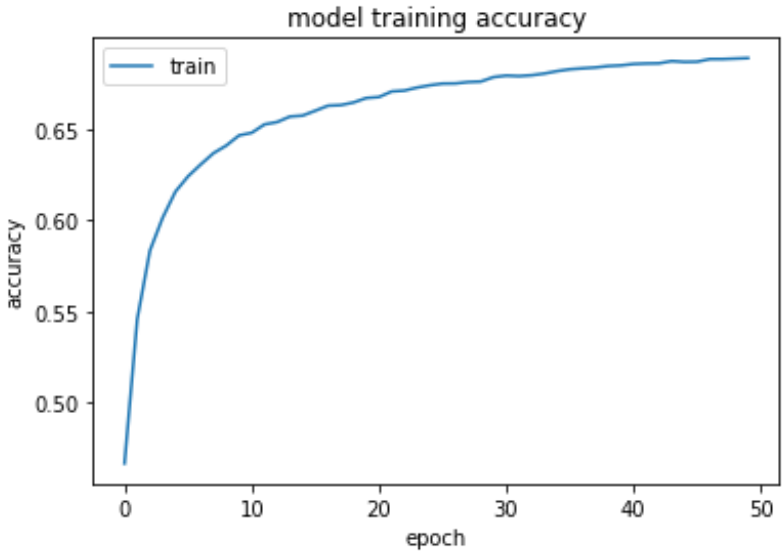

Figure 4-48 Training progress of ANN (multi-class classification).

Figure 4-49 Testing accuracy of ANN (multi-class classification).

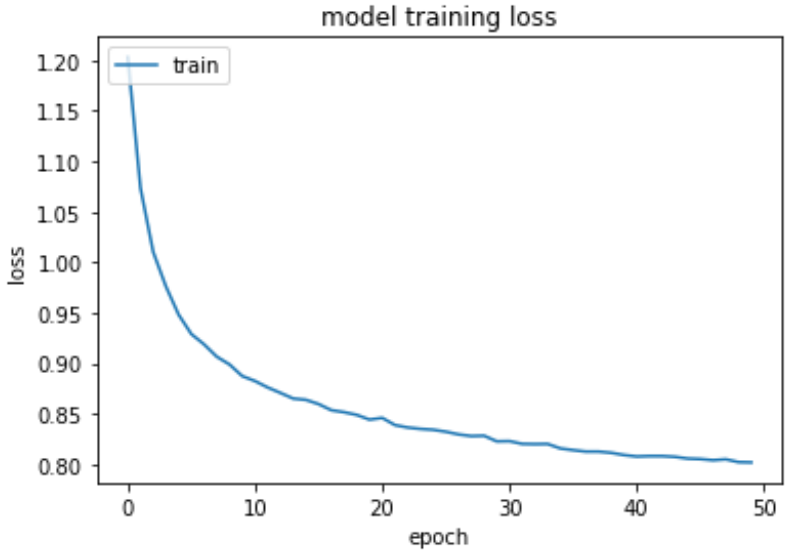

Figure 4-50 Training loss of ANN (multi-class classification).

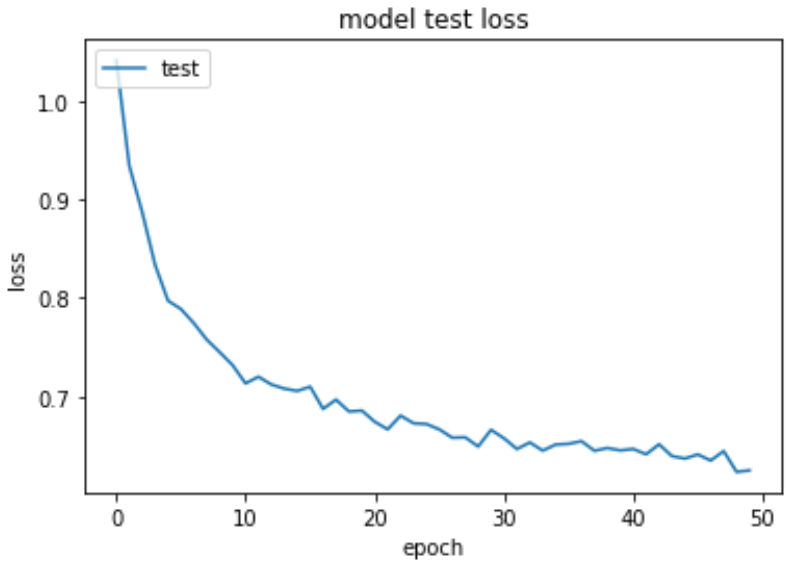

Figure 4-51 Testing loss of ANN (multi-class classification).

	precision	recall	f1-score	support
0	0.54	0.90	0.67	12058
1	0.91	0.66	0.76	11931
2	0.85	0.69	0.76	11897
3	0.83	0.65	0.73	12114
accuracy			0.73	48000
macro avg	0.78	0.73	0.73	48000
weighted avg	0.78	0.73	0.73	48000

Figure 4-52 Evaluation metrics of ANN (multi-class classification).

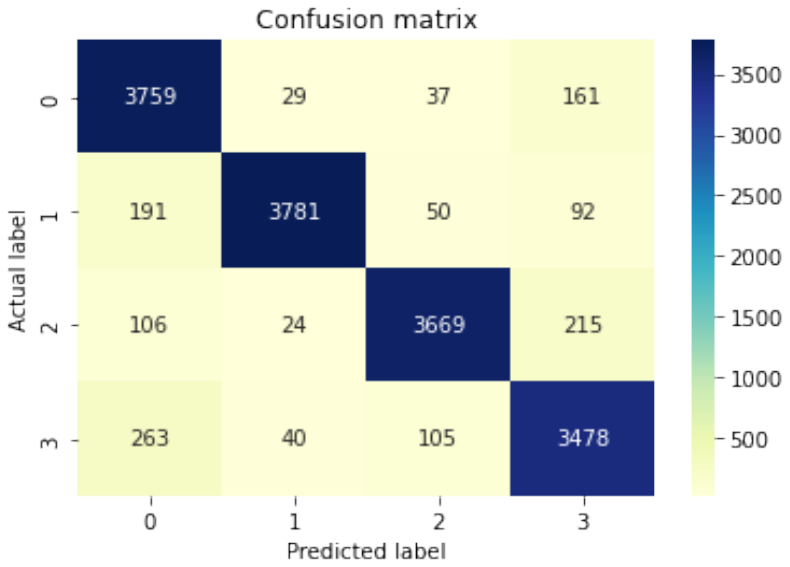

Figure 4-53 Confusion matrix of ANN (multi-class classification).

Table 4-1 lists all the models together to illustrate which gives the best result for fault diagnosis in the engine. Note that we have used binary classification and multi-class classification techniques to detect knocking from engine audio samples. ANN models will be chosen from both methods used.

Table 4-1 Accuracies of ANN algorithms

Audio Samples	Class	Preprocessing	Accuracy	F1
Knock Vs Idle	Binary	FFT	96.17	96.00
Knock Vs Start	Binary	FFT	92.80	93.00
Knock Vs Acceleration	Binary	FFT	95.06	94.00
All four classes	Multi	FFT	92.18	92.00
Knock Vs Idle	Binary	MFCC	90.28	90.00
Knock Vs Start	Binary	MFCC	85.87	85.00
Knock Vs Acceleration	Binary	MFCC	90.14	90.00
All four classes	Multi	MFCC	77.52	73.00

The results show that the signal following feature extraction can be used as the input of the ANN algorithm to achieve comparatively high accuracy, with the binary model of knock and idle audio, fault distinguishable from idle sound, and decent accuracy in multi-class. When compared to MFCC, FFT is the best preprocessing method for detecting knocking.

We discovered from section 3.2 that MFCC is appropriate for human voice patterns because it can detect differences in pitch from audio samples. Music instruments can also be classified using MFCC; in our ANN models, there is no significant difference between MFCC and FFT; in fact, FFT has won the battle by a significant margin, as seen in ANN models of MFCC and FFT.

5. DESIGN OF 2D CONVOLUTIONAL NEURAL NETWORK

5.1 2D-CNN Structure

The convolutional neural network (CNN) is one of the most popular deep neural networks. It derives its name from the linear convolution operation between matrices in mathematics. Convolutional, non-linear, pooling, and fully connected layers are among the many layers that make up CNN. Pooling and non-linearity layers lack parameters, but convolutional and fully connected layers do. In machine learning issues, CNN performs quite well. Particularly the image-related applications, such as the largest image classification data set (Image Net), computer vision, and natural language processing (NLP), and the outcomes obtained were truly astounding [45].

Four components are typically required to build a 2D-CNN model. Convolution is a critical step in the feature extraction process. Convolutional outputs are known as feature maps. We will lose information on the border if we use a convolution kernel of a certain size. As a result, padding is introduced to enlarge the input with a zero value, which can indirectly adjust the size. Furthermore, stride is used to control the density of convolving. The lower the density, the longer the stride. Following convolution, feature maps contain many features, which are prone to overfitting. As a result, pooling, including max pooling and average pooling, is proposed to eliminate redundancy [46].

The neural model's parameters must be adjusted to the necessary degree. Gradient descent is typically applied in this situation to ensure the highest level of precision. A first-order iterative optimization procedure for resolving the local minima of differentiable functions is the gradient descent method (GDM). It is frequently employed in machine learning to resolve least squares issues. Since the steepest fall is in the direction of the gradient at the current position of the function, it makes sense to repeat the steps in that direction. The data from known solutions are included in the training set, and the neural network model can be tuned to a reasonably accurate level [47].

In our research we have used the standard Adam as the learning rate to improve the accuracy using gradient descent. Keras library has been used to implement this optimizer with a categorical cross-entropy loss function for the network.

Although 2D-CNN has primarily been used for image classification [48], we have also used it to classify audio, and the results are very remarkable. Architecture that we have used for this task can be seen in Figure 5-1

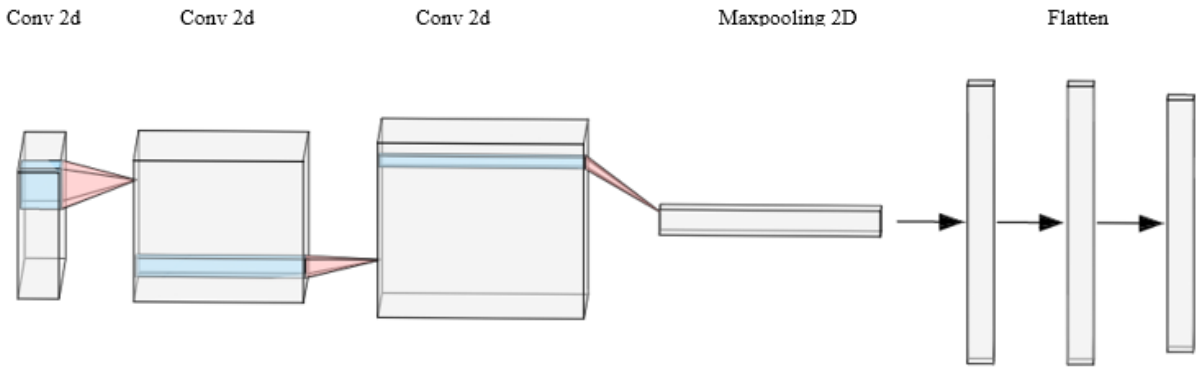

Figure 5-1 2D-CNN architecture for audio classification.

With the introduction of large datasets such as ImageNet, image classification performance has greatly improved using convolutional neural network (CNN) architectures such as AlexNet, VGG, Inception, and ResNet [49]. We have decided to make our basic design to characterize sound examples, our examination centers around information instead of deep learning networks.

Table 5-1 Information of 2D-CNN Layers

Layer name	**CNN Models (Kernel Filter, Activation Function, Strides, Padding)**
C1	Conv 2D (32*3*3, ReLU, 1, None)
C2	Conv 2D (48*3*3, ReLU, 1, None)
C3	Conv 2D (128*3*3, ReLU, 1, None)
MXP3	Maxpooling 2D (2,2,1)
FC1	Fullyconnect (128, ReLU)
FC2	Fullyconnect (64, ReLU)
Output	SoftMax, Classification

To extract high-level information, we use triplet sequential convolutional and merging levels. Each convolutional layer employs the ReLU activation function. Three convolution layers are applied before the maximum pooling process. Finally, the adoption of two fully interconnected layers, each with the ReLU activation function. A soft-max layer is used to achieve the classified output.

5.2 Experimental Results

We have used the same format as ANN in 2D-CNN to classify audio files. Simple architecture has been used to classify, when it comes to preprocessing stage which stands as central point of our research. In preprocessing stage, we have STFT and MFCC to form input for the 2D-CNN network, dependent vehicle feature extraction has been deployed. To store the data, we have used data frames and in addition to ANN steps we have separated testing and training data while capturing each class data.

This architecture also performs binary classification and multi-class classification to compare it to all previous models produced by ANN. The architecture for multi-class and binary classes does not change, but the activation function at the final layer does. We used the sigmoid activation function for binary and the SoftMax activation function for multi-class.

In general, 2D-CNN is a more potent and accurate method of classifying data and convolutional neural networks (CNN) are among the most widely used models today. This neural network computational model employs a multilayer perceptron variation and includes one or more convolutional layers that can be entirely connected or pooled. These convolutional layers generate feature maps that capture a region of the image, which is then divided into rectangles and sent out for nonlinear processing. Since ANN executes more quickly than other deep learning networks like CNN, it continues to be the industry standard for issues with small datasets and no requirement for visual inputs. Instead of using photos as the network's input, we preprocessed the data into data frames to increase 2D-CNN's efficacy and decrease the network's execution time. These data frames can be used in the same way as images and converted back to images, as we typically do with CNN issues.

2D-CNN is frequently designed with dense layers [50], which we did not use in our model; instead, we used a simple architecture. CNN has an advantage over ANN in terms of accuracy; ANN only accepts tabular data, whereas CNN can also work with images [51].

5.2.1 Results for 2D-CNN using STFT feature extraction

The 2D-CNN model has been tested with engine-knocking audio samples for each class at 17 iterations, and the multi-class model has also been validated at 50 iterations.

(1) 2D-CNN FOR ENGINE KNOCK VS ENGINE IDLE USING STFT

The average accuracy is 95.08%, and it takes 1 mins and 35 sec to finish the training which is very less than ANN due to its simple structure. This method has excellent classification accuracy.

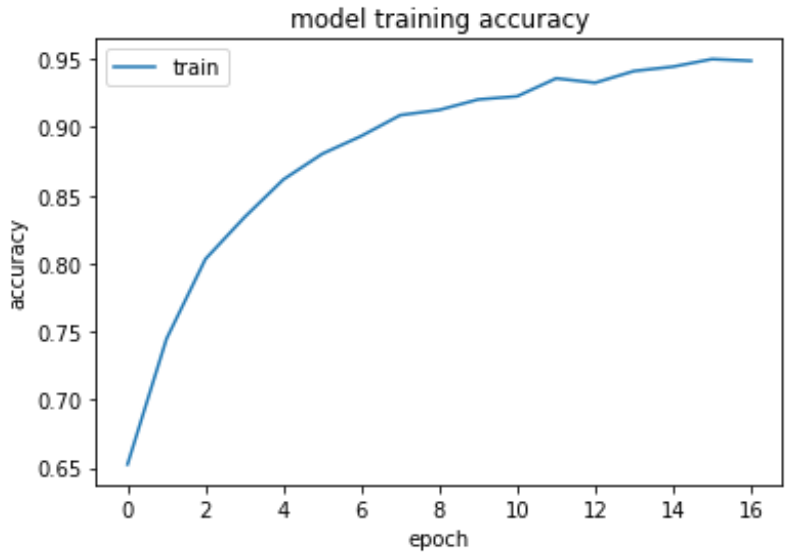

Figure 5-2 Training progress of 2D-CNN (knock vs idle).

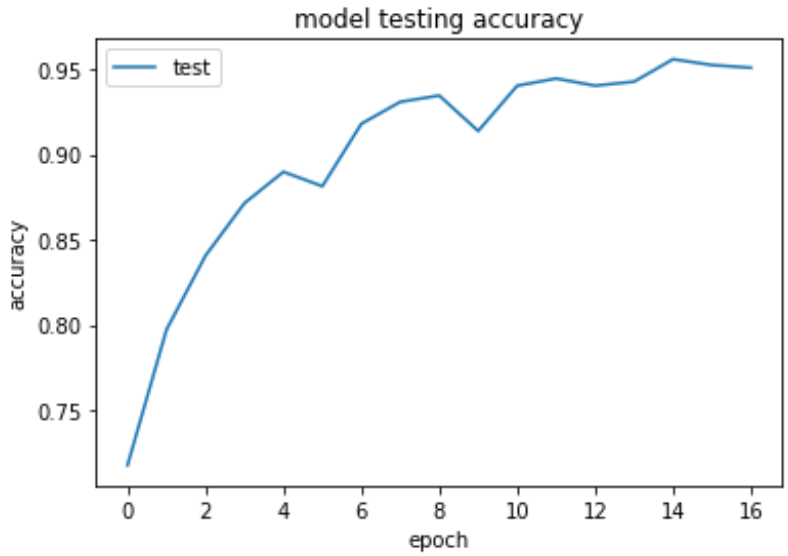

Figure 5-3 Testing accuracy of 2D-CNN (knock vs idle).

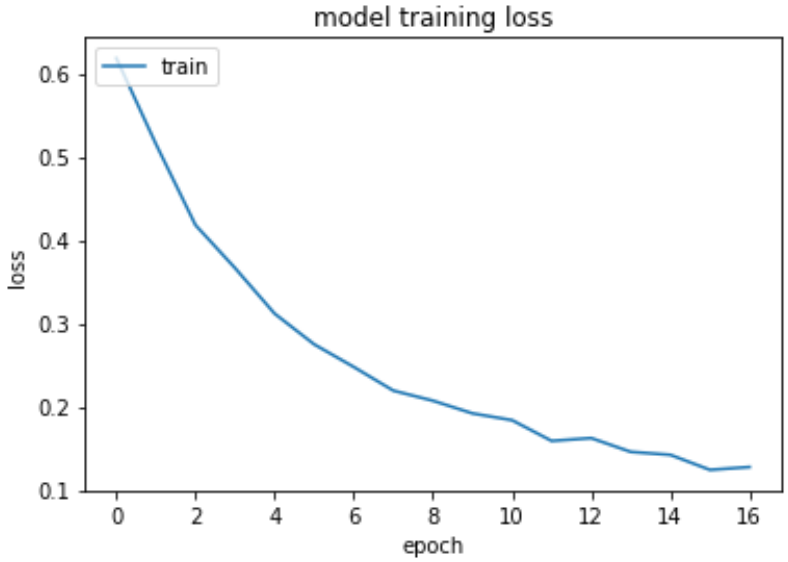

Figure 5-4 Training loss of 2D-CNN (knock vs idle).

