CONSERVAR PATRIMÓNIO

Conservation of industrial and technological heritage
Conservação de património tecnológico e industrial
in Press **ARTICLE / ARTIGO**

Data analysis techniques for the visualization and classification of historical vehicle engines' health status using data-driven solutions

Técnicas de análise de dados para a visualização e classificação do estado de conservação de motores de veículos históricos

STEFANO CARRINO [1]
LUCA MEYER [1]
JONATHAN DREYER [1]
BRICE CHALANÇON [2]
ALEJANDRO RODA-BUCH [3,4]
LAURA BRAMBILLA [3*]

1. HES-SO // University of Applied Sciences and Arts Western Switzerland, Haute Ecole Arc Ingénierie, Espace de l'Europe 11, 2000 Neuchatel, Switzerland

2. Association de Gestion du Musée National de l'Automobile, 188 Av. de Colmar, 68100 Mulhouse, France

3. HES-SO // University of Applied Sciences and Arts Western Switzerland, Haute Ecole Arc Conservation-restauration, Espace de l'Europe 11, 2000 Neuchatel, Switzerland

4. Ecole Polytechnique Fédérale de Lausanne, EPFL CH-1015, Lausanne, Switzerland

**laura.brambilla@he-arc.ch*

Abstract

In the field of cultural heritage, the use of non-destructive techniques to determine the state of conservation of an artifact is of the utmost importance, to avoid damage to the object itself. In this paper, we present a data pipeline and several machine learning techniques for the visualization, analysis and characterization of engines in historical vehicles. The paper investigates the use of vibro-acoustic signals acquired from the engines in different states of conservation and working conditions to train machine learning solutions. Data are classified according to their state of health and the presence of anomalies. The t-SNE algorithm is used for dimensionality reduction for data visualization. The machine learning algorithms tested showed encouraging performance in associating acoustic emission data with the engine signature, the type of anomaly and the working conditions. Nevertheless, a larger dataset would allow us to improve and strengthen the results.

KEYWORDS
Machine learning
Cultural heritage
Non-invasive monitoring
Acoustic emission

Resumo

Em património cultural a utilização de técnicas não destrutivas para determinar o estado de conservação de um artefacto é de extrema importância para evitar danos no próprio objeto. Neste artigo, apresentamos um canal de dados e técnicas de aprendizagem automática para a visualização, análise e caraterização de motores de veículos históricos. O artigo investiga a utilização de sinais vibro-acústicos adquiridos nos motores em diferentes estados de conservação e condições de funcionamento para treinar soluções de aprendizagem automática. Os dados são classificados de acordo com o seu estado de conservação e a presença de anomalias. O algoritmo t-SNE utilizou-se para a redução da dimensionalidade para a visualização dos dados. Os algoritmos de aprendizagem automática testados revelaram um desempenho promissor na associação dos dados de emissões acústicas com a assinatura do motor, o tipo de anomalia e as suas condições de funcionamento. Porém, um maior conjunto de dados permitir-nos-ia melhorar e reforçar os resultados.

PALAVRAS-CHAVE
Aprendizagem automática
Património cultural
Monitorização não invasiva
Emissão acústica

Introduction

As known, employing non-destructive techniques is fundamental in the field of cultural heritage when ascertaining the state of conservation of an artifact and monitoring it in order to prevent damaging the object itself. When considering technical, scientific and industrial heritage, the artifact's state of conservation is not limited to the conditions of the constituting materials (e.g., the presence of corrosion on a metallic object), but includes also the functioning status of the mechanisms [1-5].

For this, Non-Destructive Testing (NDT) methods are one of the possible techniques that can be used to provide data in order to support the expertise of the conservators to decide if a mechanism can be reactivated. One of the most versatile NDT methods is Acoustic Emission (AE). AE techniques are based on the principle that when there is a sudden release of energy inside a solid caused, for instance, by an impact between moving parts, corrosion phenomena or the growth of a crack, then some energy is dissipated in the form of elastic waves travelling inside the material itself. This release of energy can be captured by piezoelectric sensors placed in contact with the surface to be analyzed [6].

The ACUME_HV project (Acoustic Emission Monitoring of Historical Vehicles), conducted at HE-Arc between 2018 and 2020, was the first attempt to use Acoustic Emission techniques as NDT, to diagnose and monitor the state of preservation of engines in historical vehicles. AE is commonly used in the automotive industry to test the performance, in terms of mechanical and combustion fulfillments, of newly produced engines [7-13]. However, the technique has not been used before on ancient engines, where scaling of lubricants, long periods of inactivity, wear of the moving parts and presence of oxide scales can significantly affect the engine mechanism's performance. AE therefore needs to be adapted for this purpose.