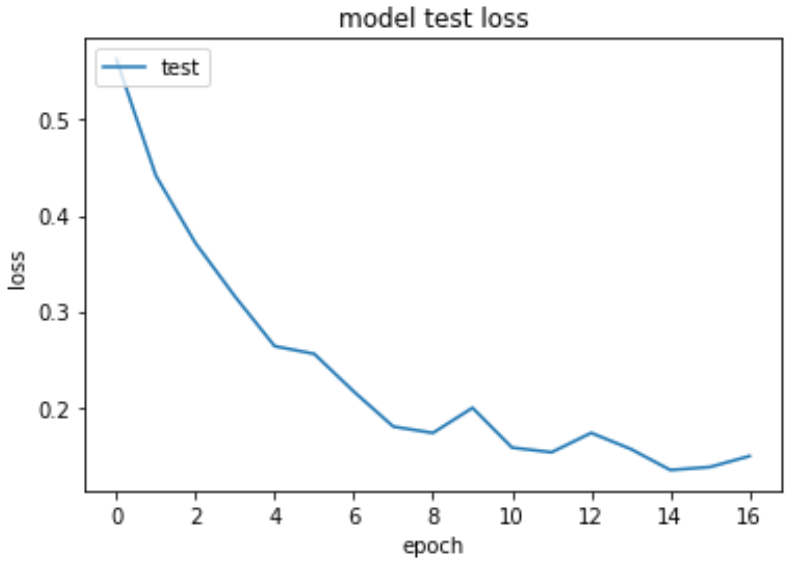

Figure 5-5 Testing loss of 2D-CNN (knock vs idle).

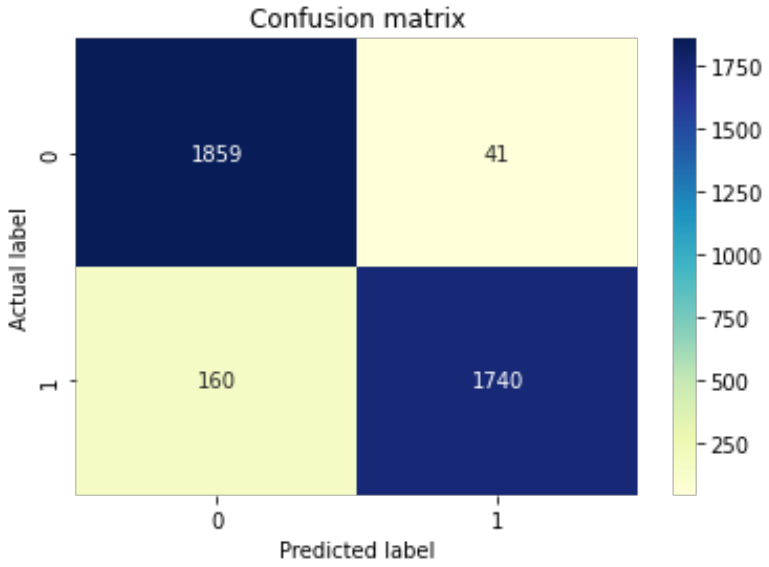

Figure 5-6 Confusion matrix for 2D-CNN (knock vs idle).

(2) 2D-CNN FOR ENGINE KNOCK VS ENGINE START USING STFT

The average accuracy is 92.29%; it takes 1 mins and 44 sec to finish the training. From Figures 5-7 till 5-11 we can see that the method has lesser classification accuracy when compared with above idle sound model and it converges in about 17 epochs.

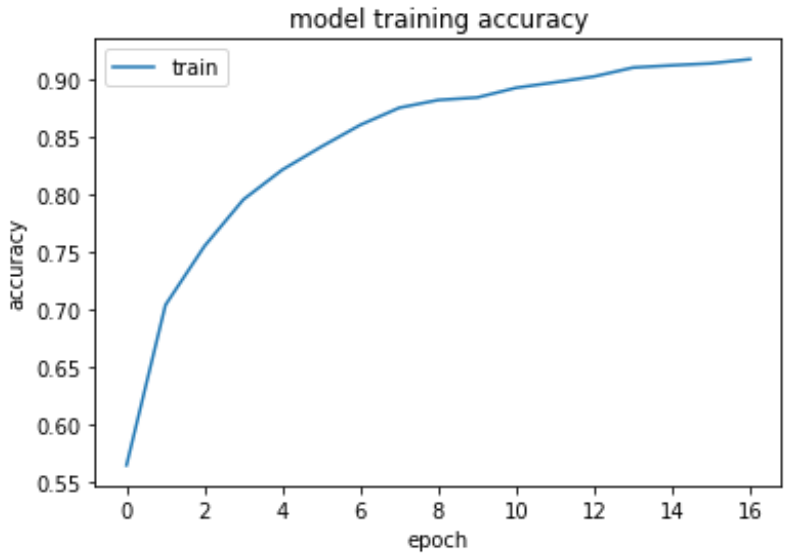

Figure 5-7 Training progress of 2D-CNN (knock vs start).

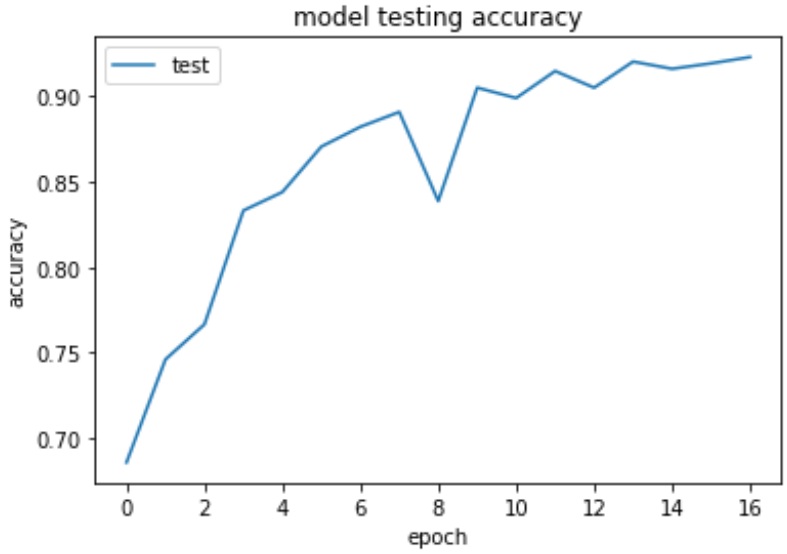

Figure 5-8 Testing accuracy of 2D-CNN (knock vs start).

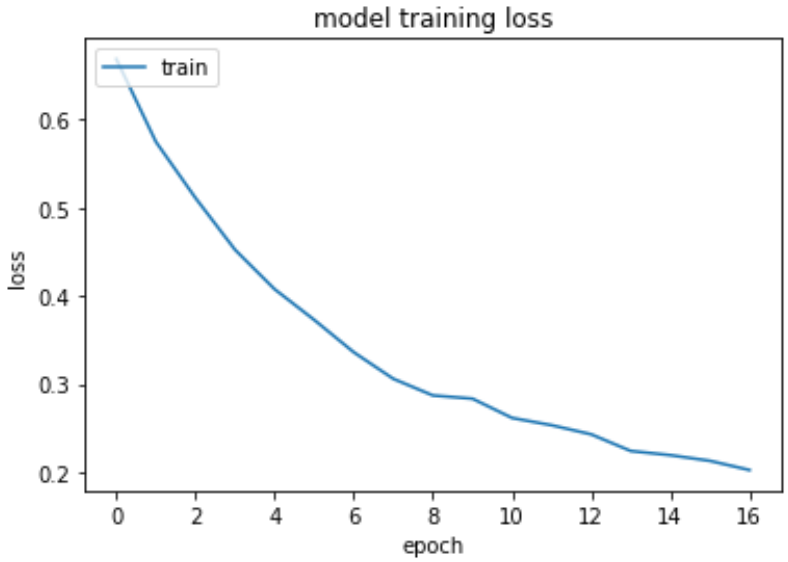

Figure 5-9 Training loss of 2D-CNN (knock vs start).

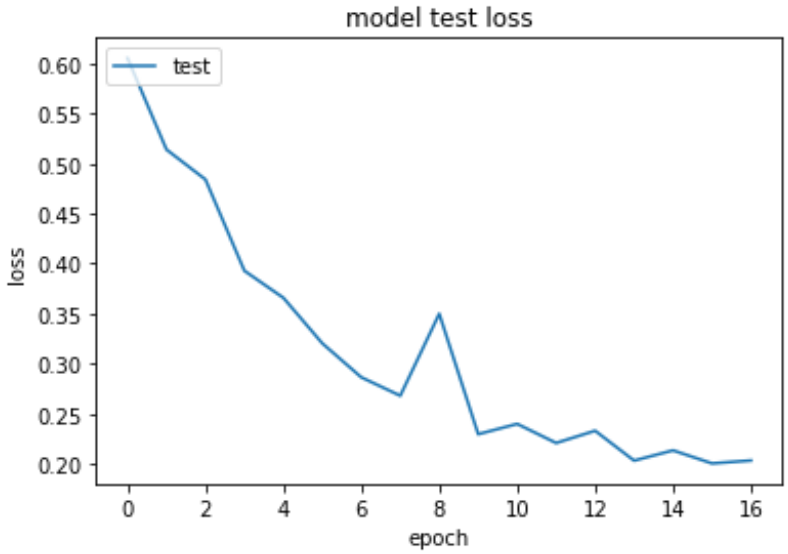

Figure 5-10 Testing loss of 2D-CNN (knock vs start).

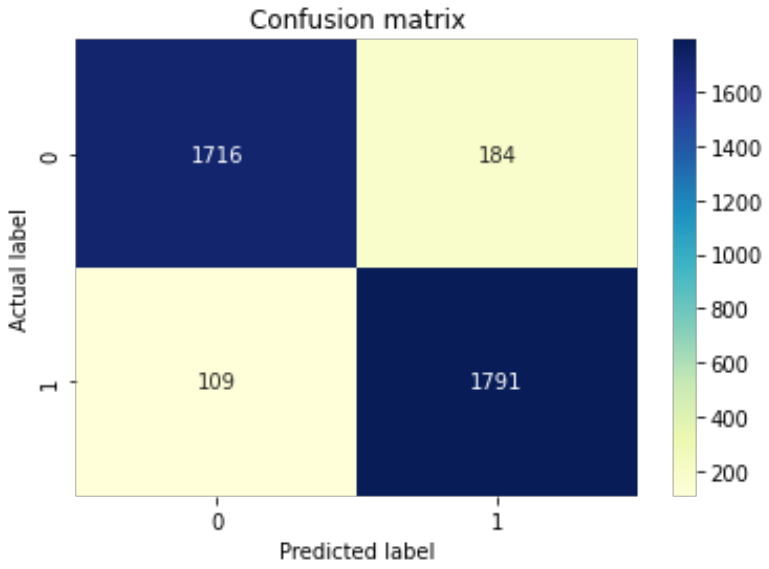

Figure 5-11 Confusion matrix for 2D-CNN (knock vs idle).

(3) 2D-CNN FOR ENGINE KNOCK VS ENGINE ACCELERATION USING STFT

The average accuracy is 94.79%. It takes 1 mins and 56 sec to finish the training. From the Figures below we can see that the method has better classification accuracy when compared with engine start but not with idle audio samples ANN model.

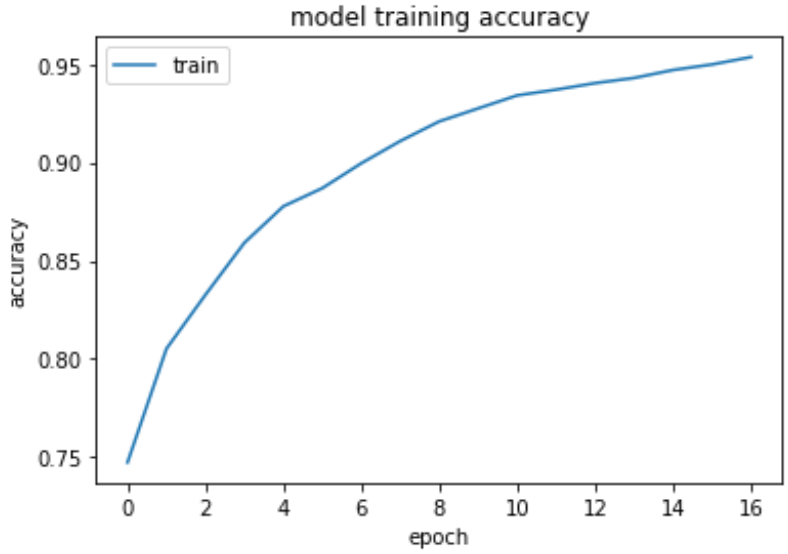

Figure 5-12 Training progress of 2D-CNN (knock vs acceleration).

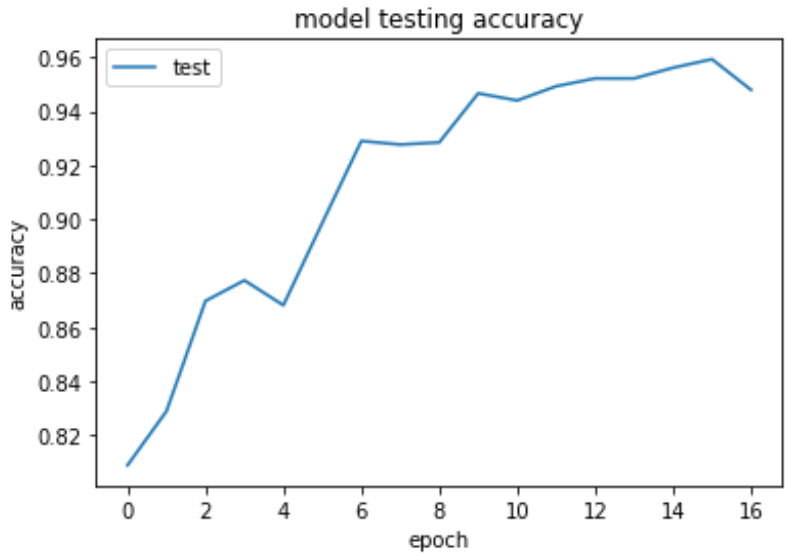

Figure 5-13 Testing accuracy of 2D-CNN (knock vs acceleration).

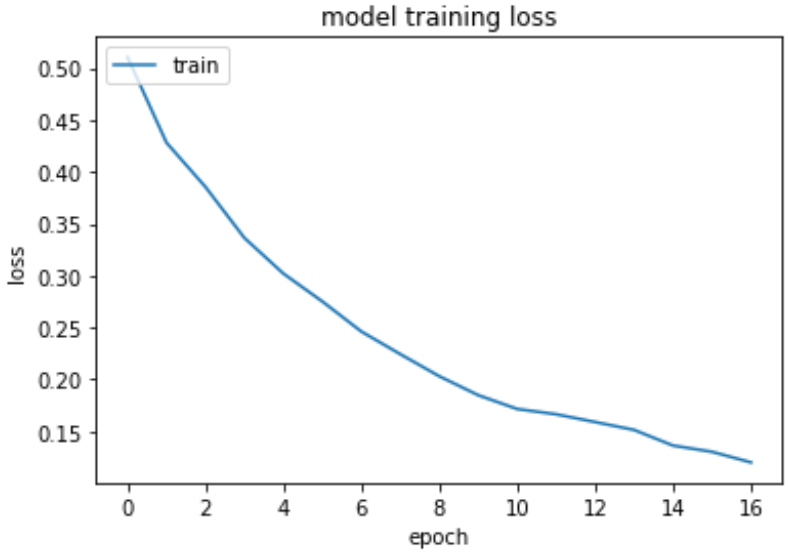

Figure 5-14 Training loss of 2D-CNN (knock vs acceleration).

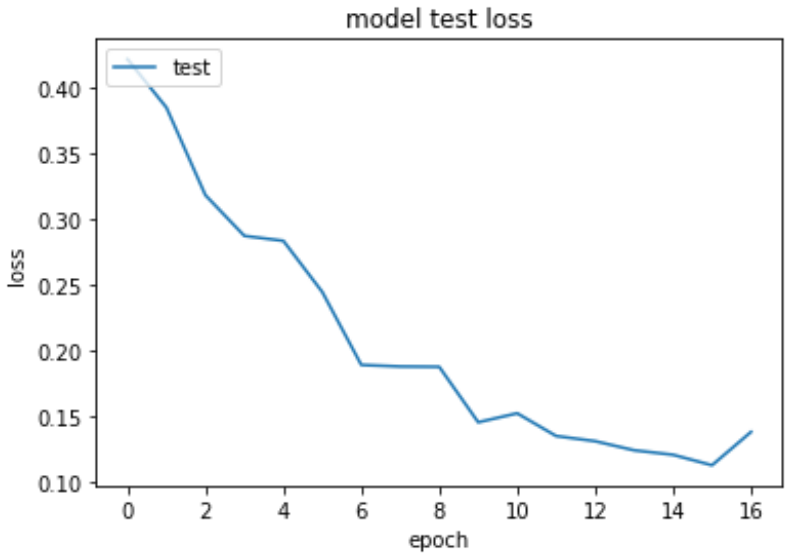

Figure 5-15 Testing loss of 2D-CNN (knock vs acceleration).

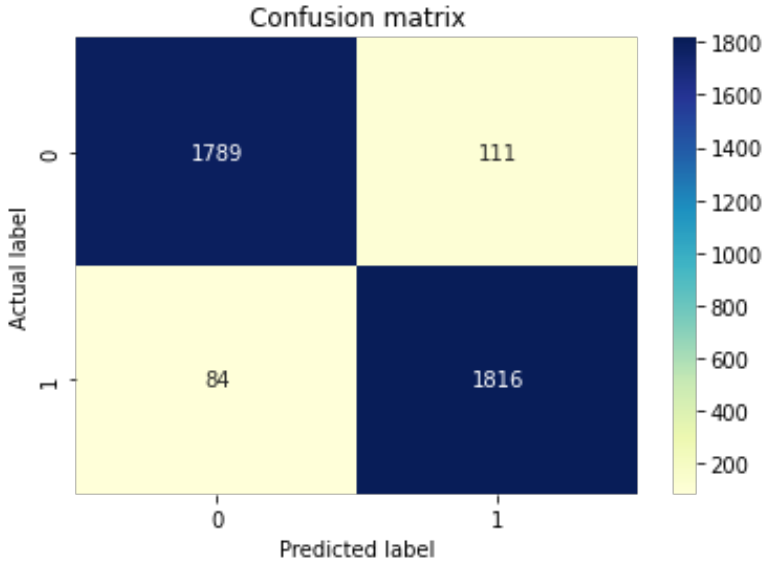

Figure 5-16 Confusion matrix for 2D-CNN (knock vs acceleration).

(4) 2D-CNN FOR MULTICLASS CLASSIFICATION USING STFT

The average accuracy is 90.70%; it takes 7 mins and 88 sec to finish the training. As we can see, the method has the finest classification accuracy for all the engine audio samples ANN model, and an illustration of its validation can be seen below in Figures.

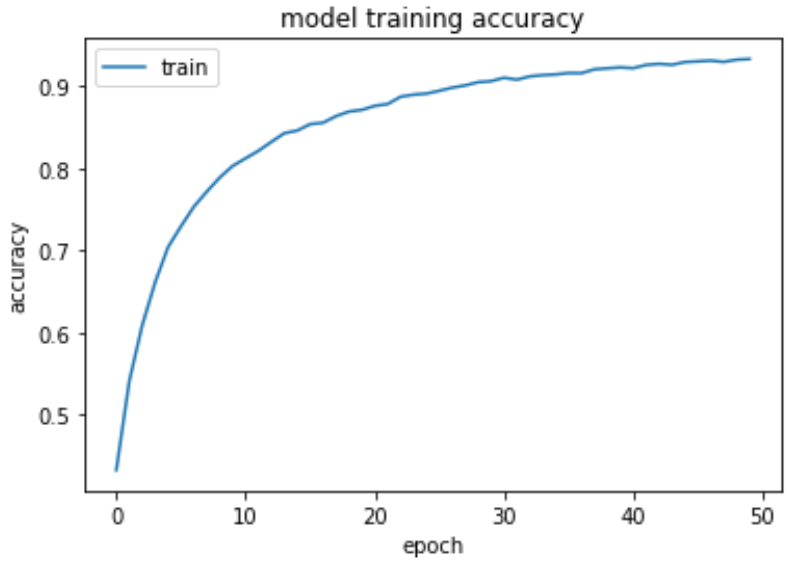

Figure 5-17 Training progress of 2D-CNN (multi-class classification).

Figure 5-18 Testing accuracy of 2D-CNN (multi-class classification).

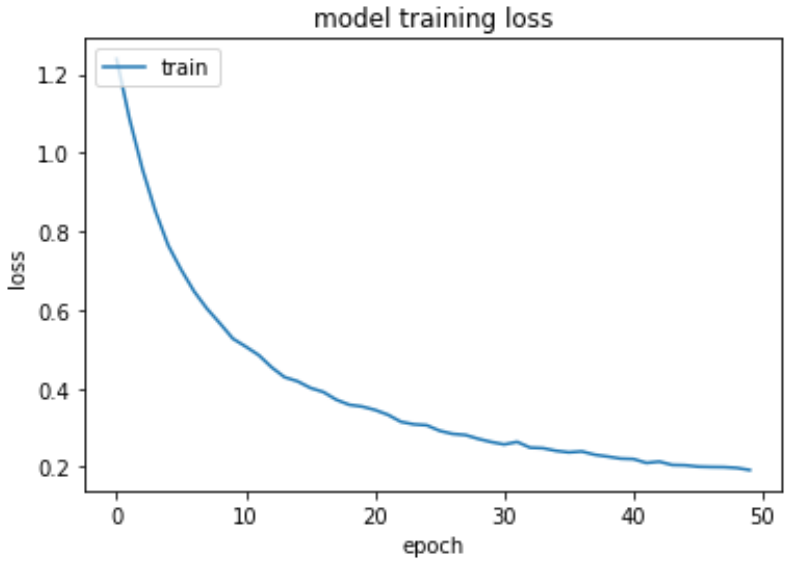

Figure 5-19 Training loss of 2D-CNN (multi-class classification).

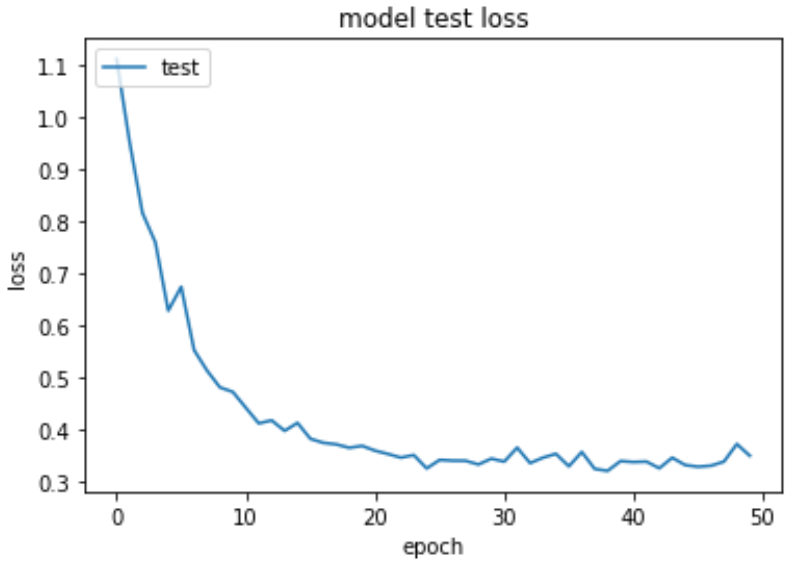

Figure 5-20 Testing loss of 2D-CNN (multi-class classification).

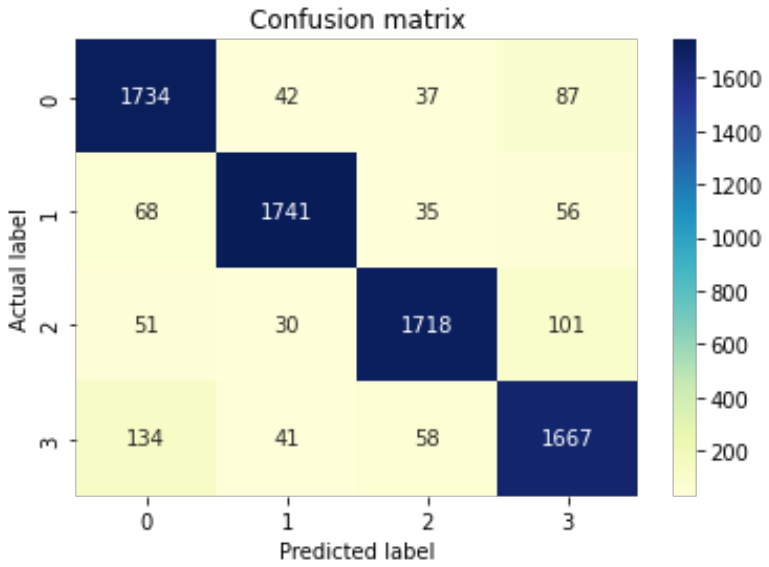

Figure 5-21 Confusion matrix of 2D-CNN (multi-class classification).

5.2.2 Results for 2D-CNN using MFCC feature extraction

We know that MFCC is appropriate for human speech, but we used it to the engine sound samples, which generated intriguing results. FFT has produced some fantastic results not only for binary classification, but also for multi-class classification, and epochs in this model are same as FFT for both classification methods.

(1) 2D-CNN FOR ENGINE KNOCK VS ENGINE IDLE USING MFCC

The average accuracy is 96.82% and it takes 1 mins and 05 sec to finish the training. Figures below show that the method has produced better classification accuracy when compared it with STFT.

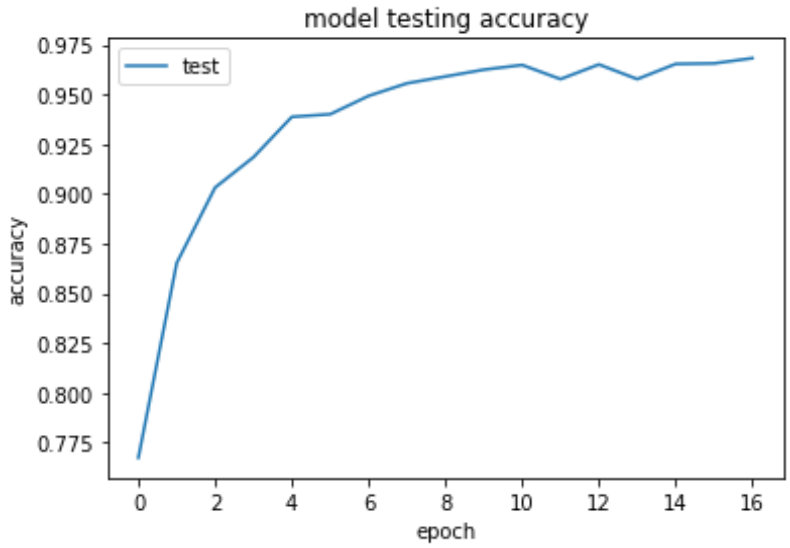

Figure 5-22 Training progress of 2D-CNN (knock vs idle).

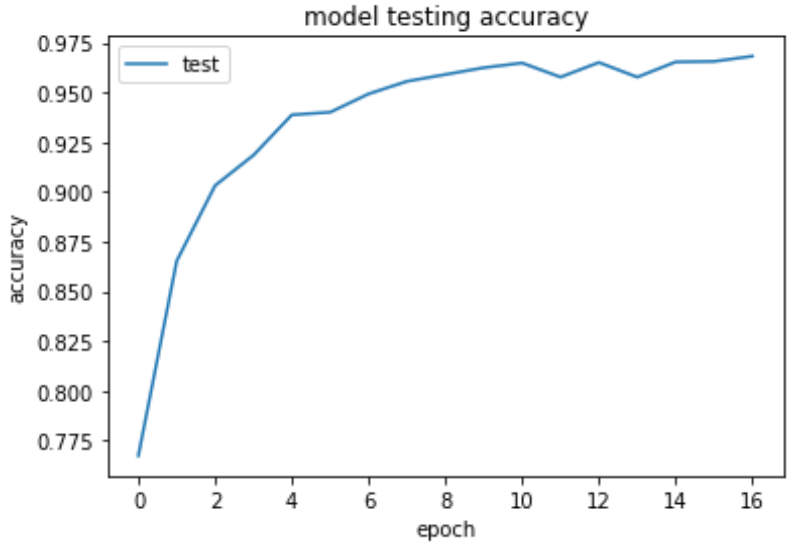

Figure 5-23 Testing accuracy of 2D-CNN (knock vs idle).

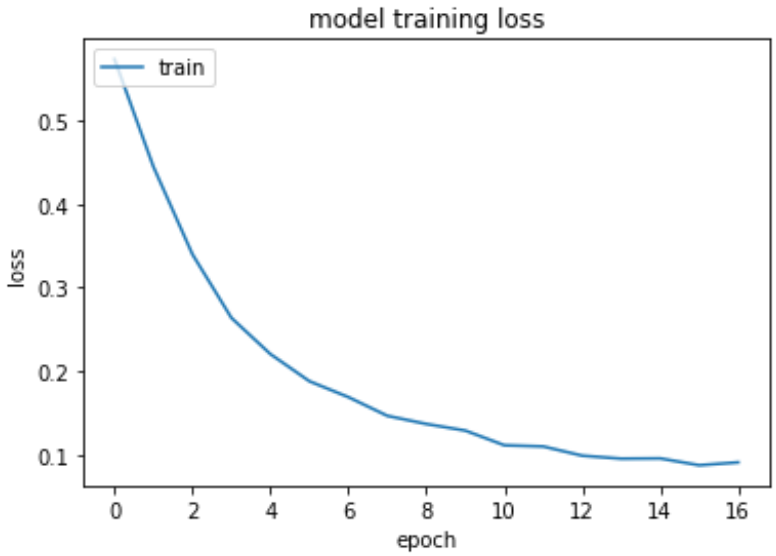

Figure 5-24 Training loss of 2D-CNN (knock vs idle).

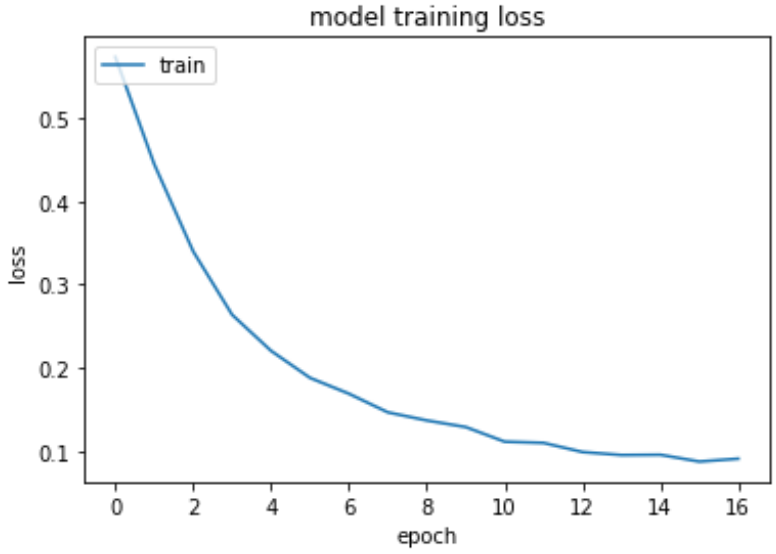

Figure 5-25 Testing loss of 2D-CNN (knock vs idle).

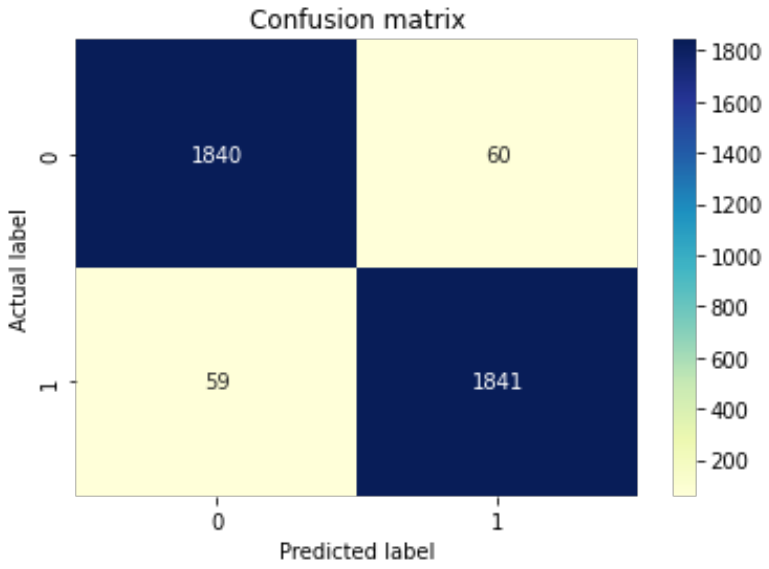

Figure 5-26 Confusion matrix for 2D-CNN (knock vs idle).

(2) ANN FOR ENGINE KNOCK VS ENGINE START USING MFCC

The average accuracy is 94.47%; it takes 1 mins and 15 sec to finish the training. This method has maintained good classification accuracy when compared with the above model as it is lower by small margin.

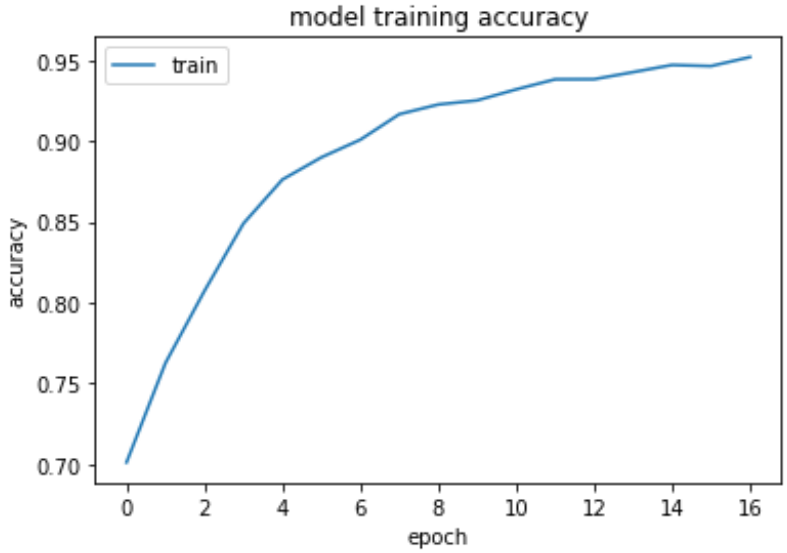

Figure 5-27 Training progress of 2D-CNN (knock vs start).

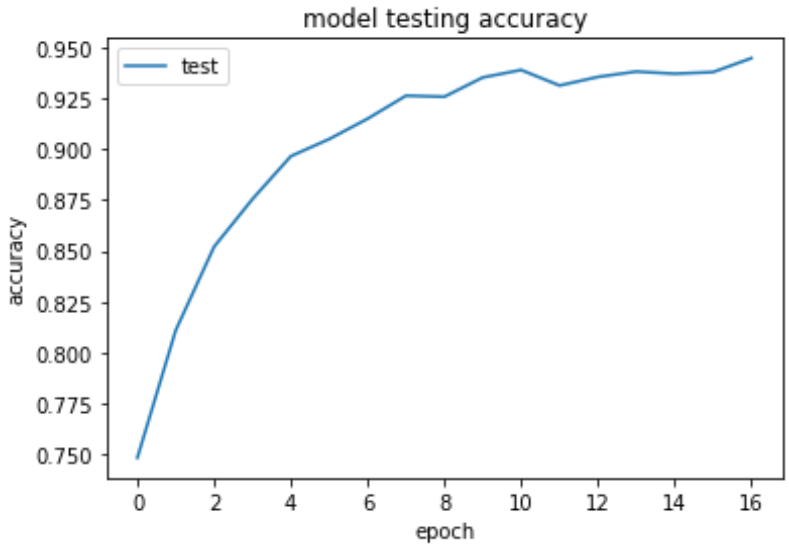

Figure 5-28 Testing accuracy of 2D-CNN (knock vs start).

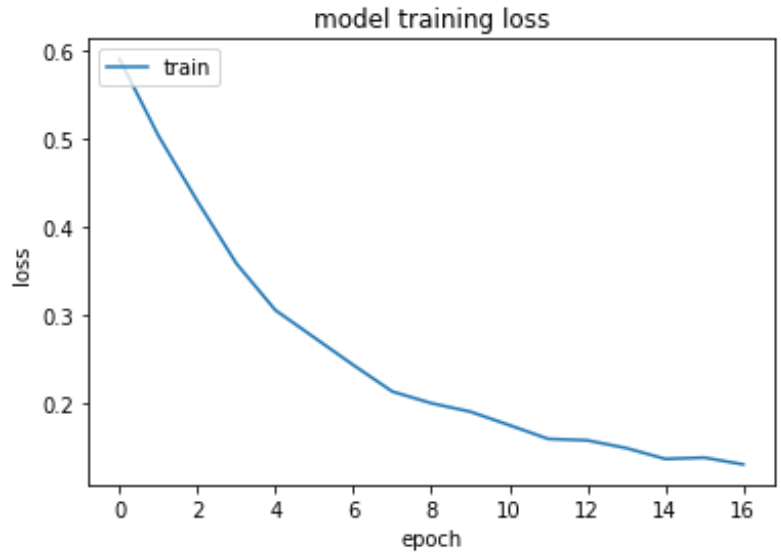

Figure 5-29 Training loss of 2D-CNN (knock vs start).

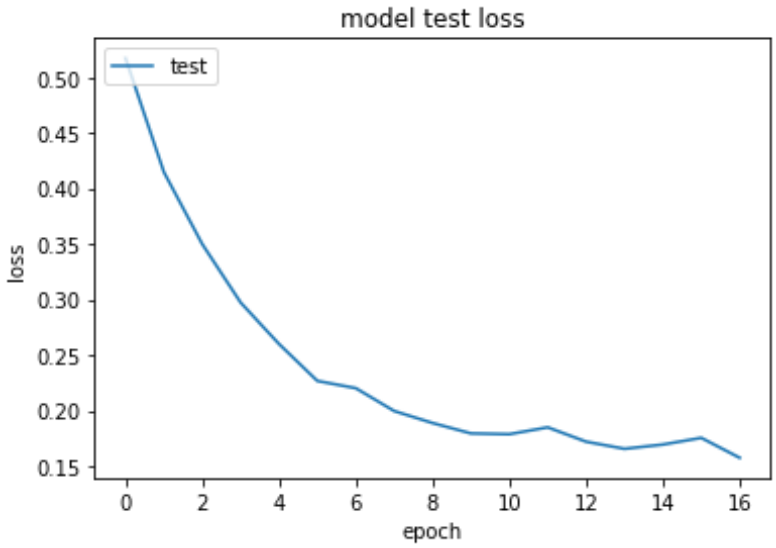

Figure 5-30 Testing loss of 2D-CNN (knock vs start).