AE has also been extensively used in the industrial field in the last 70 years for the detection and monitoring of a wide variety of phenomena ranging from corrosion of metal structure to the stability of buildings [6]. In the field of cultural heritage, this technique has gained acceptance since it was introduced in the 1990s [14]. Being non-invasive, it is used for the monitoring of ongoing processes, such as crack development and propagation in different materials such as wood and enamel, or the activity of insects in wooden instruments, just to cite some examples [15-16]. AE can therefore be considered a technique with high potentiality for the application in the field of cultural heritage and in particular of heritage with mechanisms (technical, scientific and industrial artefacts), but the three main drawbacks are the cost of the equipment, the expertise necessary to perform the measurements and the data post-processing. The latter is part of the focus of the present paper.

The developed methodology and the results of the ACUME_HV project have been previously published in different journals and conference proceedings [17-21]. The ACUME_HV project showed that acoustic emission can be used to establish the state of preservation of historical engines before their reactivation and to detect the most common faults [17-21]. Obtained AE data can be post-processed and interpreted in different ways, based on the extraction of specific features, such as peak amplitude, duration, rise time, frequency, and energy [14]. In the first phase of the project, the data analysis was limited to the detection and evolution over time of AE events' amplitude (also called "hits"), as a function of the rotational speed of the engine [17-21]. Those events are produced by the movement of the different contact pairs during the functioning of the engine and are directly related to the thermodynamic cycle of the engine itself (fuel injection, ignition, expansion, and exhaust) [17-21]. Interpretation of the data in this phase was limited to combining the obtained measurements with a good knowledge of the system and with the expertise of conservation professionals. When anomalies are detected in

the data, only the competence and previous knowledge of the operators, as well as the combination with other types of tests, could lead to a correct interpretation of the curves. It remained therefore a human-dependent methodology.

For this reason, in the following phase of the project, it was decided to process the collected data using advanced data analysis methods, thanks to the collaboration between the conservation-restoration and the engineering departments of HE-Arc. The latter had previous experience in the application of machine learning (ML) techniques to AE data [22]. Supervised ML techniques learn from expert-labelled data the pattern and characteristics intrinsic to the signals and how they relate to normal and anomalous behavior. Once the algorithms are trained, the resulting models can be automatically applied to novel unseen data without the need for human supervision. This short paper presents this second, data-driven, phase of the study.

Goal and research questions

In this paper, we investigate the use of data analysis techniques for the visualization and classification of the engine health status using data-driven solutions such as supervised machine learning. The paper focuses on the analysis of vibro-acoustic signals for the characterization of engines in different conditions (data presented in previous publications [17-21]). Research questions studied in this article focus on the use of AE signals and machine learning, and can be summarized as: 1) identify the engine that produces the signal; 2) determine if an engine has an anomaly; 3) determine which engine a record belongs to and if it has an anomaly.

Research goals 1 and 3 aim at verifying if data analysis techniques can, per se, distinguish data produced by different engines of the same type (in this case, Renault AG1). In fact, every engine is unique and produces peculiar vibro-acoustic signatures but, in this study, we wanted to verify if the techniques used are able to differentiate those unique peculiarities. If a database of AE signals from different engines is produced (which would be the natural development of this study), the method should be able to identify precisely which engine has produced the signals. In particular, the type of engine as well as the precise engine should be identifiable.

Related work

The proposed approach is inspired by Zhe Li et al. [23] dealing with the detection of anomalies from vibration data on mechanical equipment. The authors focus on the detection of anomalies in the context of unsupervised learning, and with unclassified data, on mechanical equipment.

An important point to note is that the sample rate of the data used is 4096 Hz [23], where our audio recordings are sampled at 2 MHz. In addition, the data used are accelerometer recordings [23], while we investigate the usage of acoustic emission sensors. We decided to focus on AE because these signals are commonly used in the automotive industry to test the performance, in terms of mechanical and combustion fulfillments and have the potential to detect physical phenomena that could be missed at lower frequencies [7-13]. Nevertheless, due to the analog nature of the signal in both cases, the approach proposed by Zhe Li et al. [23] can be adapted to the current use case.

The method used is based on two deep learning models: Stacked Autoencoders (SAE) and LSTM (Long Short-Term Memory) [23]. The SAE role is to find an encoded representation of the input signals by forcing the extracted features into an encoder-decoder structure. The encoder will force the reduction of the signal dimensionality while the decoder aims at reconstructing the original data. Then, the LSTM model is used in an unsupervised fashion in order to detect anomalies in the reconstructed signal. LSTMs are predominantly used to learn, process, and

classify sequential data because these networks can learn long-term dependencies between time steps of data. In our scenario, time dependencies are less relevant since we want to classify if an engine has an anomalous behavior, but we do not need to characterize its evolution over time. A faulty engine in our dataset will produce anomalous data from the beginning. For this reason, in this study we limit our implementation to the feature extraction steps and SAE model. The LSTM is not used.