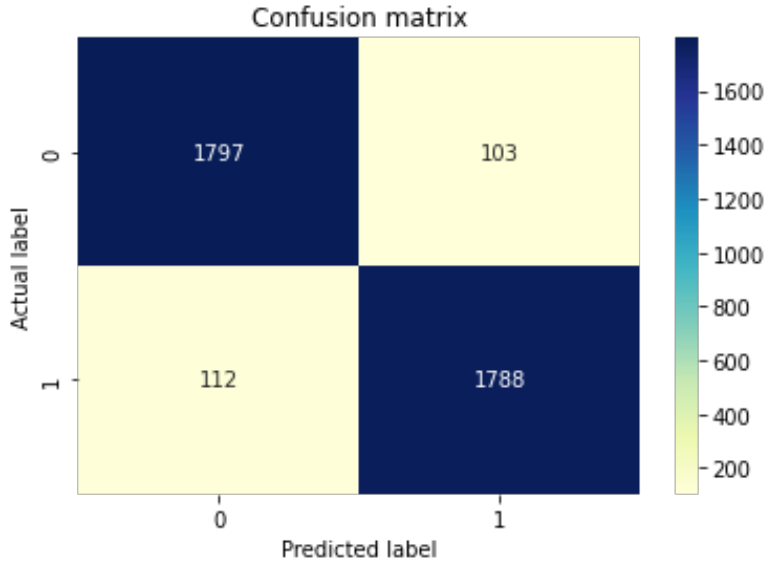

Figure 5-31 Confusion matrix for 2D-CNN (knock vs start).

(3) 2D-CNN FOR ENGINE KNOCK VS ENGINE ACCELERATION USING MFCC

The average accuracy is 95.56%; it takes less than a minute to finish the training. This method has better classification accuracy when compared with engine start but idle audio samples model still stands as best.

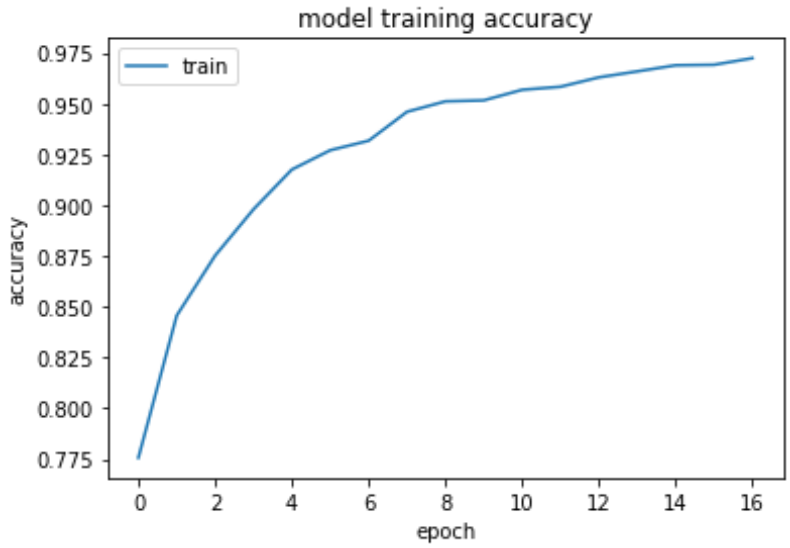

Figure 5-32 Training progress of 2D-CNN (knock vs acceleration).

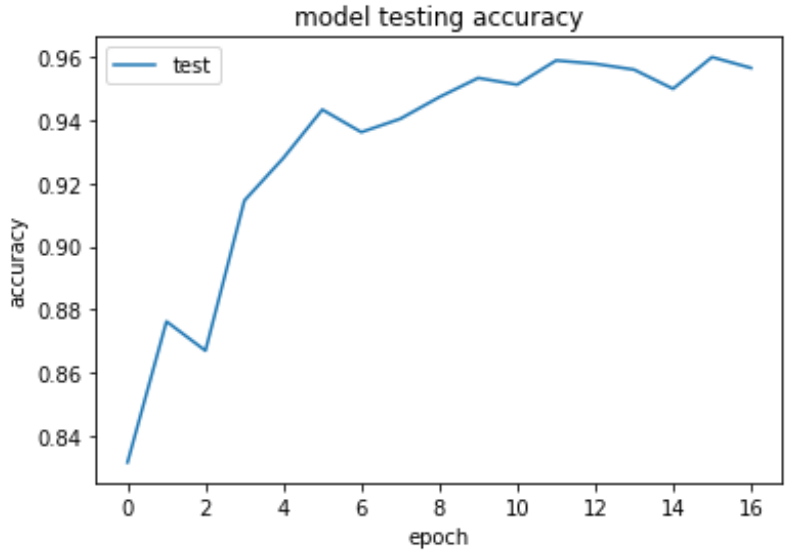

Figure 5-33 Testing accuracy of 2D-CNN (knock vs acceleration).

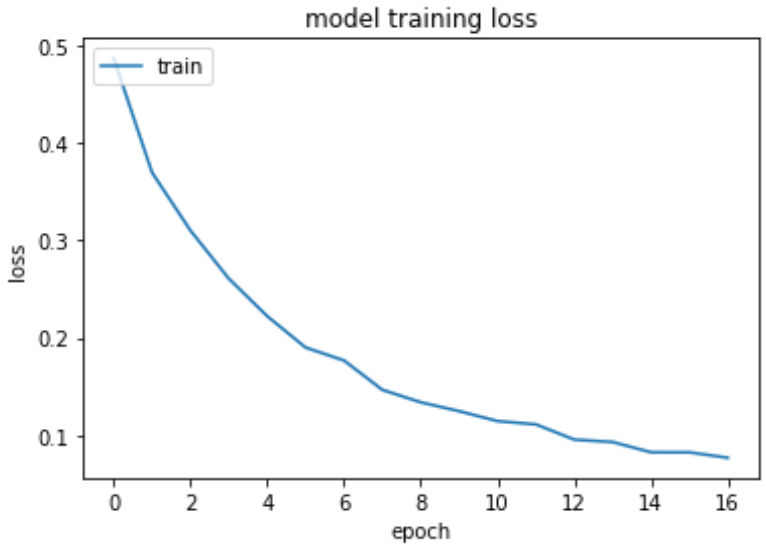

Figure 5-34 Training loss of 2D-CNN (knock vs acceleration).

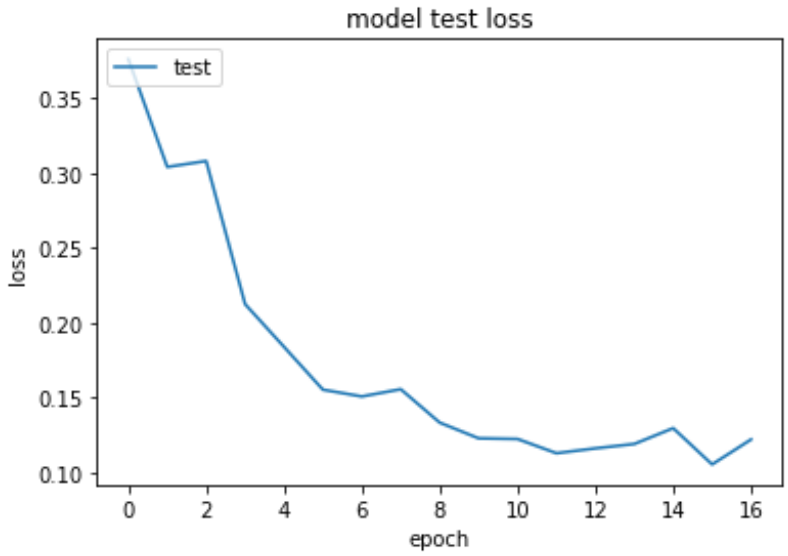

Figure 5-35 Testing loss of 2D-CNN (knock vs acceleration).

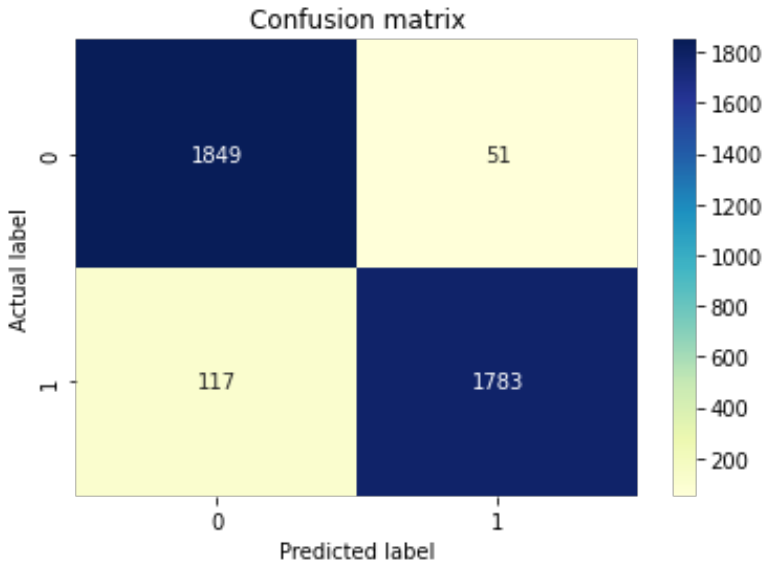

Figure 5-36 Confusion matrix for 2D-CNN (knock vs acceleration).

(4) 2D-CNN FOR MULTICLASS CLASSIFICATION USING MFCC

The average accuracy is 91.17% and it takes 4 mins and 50 sec to finish the training. This model has obtained suitable classification accuracy for all the engine audio samples.

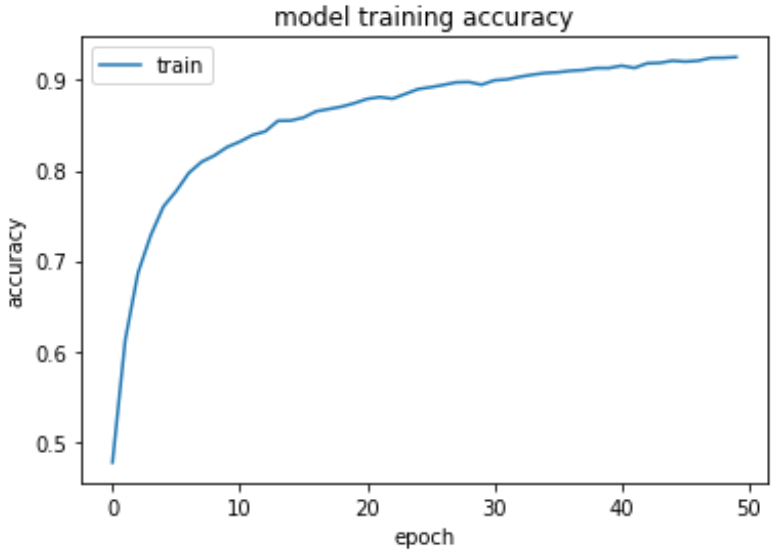

Figure 5-37 Training progress of 2D-CNN (multi-class classification).

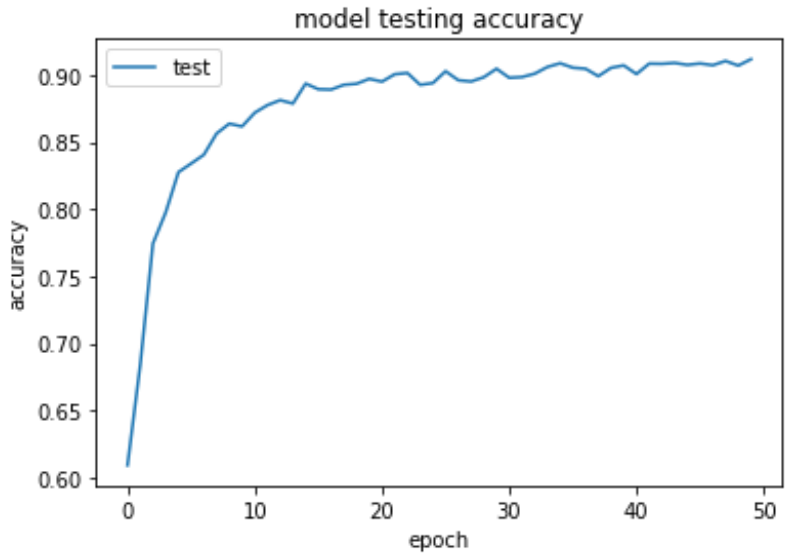

Figure 5-38 Testing accuracy of 2D-CNN (multi-class classification).

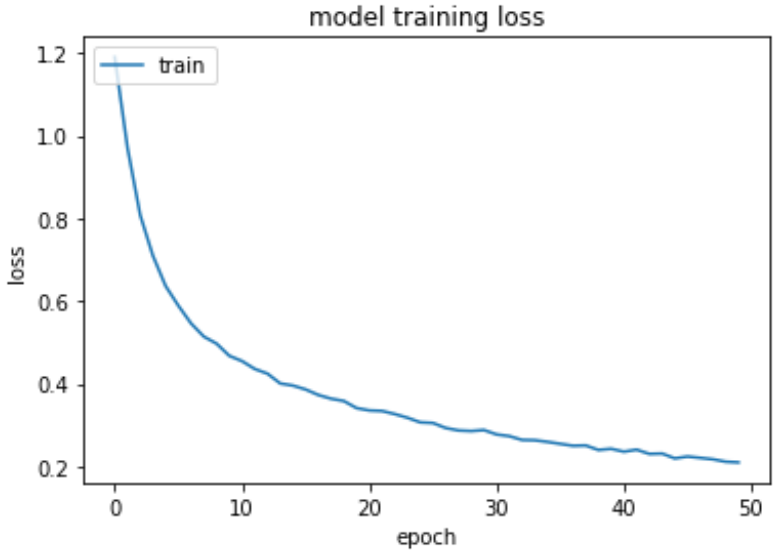

Figure 5-39 Training loss of 2D-CNN (multi-class classification).

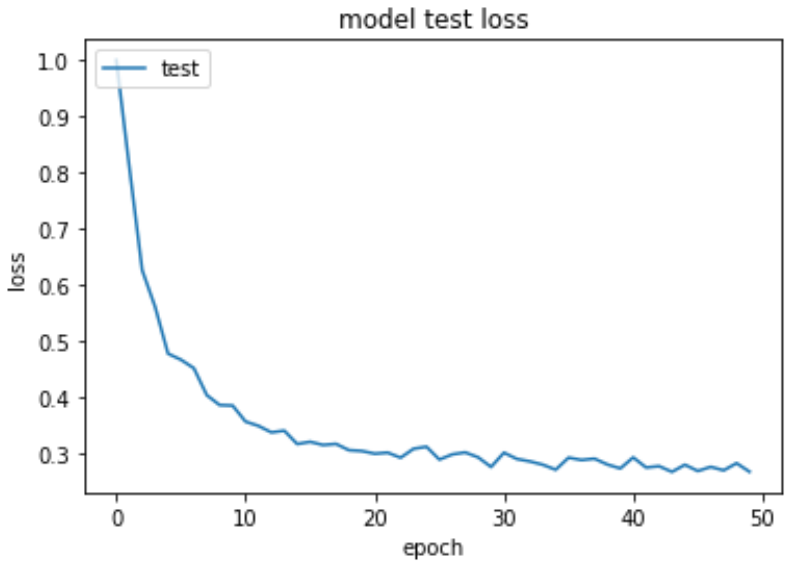

Figure 5-40 Testing loss of 2D-CNN (multi-class classification).

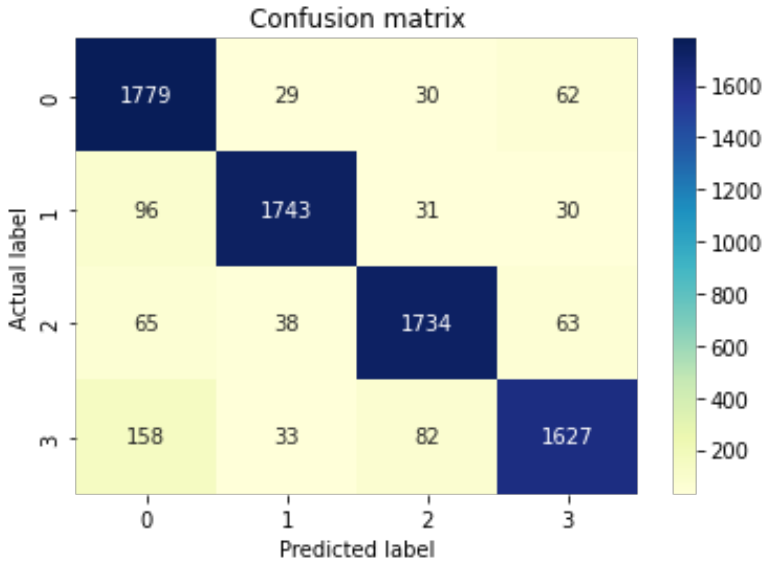

Figure 5-41 Confusion matrix of 2D-CNN (multi-class classification).

Table 5-2 provides an overview of the 2D-CNN model, with binary and multi-class models and their respective feature extraction methods validated in terms of accuracy produced by each model.

Table 5-2 Accuracies of 2D-CNN algorithms

Audio Samples	Class	Preprocessing	Accuracy
Knock Vs Idle	Binary	STFT	95.08
Knock Vs Start	Binary	STFT	92.29
Knock Vs Acceleration	Binary	STFT	94.79
All four classes	Multi	STFT	90.70
Knock Vs Idle	Binary	MFCC	96.82
Knock Vs Start	Binary	MFCC	94.47
Knock Vs Acceleration	Binary	MFCC	95.63
All four classes	Multi	MFCC	91.17

The results show that the MFCC method performs well for the 2D-CNN model because it extracts more data from the audio signal than STFT. The results of MFCC have surprised us, and we can say that MFCC can also be used for fault detection if we extract more data from the audio signal rather than just doing STFT. We can use both methods to detect the knocking sound from the engine because MFCC won by a small margin.

6. DESIGN OF 1D CONVOLUTIONAL NEURAL NETWORK

6.1 1D-CNN Structure

1D-CNN method is applied in critical wave handling functions like subject-limited ECG categorization, fundamental health examining, inconsistency identification in power electronics circuits, and motor-fault detection has lately been 1D Convolutional Neural Networks (CNNs). This is a predictable result given the many benefits of employing an efficient and shallow 1D CNN rather than a traditional (2D) deep equivalent and a correct 1D to 2D conversion is necessary when using a standard deep CNN for 1-D signal processing applications [52]. CNN is often defined as "2D CNN". The application of 1D CNN means that we can input one-dimensional audio signals directly acquired by audio clips into the network for operation.

Each convolution layer in 1D CNN is formed from numerous complication components. A back-propagation procedure enhances the limitations of every intricacy component, and the convolution activity's motivation is to extricate various elements of the information. Albeit the main convolution layer may just concentrate on a few low-layered parameters, extra layers of convolutional layers can extricate auxiliary perplexing elements from the low-layered parameters. The convolution layer plays out the convolution computation, with its convolution fragment convolved with the element submaps connected to the earlier layer.

$$x_j^i = f(\sum_{l \in M_j} x_l^{i-1} . \mathrm{k}_{lj}^i + b_j^i) \tag{6.1}$$

Where $x_l^{i-1}, \mathrm{k}_{lj}^i, b_j^i, f, and\ x_j^i$ represents the input, the kernel weights, the biases, the activation function, the feature maps of the *j* kernel in the i convolutional layer respectively in Equation (6.1).

The max amalgamating function produces new characteristic maps through subsampling the adjacent area and using the global algebraic qualities of nearby positions in the temporal field. This effectively reduces the parameter's dimension.

$$\hat{x}_j^i = f(\beta_j^i . down(x_j^i) + b_j^i) \tag{6.2}$$

Where $x_l^i, \beta_j^i, b_j^i, down(), and\ \hat{x}_j^i$ represents the input, the weight matrix, the biases, the down sampling function, the feature maps of the *j* kernel in i the pooling layer respectively in Equation (6.2).

The output of feature extraction is the input of fully connected layers. The parameters combined in dual dimensions can convert to a 1-D parameterized array using the fully connected layer. As a result, each regional characteristic of the source information unified with numerous outlier characters, also, the changed highlights are then used to compute the result in support of every classification. The SoftMax function assesses the likelihood of all categorization outcomes, can generate the output vector of predictions. The cross-entropy capability is utilized to figure out the misfortune upsides of the result and target vectors [53].

$$C = -\frac{1}{n}\sum_{k=0}^{n-1}[y_k \ln d_k + (1 - y_k)\ ln\ (1 - d_k)] \tag{6.3}$$

where C is the total loss value, y_k represents the k^{th} output vector of a fully connected neural network, $d_k$indicates the target vector of the k^{th} simulation data in Equation (6.3).

2D-CNN was the most used architecture for image classification, but several researchers have suggested 1D CNNs recently, which take the raw audio stream as input. For audio tagging, music genre classification, environmental sound categorization, and 1D residual CNN architecture, various 1D CNN architectures have been developed [54].

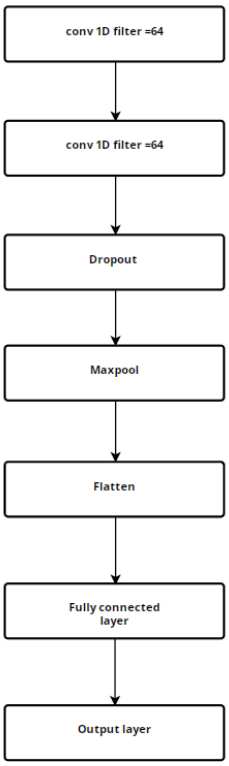

Figure 6-1 1D-CNN architecture for audio classification.

We employed a standard 1D-CNN architecture with a minimal number of layers, as shown in Figure 6-1. Researchers have developed numerous sophisticated designs for various purposes, which also increases the complexity and time of the model. Table 6-1 contains the detailed parameters we used in the 1D-CNN structure.

Table 6-1 Information of 1D-CNN Layers

Layer name	CNN Models (Kernel Filter, Activation Function, Strides, Padding)
C1	Conv 1D (64*3*1, ReLU, 1, None)
C2	Conv 1D (64*3*1, ReLU, 1, None)
MXP3	Maxpooling 1D (2,2,1)
FC1	Fullyconnect (100, ReLU)
Output	SoftMax, Classification

From Table 6-1 and Figure 6.1, we can understand a detailed explanation of all the elements involved in 1D-CNN to classify audio samples of the engine.

6.2 Experimental Results

We have used the same setup for ANN, 2D-CNN, and 1D-CNN architectures in this research. Initially, we will be testing on binary and multi-class classification.

6.2.1 Results for 1D-CNN using STFT feature extraction

After validating the 2D-CNN structure, we discovered that it has a higher computation cost than the 1D-CNN structure. Classification methods used in all the preceding models can also be applied to 1D-CNN with less computation cost. Epochs were 17 for binary class and 50 for multi-class model.

(1) 1D-CNN FOR ENGINE KNOCK VS ENGINE IDLE USING STFT

The average accuracy is 97.58%; it takes 3 mins and 24 sec to finish the training. Figures 6-2, 6-3, 6-4, 6-5, and 6-6 show that the method has best classification accuracy.

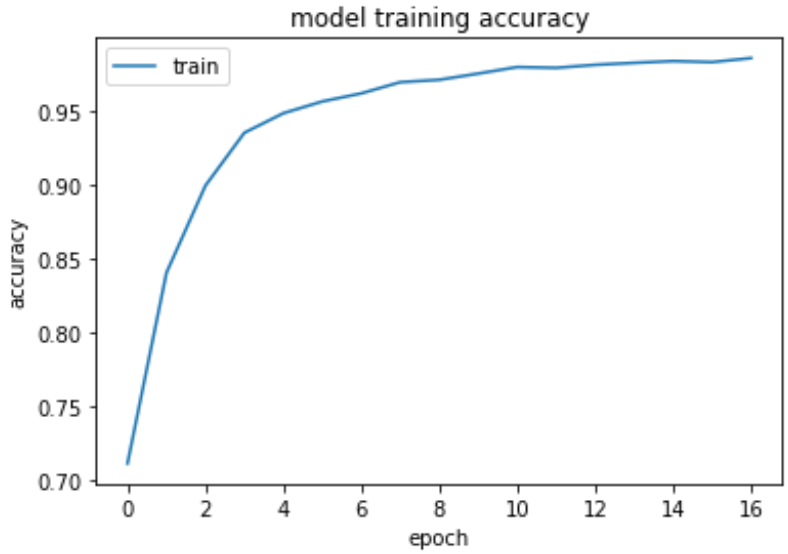

Figure 6-2 Training progress of 1D-CNN (knock vs idle).

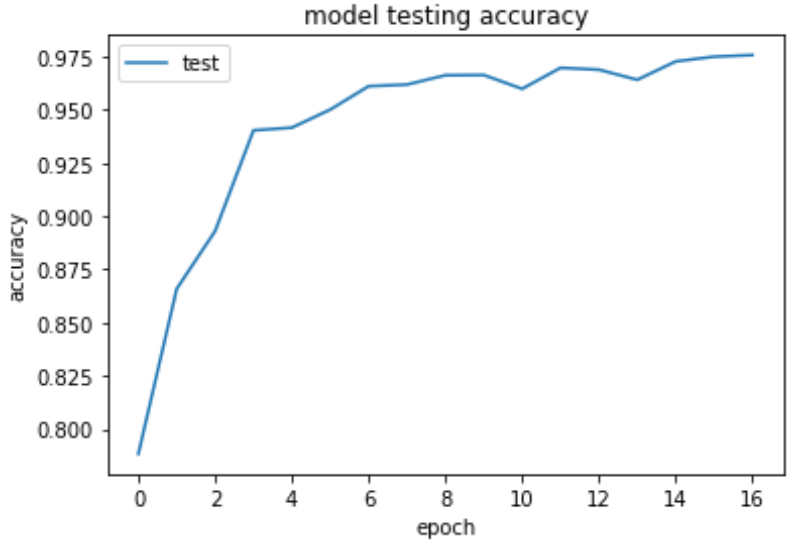

Figure 6-3 Testing accuracy of 1D-CNN (knock vs idle).

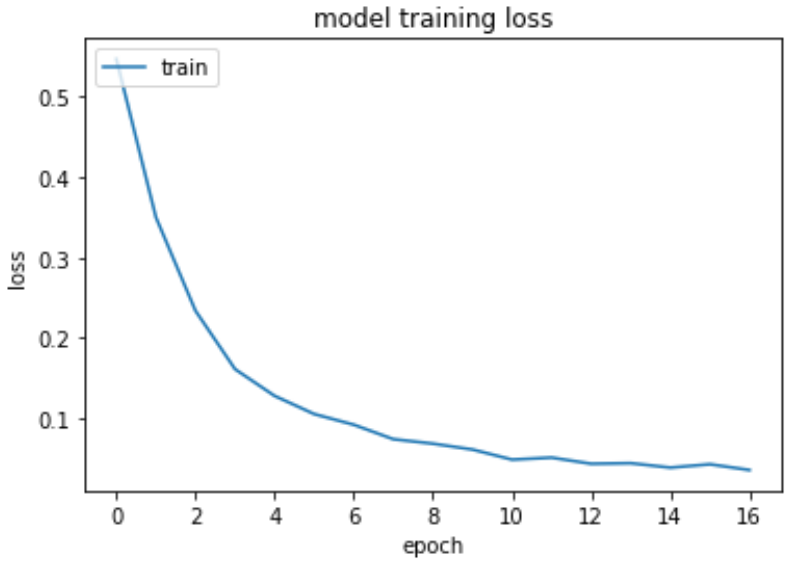

Figure 6-4 Training loss of 1D-CNN (knock vs idle).

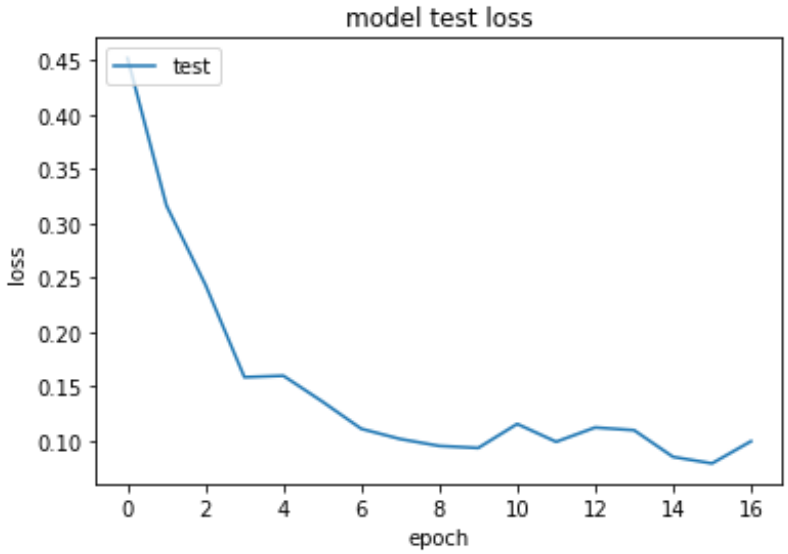

Figure 6-5 Testing loss of 1D-CNN (knock vs idle).

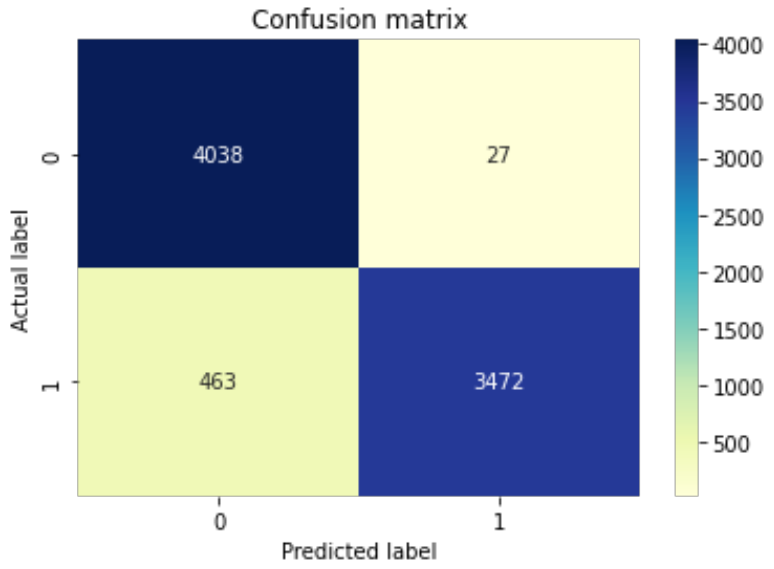

Figure 6-6 Confusion matrix for 1D-CNN (knock vs idle).

(2) 1D-CNN FOR ENGINE KNOCK VS ENGINE START USING STFT

The average accuracy is 94.79% and it takes 3 mins and 24 sec to finish the training. Lower classification accuracy has been seen in this method when compared with above idle sample sound model.

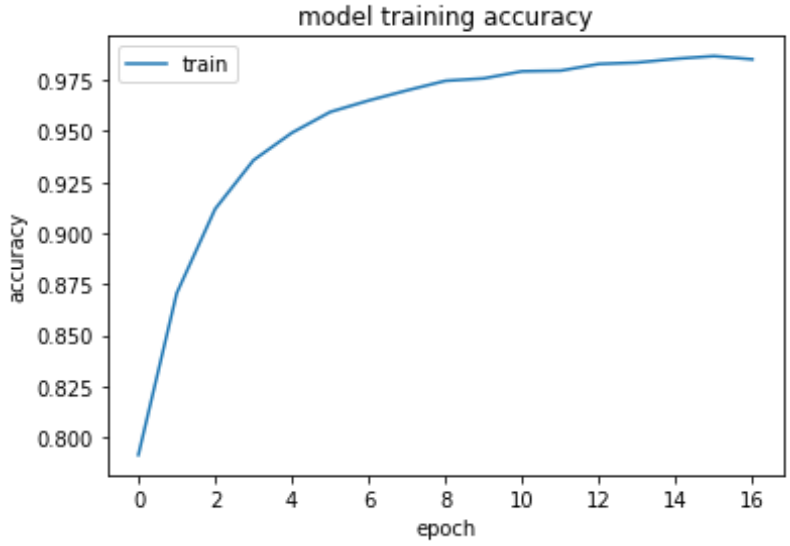

Figure 6-7 Training progress of 1D-CNN (knock vs start).

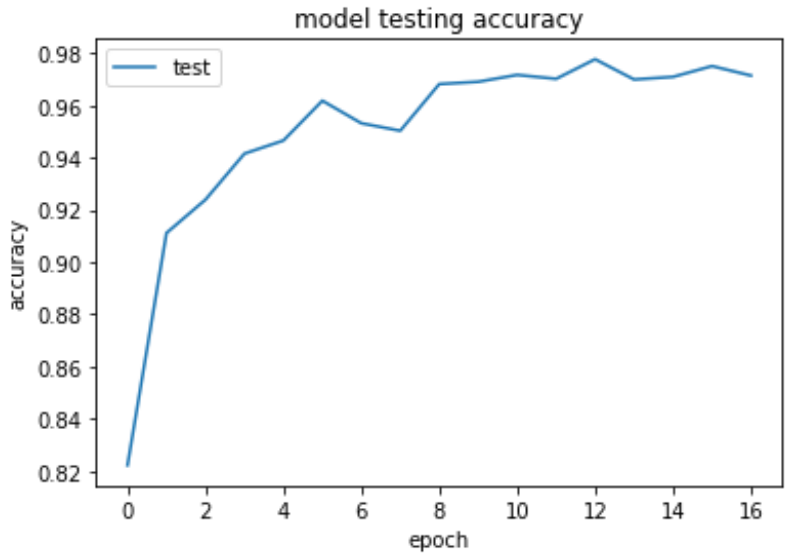

Figure 6-8 Testing accuracy of 1D-CNN (knock vs start).

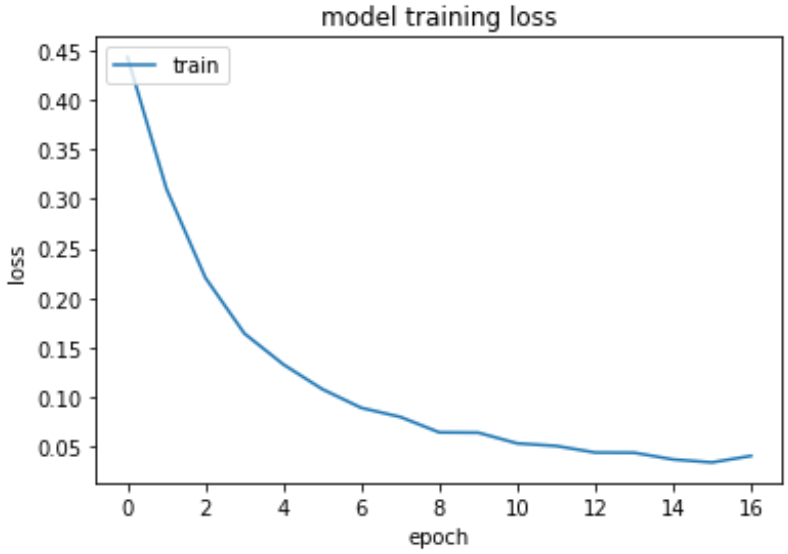

Figure 6-9 Training loss of 1D-CNN (knock vs start).

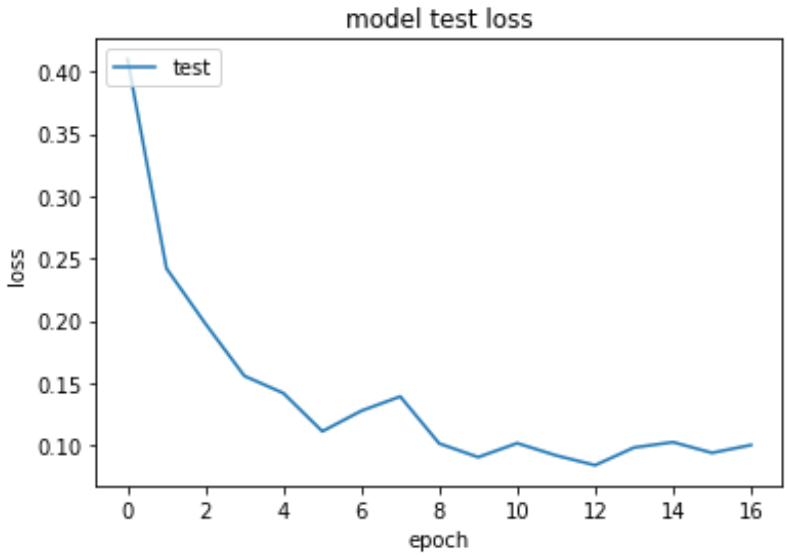

Figure 6-10 Testing loss of 1D-CNN (knock vs start).

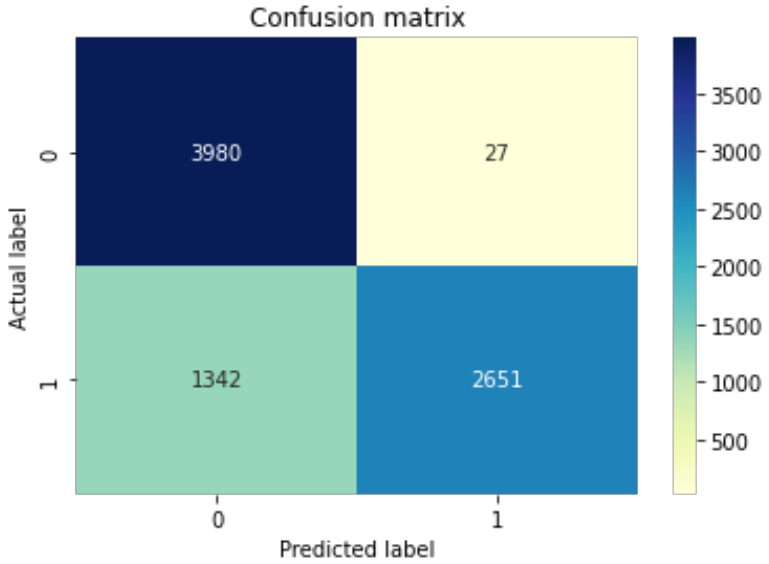

Figure 6-11 Confusion matrix for 1D-CNN (knock vs idle).

(3) 1D-CNN FOR ENGINE KNOCK VS ENGINE ACCELERATION USING STFT

The average accuracy is 97.14%. It takes 3 mins and 23 sec to finish the training. Better classification accuracy has been observed in this method when compared with engine start but idle audio samples model still stand as best.

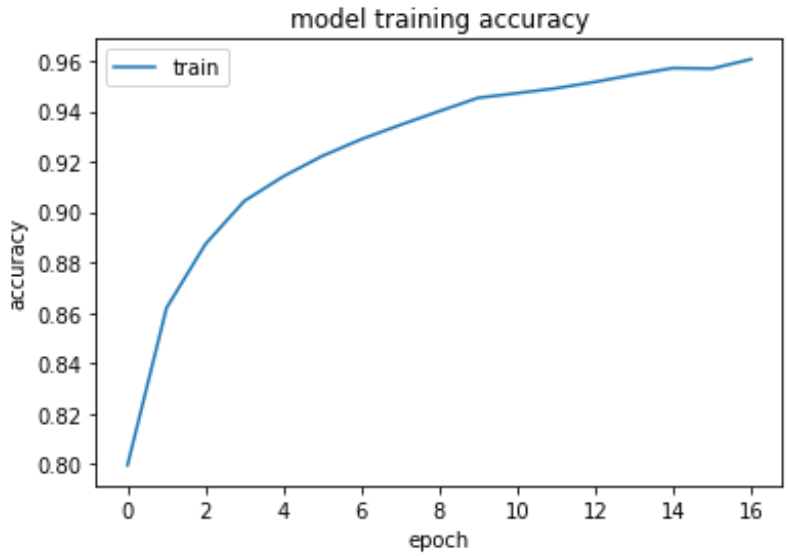

Figure 6-12 Training progress of 1D-CNN (knock vs acceleration).