Methods

Data acquision

AE signals were acquired using an AE system from Vallen. The set-up includes an MB2-V1 chassis equipped with four AE and four parametric input channels. The system included four broadband AE sensors VS900-M (between 100 and 900 kHz) and respective AEP5 preamplifiers (+ 34 dB). AE signals were recorded with the sampling frequency set to 2 MHz. A 1 MHz low pass filter was applied to the input signals, aiming to reduce the noise from the high frequencies. Additionally, the angular position of the engine's crankshaft was measured by means of a full continuous 360° VISHAY Spectrol 601-1045 smart position sensor (output signal 0-5 V) with sampling rate of 1.25 kHz [17-21].

The engine type used for this project is a two-cylinder AG1 by Renault, dating from the 1910s [17-21]. Three identical engines (hereafter indicated as ENG0, ENG1 and ENG2) were used: two of them are in working conditions (ENG1 and ENG2) [17] and one, ENG0 [17-21] that was previously bought as spare part for possible reparations. The location of the AE sensors around the engine was established during the first part of the ACUME_HV project [17-19, 21].

The engine's mechanisms were operated manually, at a relatively low rotating speed, by means of an external handle to move the crankshaft. This procedure allowed to maintain good control of the interaction of the different parts during pistons slow motion, therefore avoiding possible damage to the mechanism, which is a museum standards requirement. During these so-called "cold tests" the mechanical and air compression signatures of the engine were extracted [17-21]. During those tests, the influence of the rotational speed, as well as the reproducibility of the measurements were verified [17-21].

Moreover, some malfunctions, commonly encountered in this type of engine, were simulated on one of the available AG1 engines: ENG0. The engine was removed from a vehicle and mounted on a test bench, in order to have complete access to all the parts of the engine itself [17, 20].

The whole procedure and set-up for AE measurements on AG1 engines have already been described in previous publications [17-21]. To summarize four categories of failure were introduced: removal of a spark plug (F0), added clearance between the connecting rod and the crankshaft (connecting rod failure) (F1); wear of the piston rings (F2); clearance between the piston pin and the connecting rod (F3) [20]. Table 1 summarizes the different conditions and the labelling used for classification (see "Predict both the presence of an anomaly and the engine", below).

Table 1. *Classes for anomaly and engine prediction.*

Engine	Configuration	Class (label)
ENG0	No anomaly, with spark plugs	0
	No anomaly, without spark plugs (F0)	1
	F1	2
	F2	3
	F3	4
ENG1	No anomaly, with spark plugs	5
	No anomaly, without spark plugs (F0)	6
ENG2	No anomaly, with spark plugs	7
	No anomaly, without spark plugs (F0)	8

Data analysis technique – overview

There is a wide range of machine learning (ML) techniques utilized for detecting anomalies in time series data [24-26]. However, there is a lack of approaches that specifically target the identification of anomalies in AE signals. In a recent study, Zhe Li et al. [23] proposed a deep learning-based approach that demonstrated promising results for detecting anomalies in accelerometer data. Given that the underlying nature of accelerometer time series and AE time series is comparable, we investigated the potential feasibility and necessary procedures for adapting the aforementioned architecture to our use case and signal data. In the article by Zhe Li et al. [23], from a signal recorded by an accelerometer mounted on the machine which has X-Y-Z acceleration components, the characteristics of these signals are extracted by means of a Wavelet Packet Decomposition solution (WPD) [27] (see "Wavelet Packet Decomposition", below). These characteristics are 33 in number (11 per component) and contain the following information:

- Features 1 to 6 represent the two largest peaks in the frequency domain of each of the components (e.g., features 1 and 2 represent, respectively, the largest frequency along the X-axis signal as well as the second largest).
- Features 7 to 18 represent the standard deviation of the detail of the coefficients of the wavelets.
- Features 19 to 21 are the energy percentages of the approximation in the three directions.
- Features 22 to 33 are the energy percentages of the details in levels 1 to 4, again for each of the coordinates.

The characteristics are then normalized with a *Unity Based Normalization*, which simply corresponds to a normalization between 0 and 1 of the values (Equation 1).

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad \textbf{(1)}$$

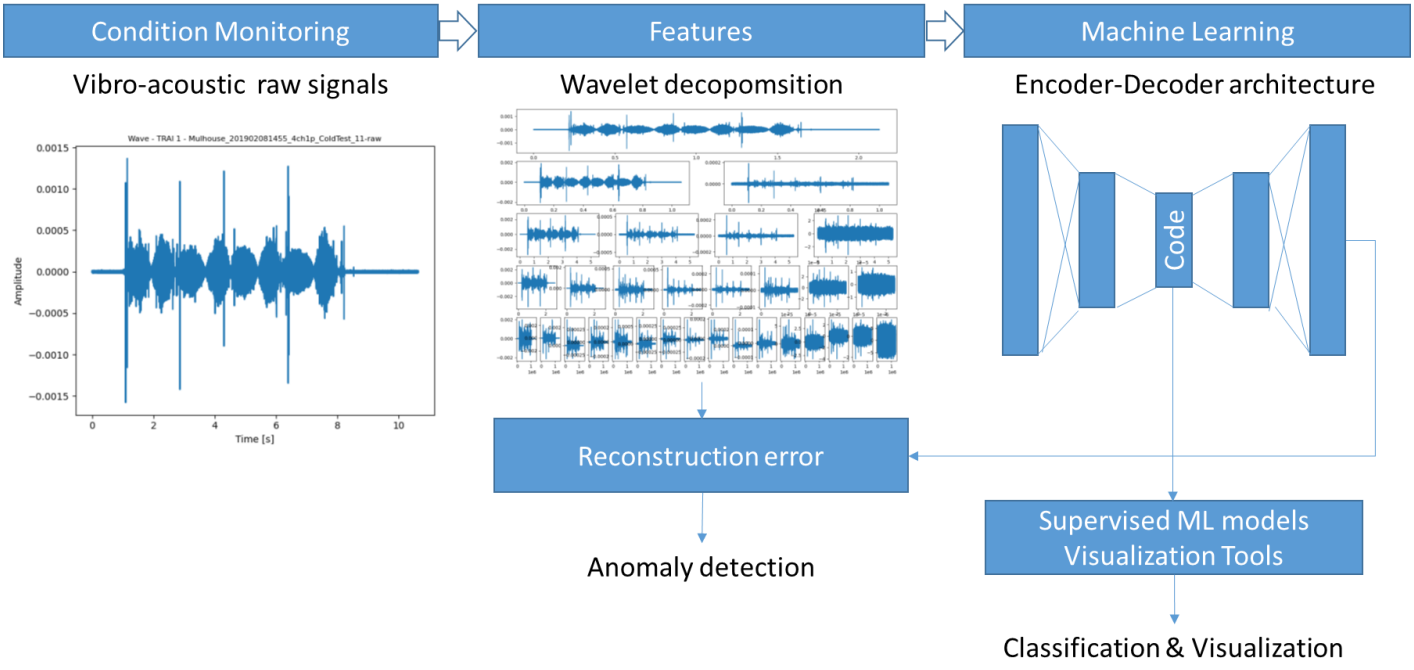

Figure 1. The used data engineering pipeline, adapted from Zhe Li et al. [23].

The extracted features are then sent through an encoder-decoder machine learning solutions. In Zhe Li et al. [28] a LSTM model is used for the automatic thresholding and anomaly detection. The LSTM is not used in this paper, in which we focus on AE analysis. The main difference from a traditional accelerometer sensor or microphone is the much higher sample rate (up to 2 MHz).

Features extracted from the raw signals are passed to a machine learning module using an encoder-decoder architecture. The "code" created by the model is used to compute a reconstruction error. The reconstruction error is the difference between the input signal and the signal reconstructed by the decoder. The underlying hypothesis is that bigger errors are observed in the case of anomalies. Figure 1 depicts the different steps.

The following sections describe the data pipeline implemented to realize such an approach and highlight the differences from the original article.

Feature extraction

No data pre-treatment

Some signals in the dataset are noisy. Filters can be applied to mitigate the impact of noise on the informative part of the signal. Since feature learning is one of the major advantages of deep learning approaches, in this study, we decided to let the machine learning logic filters it out on its own.

Fast Fourier Transform

The Fast Fourier Transform (FFT) allows a signal to be decomposed into its component frequencies and the data to be visualized and analyzed in the frequency domain. An anomaly in mechanical equipment can take the form of a frictional force that can generate high frequencies and therefore would be visible on an FFT graph (Figure 2).

Engine problems can produce signal changes that are not easily detected by a human, such as a shift in the frequency peak, or a combination of other factors that a machine learning algorithm would be able to detect.