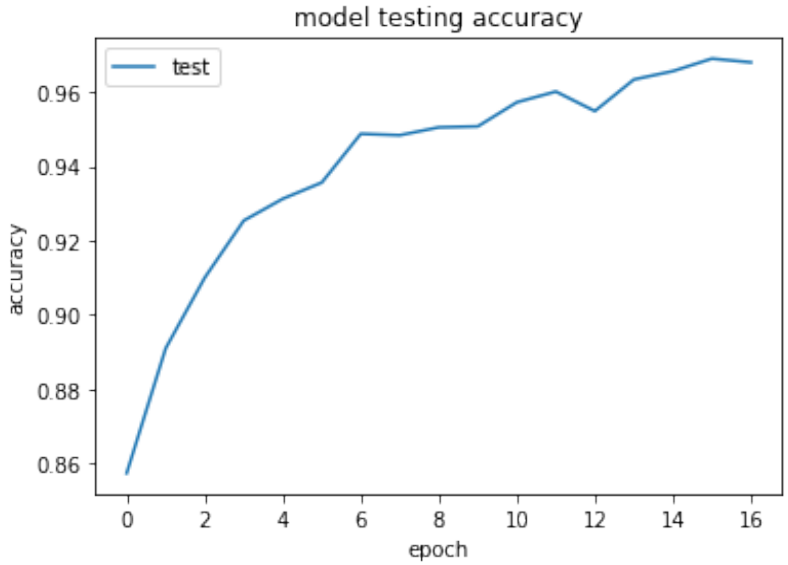

Figure 6-13 Testing accuracy of 1D-CNN (knock vs acceleration).

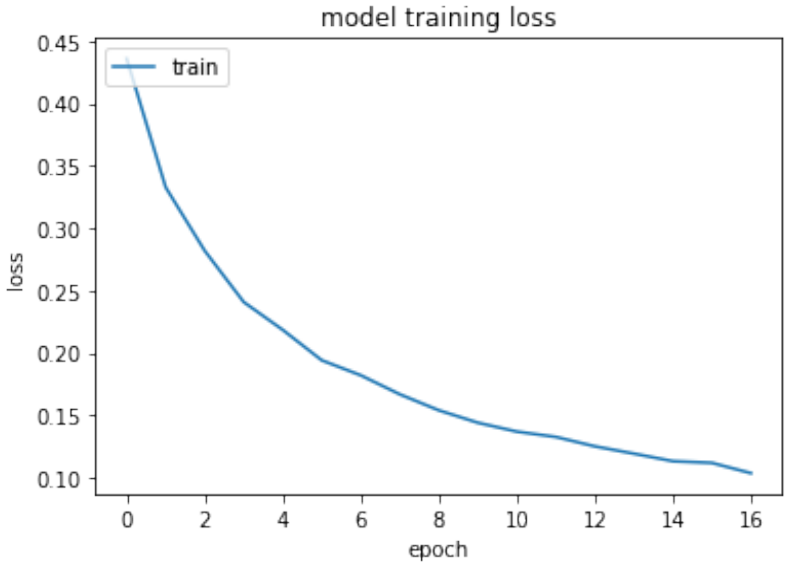

Figure 6-14 Training loss of 1D-CNN (knock vs acceleration).

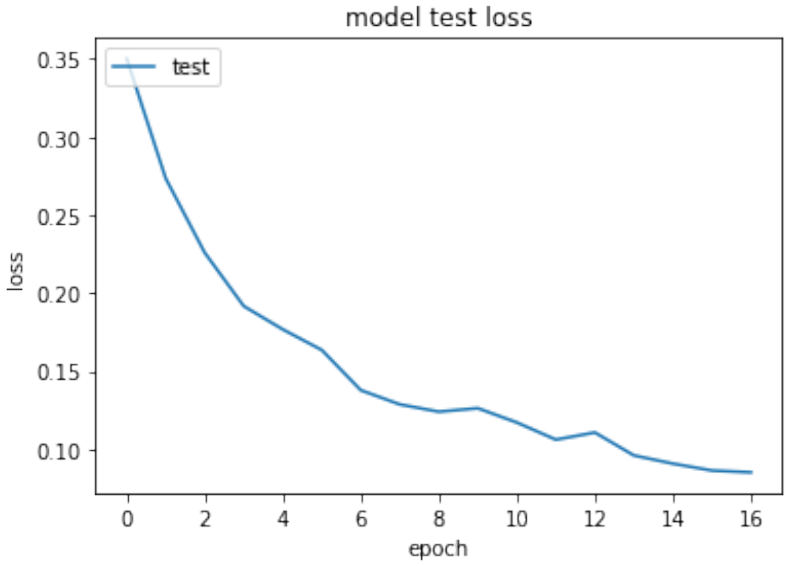

Figure 6-15 Testing loss of 1D-CNN (knock vs acceleration).

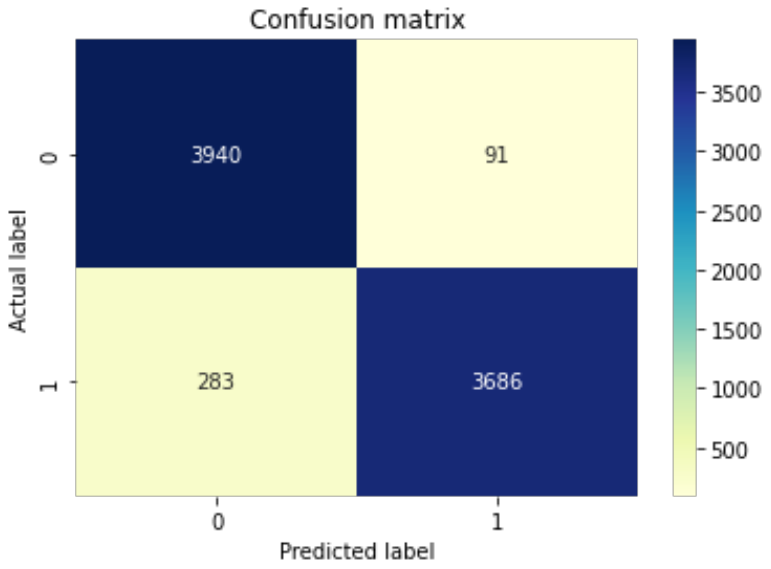

Figure 6-16 Confusion matrix for 1D-CNN (knock vs acceleration).

(4) 1D-CNN FOR MULTICLASS CLASSIFICATION USING STFT

The average accuracy is 90.00%; it takes 4 mins and 45 sec to finish the training. High quality classification accuracy for all the engine audio samples model is seen in this method which is demonstrated in below figures.

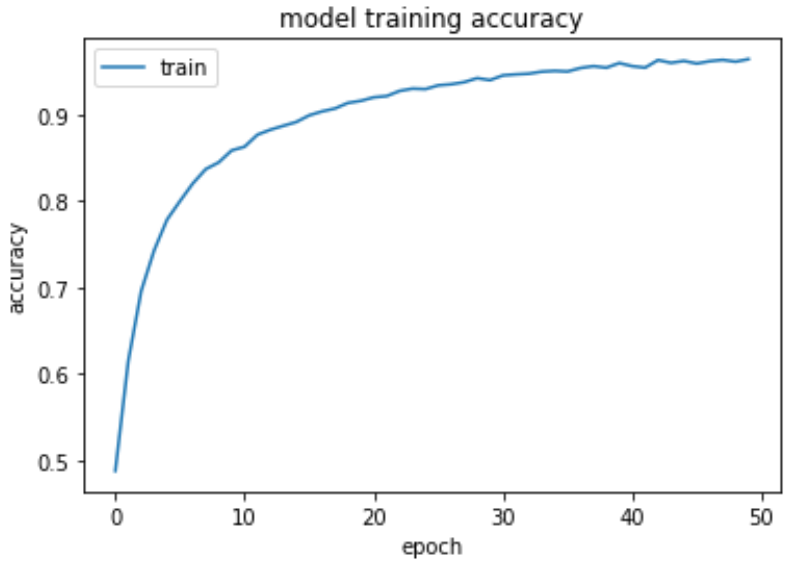

Figure 6-17 Training progress of 1D-CNN (multi-class classification).

Figure 6-18 Testing accuracy of 1D-CNN (multi-class classification).

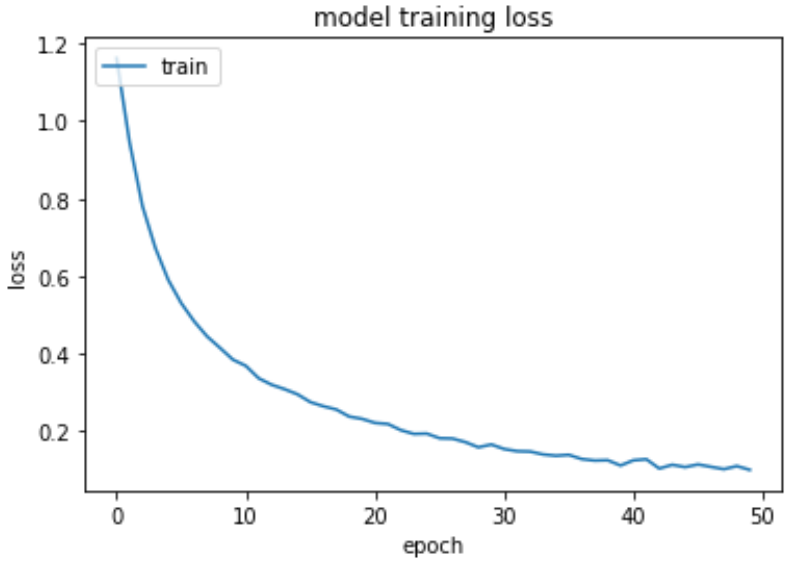

Figure 6-19 Training loss of 1D-CNN (multi-class classification).

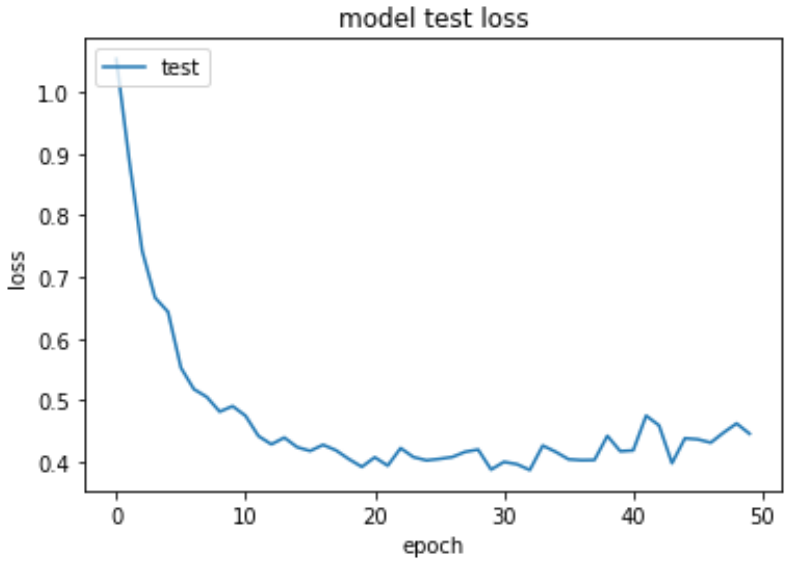

Figure 6-20 Testing loss of 1D-CNN (multi-class classification).

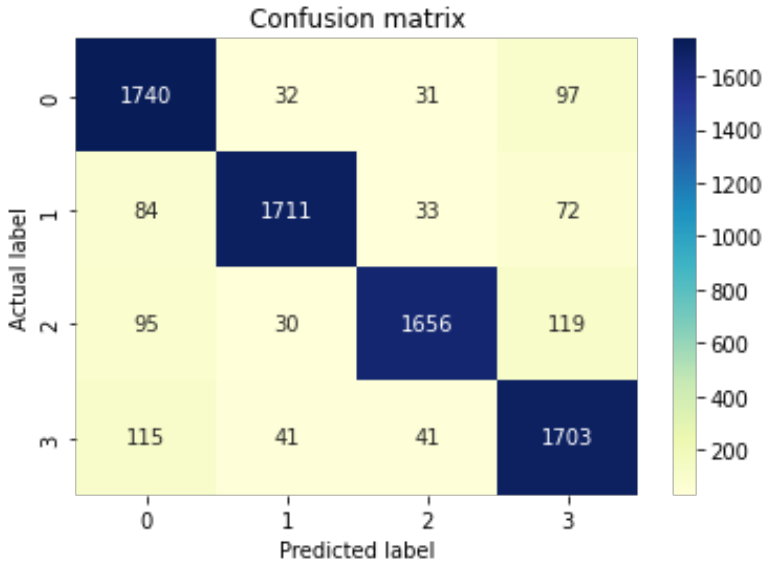

Figure 6-21 Confusion matrix of 1D-CNN (multi-class classification).

6.2.2 Results for 1D-CNN using MFCC feature extraction

MFCC is a popular technique for extracting features from an audio signal. Because these audio signals are mostly human voices, validating MFCC for engine audio samples in the above models yielded significant results, and 1D-CNN confirmed that MFCC is a good feature extraction technique for engine fault diagnosis. Epochs setup is same as STFT 1D-CNN.

(1) 1D-CNN FOR ENGINE KNOCK VS ENGINE IDLE USING MFCC

The average accuracy is 92.72% and it takes 3 mins and 27 sec to finish the training. Figures 6-22, 6-23, 6-24, 6-25, and 6-26 show that the method has produced lower classification accuracy as STFT.

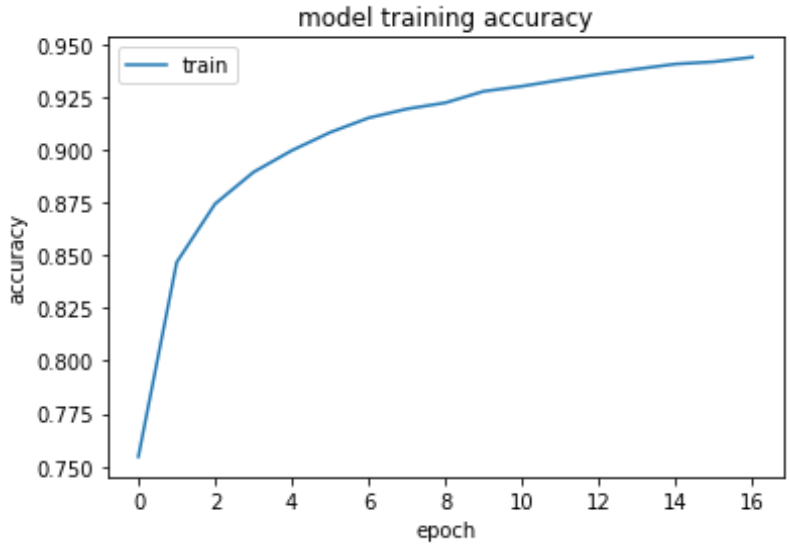

Figure 6-22 Training progress of 1D-CNN (knock vs idle).

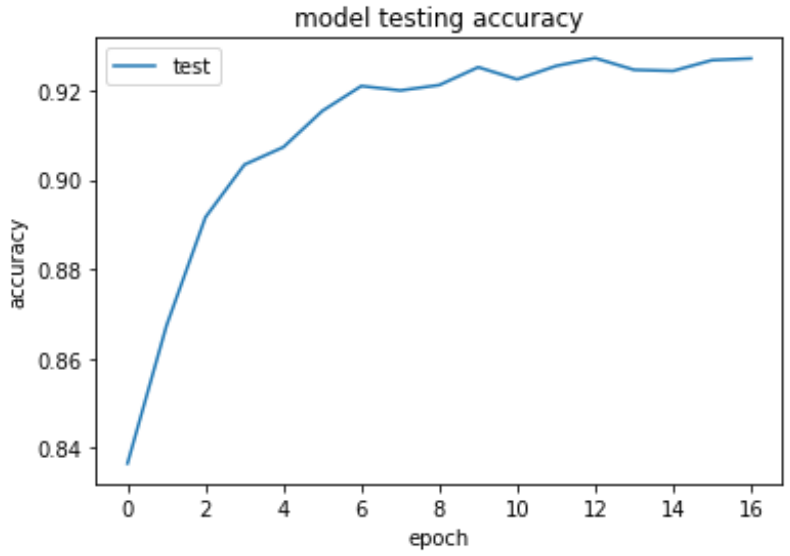

Figure 6-23 Testing accuracy of 1D-CNN (knock vs idle).

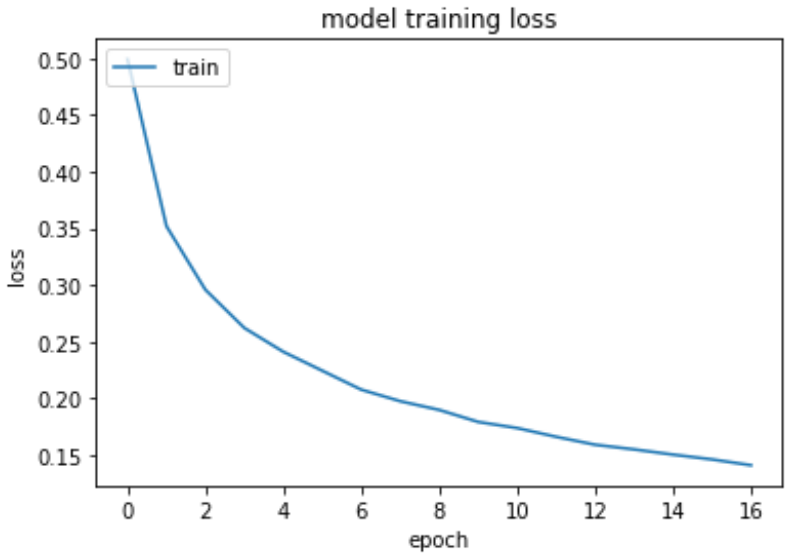

Figure 6-24 Training loss of 1D-CNN (knock vs idle).

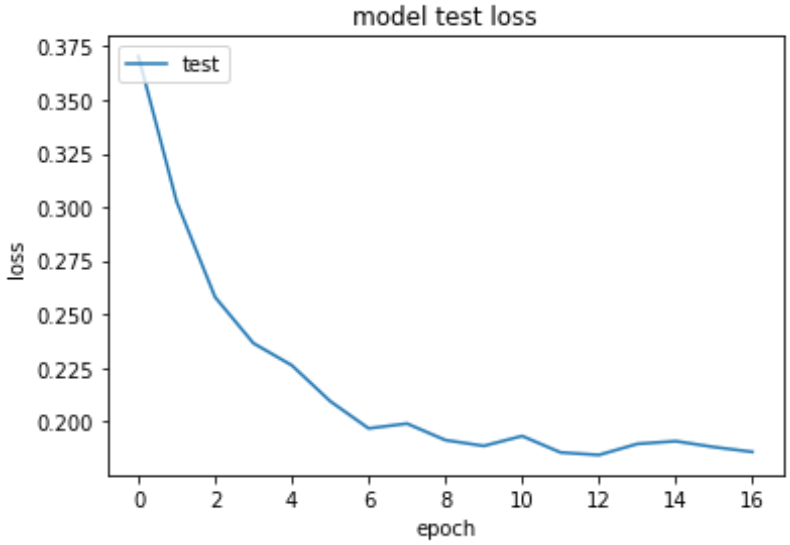

Figure 6-25 Testing loss of 1D-CNN (knock vs idle).