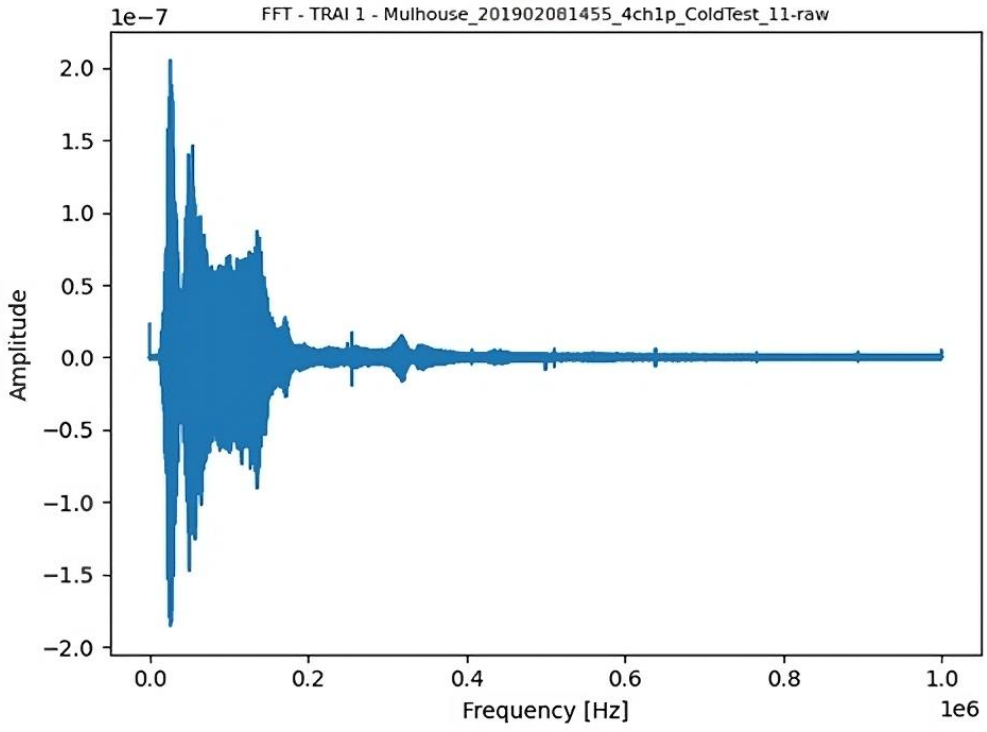

Figure 2. FFT of an input signal.

Spectrogram

A spectrogram is a visual representation of the frequency spectrum of a signal as it varies with time. This representation visualizes the distribution of frequencies and their evolution over time. The amplitude of a frequency is represented by a color code (Figure 3). Logarithmic values are used to visually highlight changes in the values.

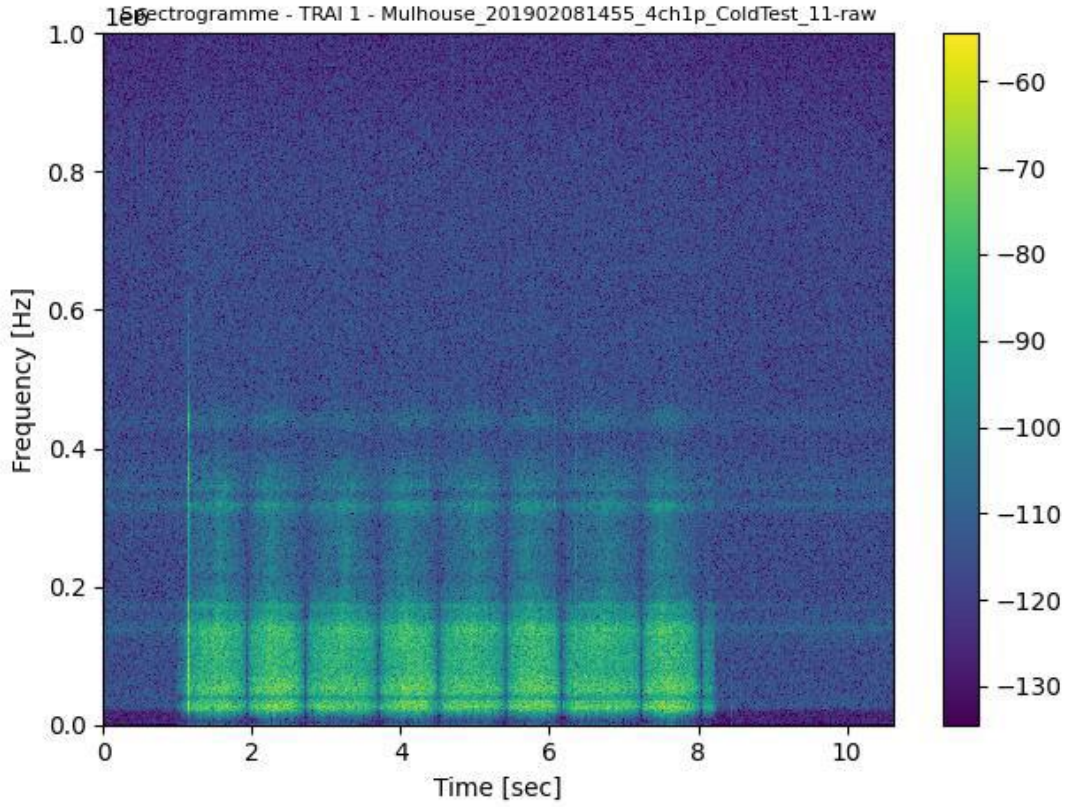

Figure 3. Spectrogram of the acoustic signal, yellow is used to represent higher amplitudes and blue represents the portion of the spectrum without relevant signals.

Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) [29] is, as the name implies, a signal transformation technique based on wavelets, which are simple waveforms with an amplitude that begins at zero, increases or decreases, and then returns to zero one or more times. The DWT decomposes a given signal into several sets, where each set is a time series of coefficients describing the time evolution of the signal in the corresponding frequency band. The original signal can be reconstructed as a linear combination of wavelets and weighting wavelet coefficients. In the case of our project, we worked with the Daubechies 4 (DB4) wavelet.

For example, a DWT decomposes the signal into two equal parts, a high-frequency part and a low-frequency part. Each additional level of decomposition divides the lower part into an upper part and a lower part, as illustrated in the image below (Figure 4).

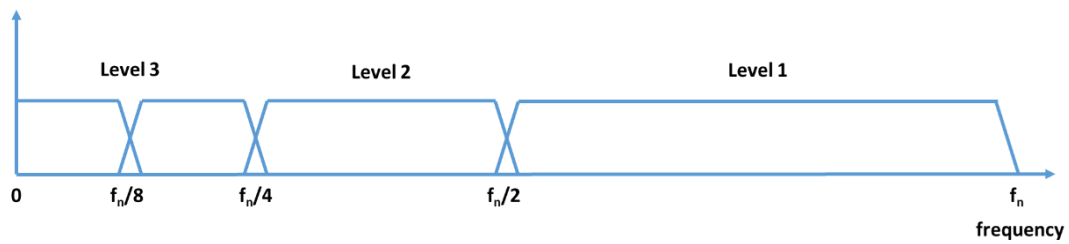

Figure 4. Frequency structure of a DWT.

Wavelet Packet Decomposition

In our case, we did not use the DWT, but the Wavelet Packet Decomposition (WPD). DWT and WPD are relatively similar, with the major difference that WPD also breaks the high part into a high part and a low part. More precisely the lower part is called Approximation (A) and the upper part Detail (D). In other words, WPD is identical to DWT except that the detail coefficients are further decomposed as well.

The structure thus generated is that of a binary tree. It is possible to access a decomposition level by specifying its path. For example, the path "A" indicates the first approximation. The access "the approximation of the detail" of the original signal will be indicated using the path "DA".

For illustration, a WPD decomposition into three levels gives the following structure, with each of the elements of the tree, leaves or not, being accessible with their path (

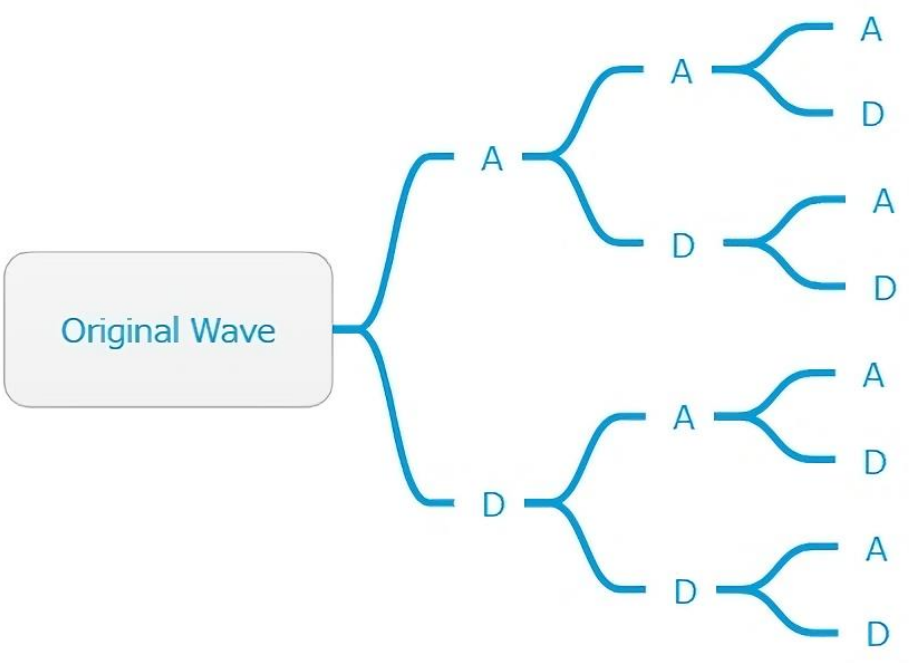

Figure 5).

In the case of the data of our project, the analysis of one of the files in WPD on four levels gives the following result, with the original signal at the very top, and at each line an additional level of decomposition. We therefore notice that at the fourth level we have 16 (= 2^4) coefficients (Figure 6).

In our case, we use the WPDs to calculate the energy level in certain regions of the signal. This allows us to determine if most of the energy is in the high or low frequencies.

To summarize the following features are extracted:

- The two most significant frequencies of the FFT of the whole recording (frequencies where the FFT has the higher magnitude). Frequencies are stored as a multiple of the transient sample rate (dynamically retrieved) divided by 2. In effect this is a multiple of 1 MHz since the recordings were made at 2 MHz.
- The energy levels (WPD coefficients) of the WPD decomposition of the signal at the A, D, AD, AAD and AAAD levels. These values are normalized per channel between 0 and 1 as previously mentioned. These features allow the model to determine at which level is most of the energy in each channel.
- The distribution of the energy of the level A of the WPD of the signal between the four sensors. For example, a value of 0.34 indicates that the channel in question has 34 % of the energy of the total approximation recorded by all the sensors.

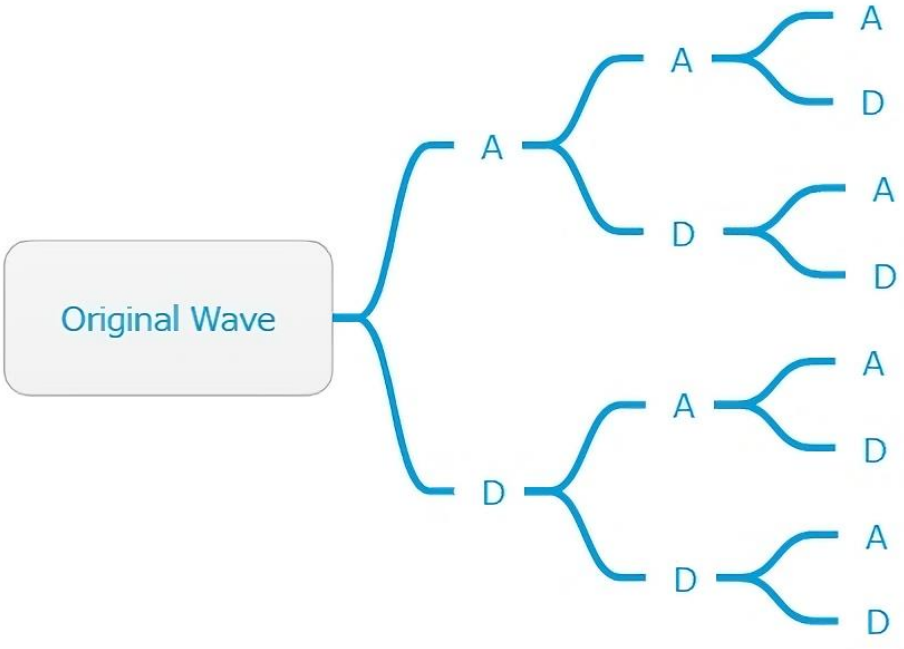

Figure 5. WPD 3-level decomposition.