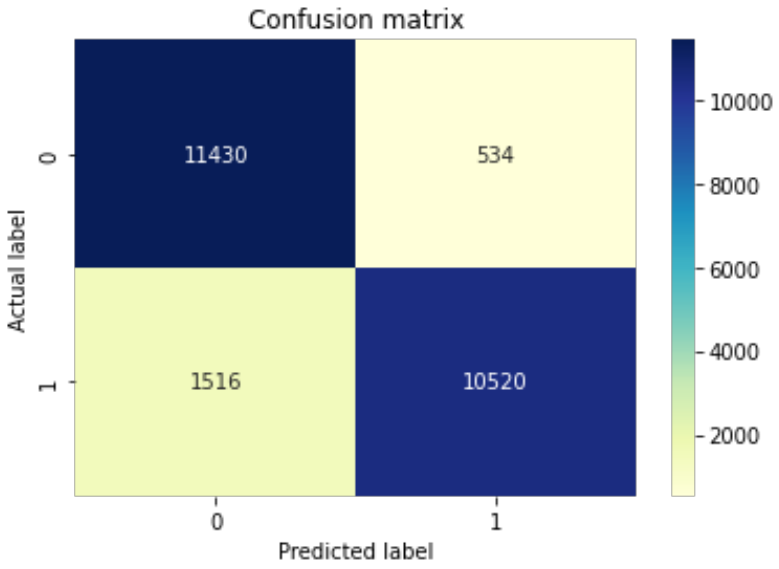

Figure 6-26 Confusion matrix for 1D-CNN (knock vs idle).

(2) 1D-CNN FOR ENGINE KNOCK VS ENGINE START USING MFCC

The average accuracy is 90.04%; it takes 4 mins and 22 sec to finish the training. From Figures 6-27 till 6-31 we can see that the method has maintained good classification accuracy when compared with above idle model it is lower by small percentage.

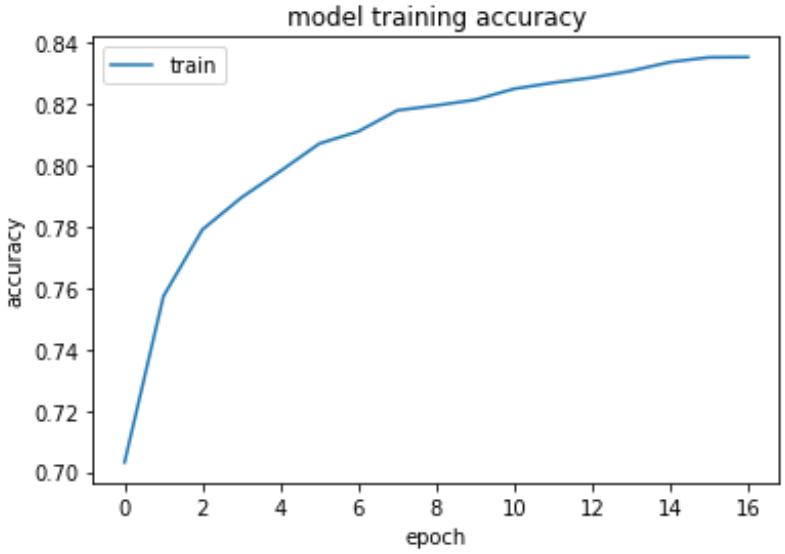

Figure 6-27 Training progress of 1D-CNN (knock vs start).

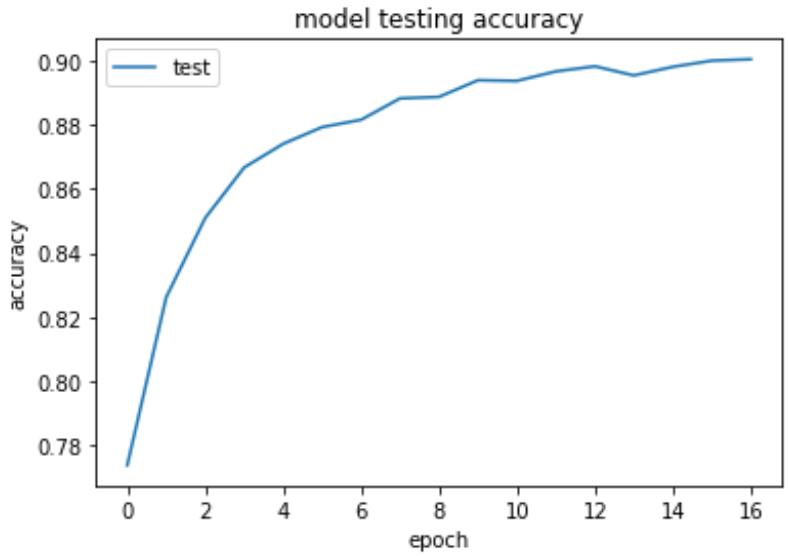

Figure 6-28 Testing accuracy of 1D-CNN (knock vs start).

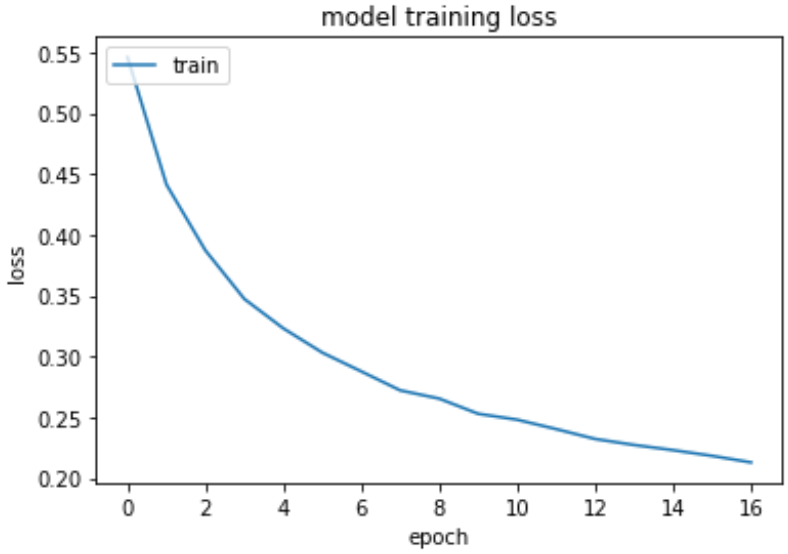

Figure 6-29 Training loss of 1D-CNN (knock vs start).

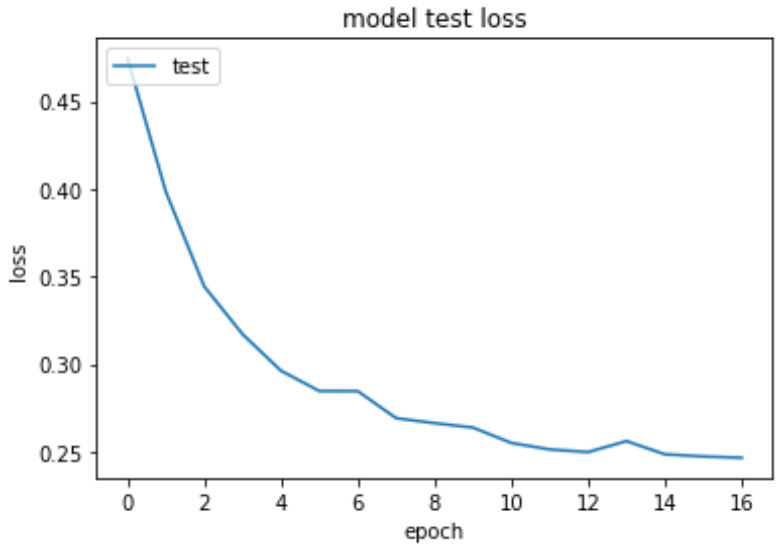

Figure 6-30 Testing loss of 1D-CNN (knock vs start).

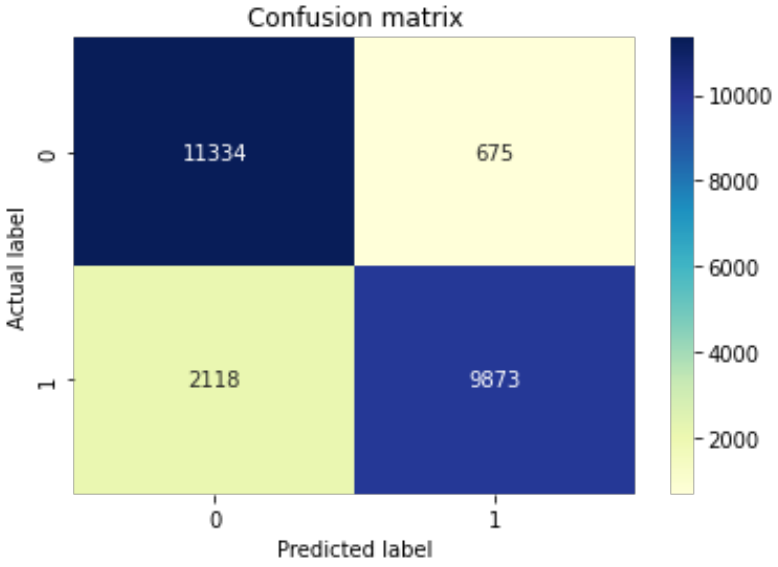

Figure 6-31 Confusion matrix for 1D-CNN (knock vs start).

(3) 1D-CNN FOR ENGINE KNOCK VS ENGINE ACCELERATION USING MFCC

The average accuracy is 93.21%. It takes 4 mins and 45 sec to finish the training. From Figures below we can see that the method has better classification accuracy when compared with engine start and idle audio samples model.

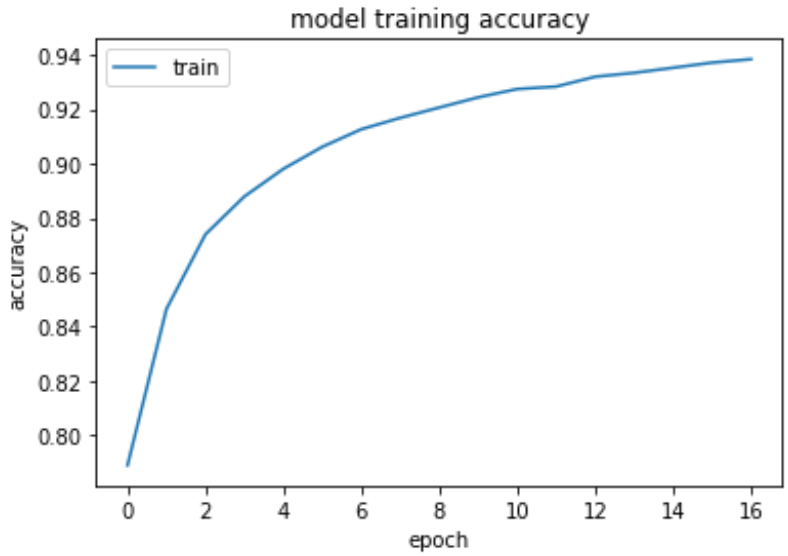

Figure 6-32 Training progress of 1D-CNN (knock vs acceleration).

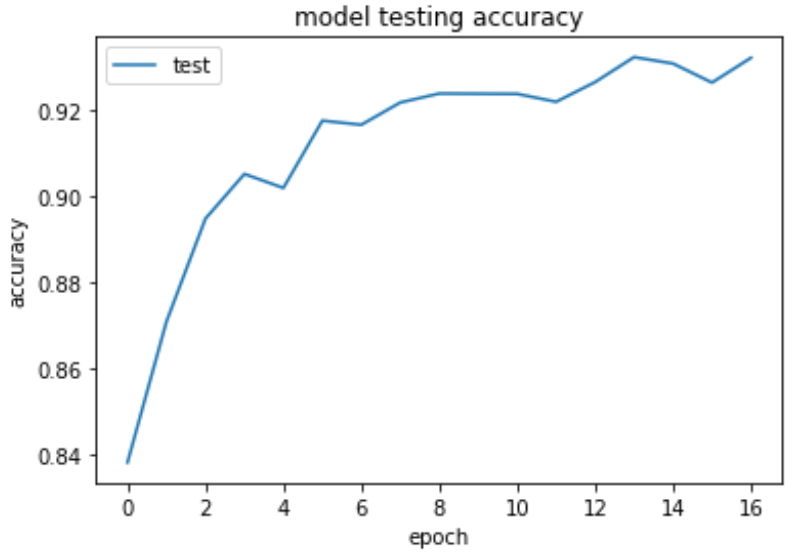

Figure 6-33 Testing accuracy of 1D-CNN (knock vs acceleration).

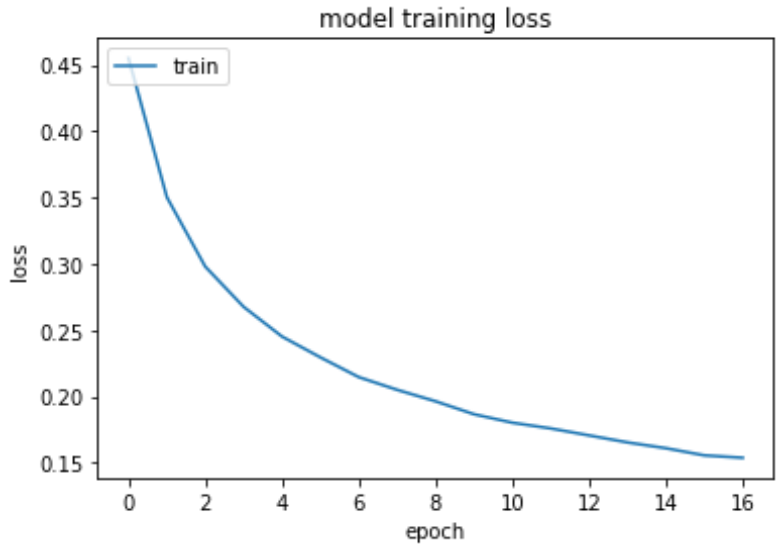

Figure 6-34 Training loss of 1D-CNN (knock vs acceleration).

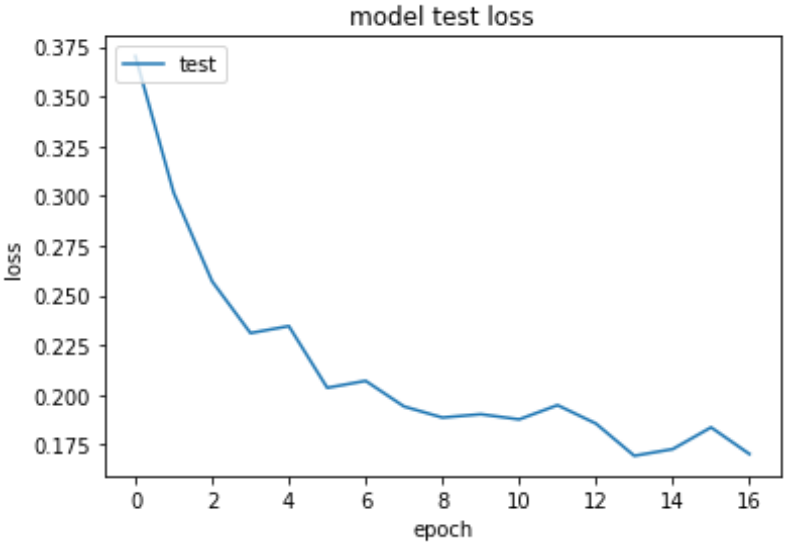

Figure 6-35 Testing loss of 1D-CNN (knock vs acceleration).

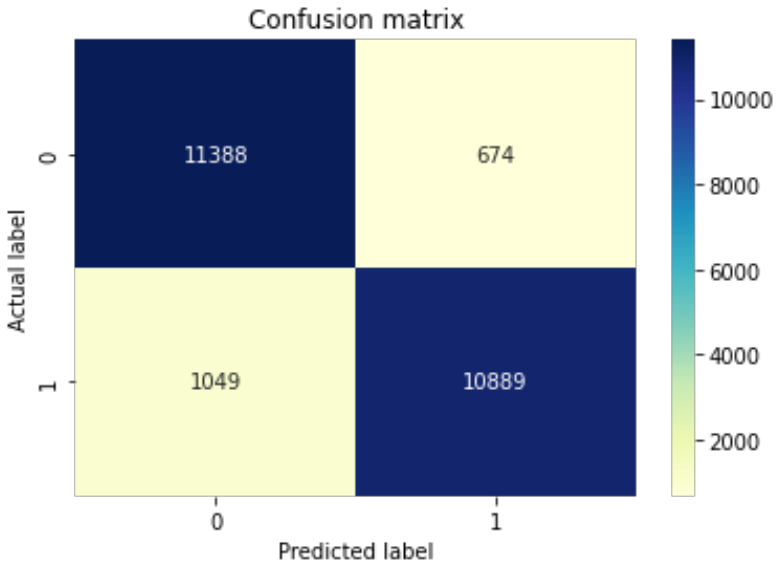

Figure 6-36 Confusion matrix for 1D-CNN (knock vs acceleration).

(4) 1D-CNN FOR MULTICLASS CLASSIFICATION USING MFCC

The average accuracy is 91.24%; it takes 4 mins and 57 sec to finish the training. We can see in below Figures that the method has done very well on classification accuracy for all the engine audio samples model.

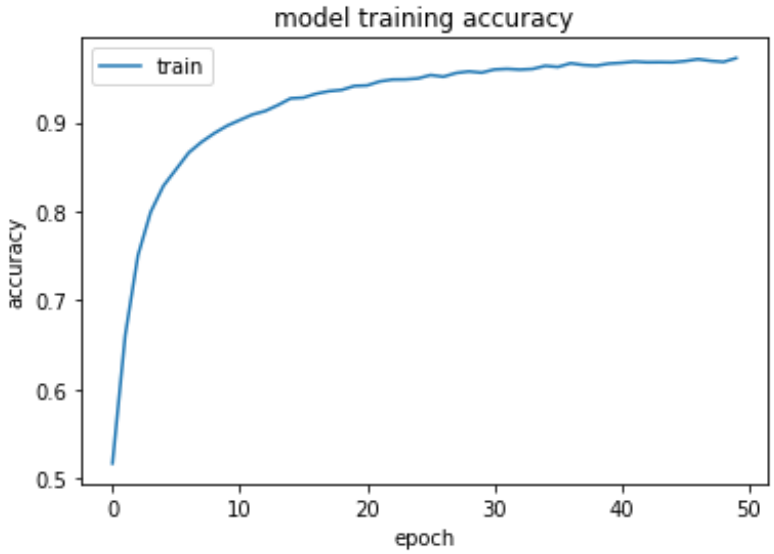

Figure 6-37 Training progress of 1D-CNN (multi-class classification).

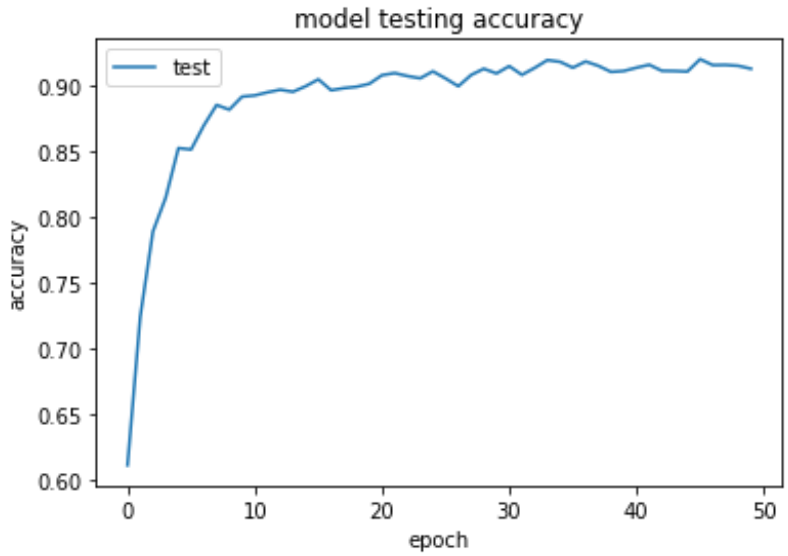

Figure 6-38 Testing accuracy of 1D-CNN (multi-class classification).

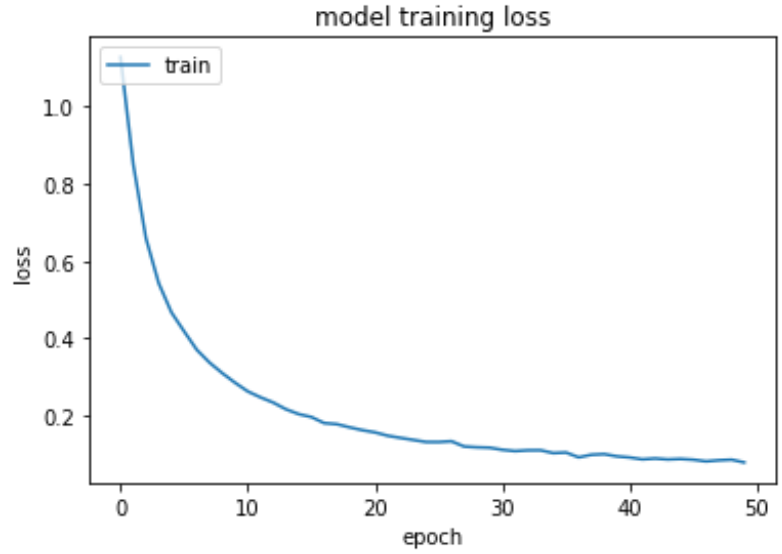

Figure 6-39 Training loss of 1D-CNN (multi-class classification).

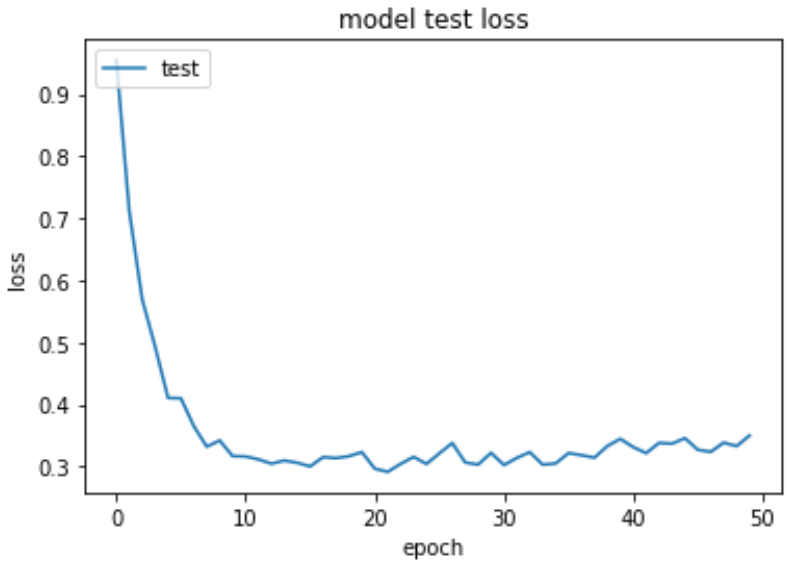

Figure 6-40 Testing loss of 1D-CNN (multi-class classification).

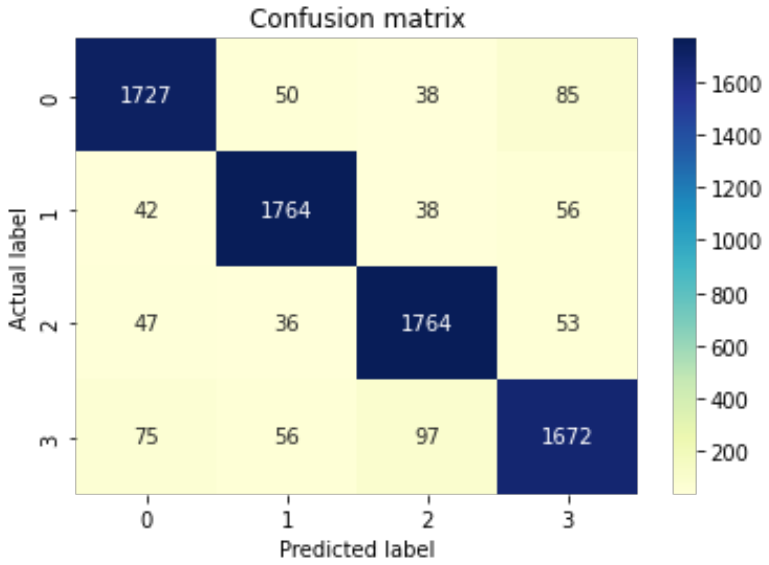

Figure 6-41 Confusion matrix of 1D-CNN (multi-class classification).

Table 6-2 is the summarized version of all the variations of classes and audio samples used in 1D-CNN has been mentioned with accuracy as the metric to evaluate the model for binary and multi-class.

Table 6-2 Accuracies of 1D-CNN algorithms

Audio Samples	Class	Preprocessing	Accuracy
Knock Vs Idle	Binary	FFT	97.58
Knock Vs Start	Binary	FFT	94.79
Knock Vs Acceleration	Binary	FFT	97.14
All four classes	Multi	FFT	90.07
Knock Vs Idle	Binary	MFCC	92.72
Knock Vs Start	Binary	MFCC	90.04
Knock Vs Acceleration	Binary	MFCC	93.21
All four classes	Multi	MFCC	91.24

Regardless of their applications, both preprocessing methods from Table 6-2 MFCC and FFT can be used to detect knocking.

We can see that FFT produces better accuracy than MFCC, but we cannot ignore the accuracy produced by both preprocessing methods on the simple architecture of 1D-CNN. FFT with idle samples is the best binary model, and MFCC has good accuracy in multi-class.

7. CONCLUSION AND FUTURE WORK

In our research, we have employed three different designs to identify engine faults. We have discovered knocking samples in multi-class settings in addition to finding them in various engine circumstances. While the complexity of deep learning models was the focus of all prior research, we were more concerned with the data, which is the industry's biggest challenge. The accuracy of finding engine knocking was increased by using a dependent vehicle feature extraction technique.

Table 7-1 Comparison of all models used to detect knocking

Models	Class	Preprocessing	Accuracy
ANN	Multi	FFT	92.18
ANN	Multi	MFCC	77.52
2D CNN	Multi	STFT	90.70
2D CNN	Multi	MFCC	91.17
1D CNN	Multi	FFT	90.07
1D CNN	Multi	MFCC	91.24

Compared to ANN, 2D-CNN and 1D-CNN all outperformed it with the feature extraction method, with 1D-CNN coming out on top. If time complexity is a concern, we can switch to ANN which can be seen in Table 7-2 which has average time utilized by each model and look for knocking in the samples.

Table 7-2 Time complexity of all models used to detect knocking

Models	Training time	Testing time
ANN	279.25 sec	0.1 sec
2D CNN	457.25 sec	0.1 sec
1D CNN	336.37 sec	0.1 sec

The exhibition of well-known convolutional neural networks (CNN), and Artificial neural networks (ANN) for engine error identification built on audio signal processing has been validated. The audio wave handling methods incorporate the dependent vehicle feature extraction technique with short-time Fourier transform (STFT), Mel-frequency cepstral coefficient (MFCC), and fast Fourier transform (FFT). The Audio set of the engine ontology dataset is embraced for our engine fault detection. FFT-CNN achieves the most elevated precision in the CNN-based networks, and the FFT-ANN networks have the most elevated exactness in the ANN-based networks. The CNN-based networks perform better compared to the ANN networks overall. MFCC model offers the finest performance in a multi-class environment. Our simulations validate the effectiveness of each network input formulation. Moreover, the one-dimensional convolutional neural network is studied and constructed.

We can develop an independent vehicle feature extraction technique that can be applied to any vehicle in the future and the current research will stand as a base for such a strategy. Our engine fault detection method detects faults brilliantly from the vehicle samples present in the Audio set; however, each vehicle has unique characteristics and throughputs, it is difficult to apply the same technique to all vehicles. Engine faults detected in different audio samples from the Audio set can be further classified into various faulty engine parts that are causing such a knocking sound.

Techniques like data augmentation [55] and an extensive collection of engine audio samples will lead us to a more generalized version of engine-knocking detection. Our research has some limitations such as the fact that it can only be applied to cars that have been used in training our deep learning models, and that it cannot be used to detect knocking on running road cars because the source will be a microphone that will collect other sounds besides engine sound; the challenge will be separating the engine sound from the noise. Blind source separation is a technique for distinguishing engine sounds from other sounds. We can have the perfect tool to detect knocking across any car by using a denser audio set that contains audio samples of cars that are frequently seen on the road and a blind source separation technique.

REFERENCES

[1] F. Lv, C. Wen, Z. Bao, and M. Liu, “Fault diagnosis based on deep learning,” IEEE American Control Conference (ACC), pp. 6851-6856, 2016.

[2] Carlos Alba (chief digital officer at ArcelorMittal), Peter D’haese (chief digital officer (CDO) at ArcelorMittal Flat Europe), Personal interview, 2020.

[3] M. Buscema, S. Terzi, and W. Tastle, “A new meta-classifier,” IEEE Annual Meeting of the North American Fuzzy Information Processing Society, pp. 1-7, 2010.

[4] A. Saxena, K. Goebel, D. Simon, and N. Eklund, “Damage propagation modeling for aircraft engine run-to-failure simulation,” IEEE International Conference on Prognostics and Health Management, 2008, pp. 1-9, 2008.

[5] J. C. P. Jones, J. M. Spelina, and J. Frey, “Likelihood-Based Control of Engine Knock,” in IEEE Transactions on Control Systems Technology, vol. 21, no. 6, pp. 2169-2180, Nov. 2013.

[6] Infosys, Predict Vehicle Failure and Auto Scheduling of Vehicle Preventive Maintenance in Garage/Depot, 2021.

[7] A. Salgame, N. Raju D, “How to get started with Predictive Maintenance using Machine Learning,” Inuceo, 2020.

[8] D. O'Shea, “Questar uses AI, predictive analytics to keep fleets on the road,” Fierce Electronics, Questar, 2022.

[9] S. Dube, S. Gupta, A. Iyengar, G. Indurkhya, “IoT, edge computing, and AI combine to disrupt the automotive industry,” IBM, 2019.

[10] T. Holzmueller, “Predictive maintenance: When a machine knows in advance that repairs are needed,” BMW Group, 2021.

[11] B. Nagy, and C. Seymor, “Drivers to Benefit From Smarter Insurance, Maintenance And Recovery Following Connected Vehicle Data Collaboration,” Ford and CARUSO, 2020.

[12] Hyundai and Kia Motors Namyang R&D Center, “HMG, the World’s First AI-Based Diagnostics for Fault Detection,” Hyundai Motor Group, 2019.

[13] L. Zhang, P. Shen, and F. Bi, “Engine knock detection and intensity evaluation based on sparse maximum correlation kurtosis devonvolution,” IEEE 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT), pp. 598-605, 2020.

[14] M. Rakesh, “Knocking and Combustion Noise Analysis,” Reciprocating Engine Combustion Diagnostics, pp.461-542, 2019.

[15] Sung Tae Park, and Jinguo Yang, “Engine knock detection based on wavelet transform,” Proceedings. The 8th Russian-Korean International Symposium on Science and Technology, KORUS 2004., pp. 80-83 vol. 3, 2004.

[16] F. Molinaro, F. Castanie, and A. Denjean, “Knocking recognition in engine vibration signal using the wavelet transform,” Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis, pp. 353-356, 1992.

[17] H. Lee, and Y. Park, “Adaptive Algorithm of Active Sound Design for the Engine Noise,” 12th Asian Control Conference (ASCC), 2019, pp. 343-347, 2019.

[18] O. Boubai, “Knock detection in automobile engines,” in IEEE Instrumentation & Measurement Magazine, vol. 3, no. 3, pp. 24-28, 2000.

[19] A. Razzak, A. Yasin, A. Abas, and H. Gitano, “Modern Methods in Engine Knock Signal Detection” ScienceDirect, Procedia Technology, vol. 11, pp. 40-50, 2013.

[20] L. Petrucci, F. Ricci, F. Mariani, and V. Cruccolini, “Engine Knock Evaluation Using a Machine Learning Approach,” Conference on Sustainable Mobility, 2020.

[21] Z. Zhou, S. Xiong, Y. Chen, C. Zhang, and Y. Cao, “Deep learning approach for super-knock event prediction of petrol engine with sample imbalance,” Fuel, vol. 311, 2020.

[22] J.Galloway, “Lewis Hamilton Malaysia GP engine blow-up traced to faulty bearing,” Suzuka, F1 News Article, Sky Sports, 2016.

[23] J. F. Gemmeke et al., “Audio Set: An ontology and human-labeled dataset for audio events,” IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 776-780, 2017.

[24] L. Panait, “A large-scale dataset of manually annotated audio events,” AudioSet, 2017.

[25] A. McDonagh, “Toolkit for downloading and processing Google's AudioSet dataset,” GitHub, 2021.

[26] T. Viarbitskaya, and A. Dobrucki, “Audio processing with using Python language science libraries,” Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), pp. 350-354, 2018.

[27] McFee, Brian, C. Raffel, D. Liang, D. PW. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” In Proceedings of the 14th python in science conference, pp. 18-25, 2015.

[28] Y. Cai, L. Tan and J. Chen, “Evaluation of Deep Learning Neural Networks with Input Processing for Bearing Fault Diagnosis,” 2021 IEEE International Conference on Electro Information Technology (EIT), 2021, pp. 140-145.

[29] Y. Song, Y. Cai and L. Tan, “Video-Audio Emotion Recognition Based on Feature Fusion Deep Learning Method,” 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 2021, pp. 611-616.

[30] L. Tan, J. Jiang, Digital Signal Processing: Fundamentals and Applications. Third Edition, Elsevier/Academic Press, 2018.

[31] E. O. Brigham, and R. E. Morrow, “The fast Fourier transform,” in IEEE Spectrum, vol. 4, no. 12, pp. 63-70, 1967.

[32] M. A. Hossan, S. Memon, and M. A. Gregory, “A novel approach for MFCC feature extraction,” IEEE 4th International Conference on Signal Processing and Communication Systems, pp. 1-5, 2010.

[33] T. Chen, L. -q. Han, S. -x. Xing, H. -r. Wang, and K. -p. Wang, “STFT-Based Comparative Studies of Heart Sound Signals,” IEEE 3rd International Conference on Bioinformatics and Biomedical Engineering, pp. 1-4, 2009.

[34] A. Shrestha, and A. Mahmood, “Review of Deep Learning Algorithms and Architectures,” in IEEE Access, vol. 7, pp. 53040-53065, 2019.

[35] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. -Y. Chang, and T. Sainath, “Deep Learning for Audio Signal Processing,” in IEEE Journal of Selected Topics in Signal Processing, vol. 13, no. 2, pp. 206-219, 2019.

[36] R. E. Uhrig, “Introduction to artificial neural networks,” IEEE Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics, vol. 1, pp. 33-37, 1995.

[37] M. Mishra, and M. Srivastava, “A view of Artificial Neural Network,” IEEE International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), pp. 1-3, 2014.

[38] X. Yang, P. Roop, H. Pearce, J. W. Ro, “A compositional approach using Keras for neural networks in real-time systems,” IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1109-1114, 2020.

[39] S. Kim, H. Wimmer, and J. Kim, “Analysis of Deep Learning Libraries: Keras, PyTorch, and MXnet,” IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA), pp. 54-62, 2022.

[40] M. M. M. Sukri, U. Fadlilah, S. Saon, A. K. Mahamad, M. M. Som, and A. Sidek, “Bird Sound Identification based on Artificial Neural Network,” IEEE Student Conference on Research and Development (SCOReD), pp. 342-345, 2020.

[41] A. Kaplunovich, and Y. Yesha, “Automatic Tuning of Hyperparameters for Neural Networks in Serverless Cloud,” IEEE International Conference on Big Data (Big Data), pp. 2751-2756, 2020.

[42] M. Özdeş, and B. M. Severoğlu, “Sound Spectrum Detection Using Deep Learning,” IEEE Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), pp. 1-4, 2019.

[43] M. S. Imran, A. F. Rahman, S. Tanvir, H. H. Kadir, J. Iqbal, and M. Mostakim, “An Analysis of Audio Classification Techniques using Deep Learning Architectures,” IEEE 6th International Conference on Inventive Computation Technologies (ICICT), pp. 805-812, 2021.

[44] F. F. Firdaus, H. A. Nugroho, and I. Soesanti, “Deep Neural Network with Hyperparameter Tuning for Detection of Heart Disease,” IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), pp. 59-65, 2021.

[45] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” IEEE International Conference on Engineering and Technology (ICET), pp. 1-6, 2017.

[46] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,” in IEEE Transactions on Neural Networks and Learning Systems, 2021.

[47] D. Dai, “An Introduction of CNN: Models and Training on Neural Network Models,” IEEE International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR), pp. 135-138, 2021.

[48] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, “Convolutional Neural Network (CNN) for Image Detection and Recognition,” IEEE First International Conference on Secure Cyber Computing and Communication (ICSCCC), pp. 278-282, 2018.

[49] S. Hershey et al., “CNN architectures for large-scale audio classification,” IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 131-135, 2017.

[50] M. Massoudi, S. Verma, and R. Jain, “Urban Sound Classification using CNN,” IEEE 6th International Conference on Inventive Computation Technologies (ICICT), pp. 583-589, 2021.

[51] P. Chagas et al., “Evaluation of Convolutional Neural Network Architectures for Chart Image Classification,” IEEE International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2018.

[52] S. Kiranyaz, T. Ince, O. Abdeljaber, O. Avci, and M. Gabbouj, “1-D Convolutional Neural Networks for Signal Processing Applications,” ICASSP - IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8360-8364, 2019.

[53] K. Liang, N. Qin, D. Huang, L. Ma, Y. Fu, and C. Chen, “1D Convolutional Neural Networks For Fault Diagnosis of High-speed Train Bogie,” IEEE 23rd International Conference on Digital Signal Processing (DSP), pp. 1-5, 2018.

[54] S. Allamy, and A. L. Koerich, “1D CNN Architectures for Music Genre Classification,” IEEE Symposium Series on Computational Intelligence (SSCI), pp. 01-07, 2021.

[55] M. Muthumari, C. A. Bhuvaneswari, J. E. N. S. Kumar Babu, and S. P. Raju, “Data Augmentation Model for Audio Signal Extraction,” IEEE 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 334-340, 2022.