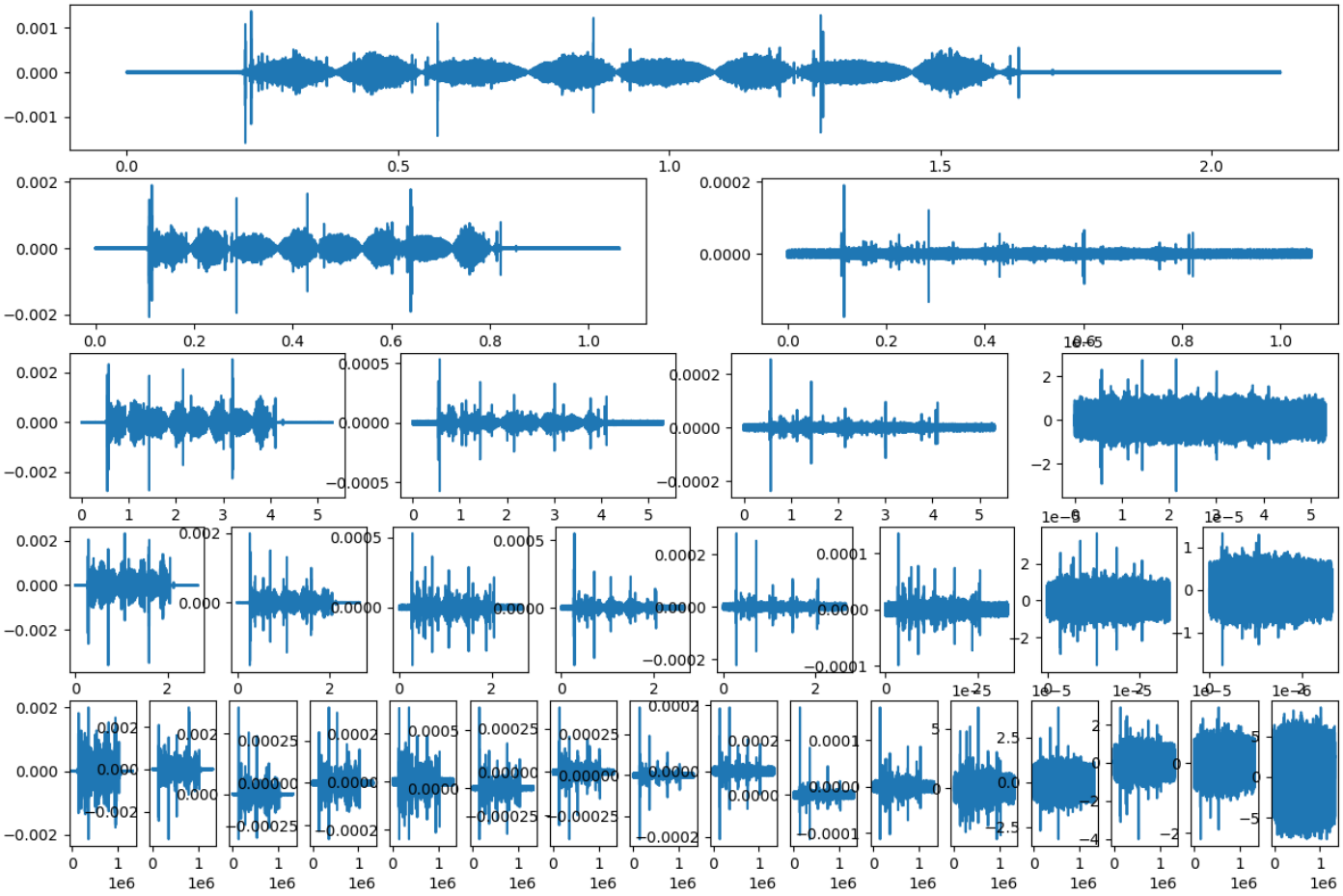

Figure 6. WPD decomposition applied to the studied dataset.

Clustering visualization using t-distributed Stochastic Neighbor Embedding

A data clustering and visualization step is then performed in order to visually assess the quality of the selected features. The approach used is based on t-SNE [30]. The acronym t-SNE stands

for *t-distributed Stochastic Neighbor Embedding*. t-SNE makes it possible to represent datasets with many dimensions and reduce them to a low-dimensional graph, while keeping a large amount of the original information.

The principle of functioning is as follows: the algorithm first calculates the similarity between the two points to give them greater or lesser importance. The similarities used in t-SNE are based on a Gaussian kernel that measures the similarity between pairs of high-dimensional data points. The similarity between two data points is higher if they are close to each other in the high-dimensional space, and lower if they are far apart. This is done between each pair of points in the original dataset. It then places them randomly in a lower dimensional space (e.g., 2D) and then calculates the similarity of these points in this new space. At the first iteration the similarity will be very low, so the algorithm proceeds to bring similar points closer together in small steps and move away those that are not similar. The algorithm repeats this step several times until it obtains similarity values closest to those of the original dataset. t-SNE still takes an additional parameter called "perplexity" which gives it an additional indication of the size of the groups it should form.

This approach, not presented in the original paper has been added, to facilitate the understanding of the data. The assumption is that clusters visible using such an approach can later be identified using machine learning.

Classification using machine learning – methodology overview

Classification using machine learning typically consists of three main phases: feature extraction, algorithm training and evaluation. The first phase is to extract the features used by the algorithm from the engine records (the approach used is presented in the previous section). The second phase is the training of a machine learning algorithm using the extracted features and the labels. Labels associate the features to a class. This phase is typically repeated iteratively to tune and optimize the algorithm hyperparameters. Finally, the third step consists in evaluating the performances of the trained algorithm on a test set of data unseen by the algorithm during the training phase. This split of the data in training and test set is a typical approach in ML required to demonstrate the generalization of the model and reducing the risk of overfitting the model to the data.

Results

Visualization with t-SNE

Figure 7 shows the results of the clustering done with t-SNE. Signals representing the behavior of an engine are represented by a point in a 2D space. Points with the same color belong to the same class, a number is introduced to link the cluster to the selected class ("0" stands for ENG0, etc.). Points that are displayed in proximity indicate that t-SNE considers such points similar. In other words, if points of the same color are close, it means that also the associated signals are similar.

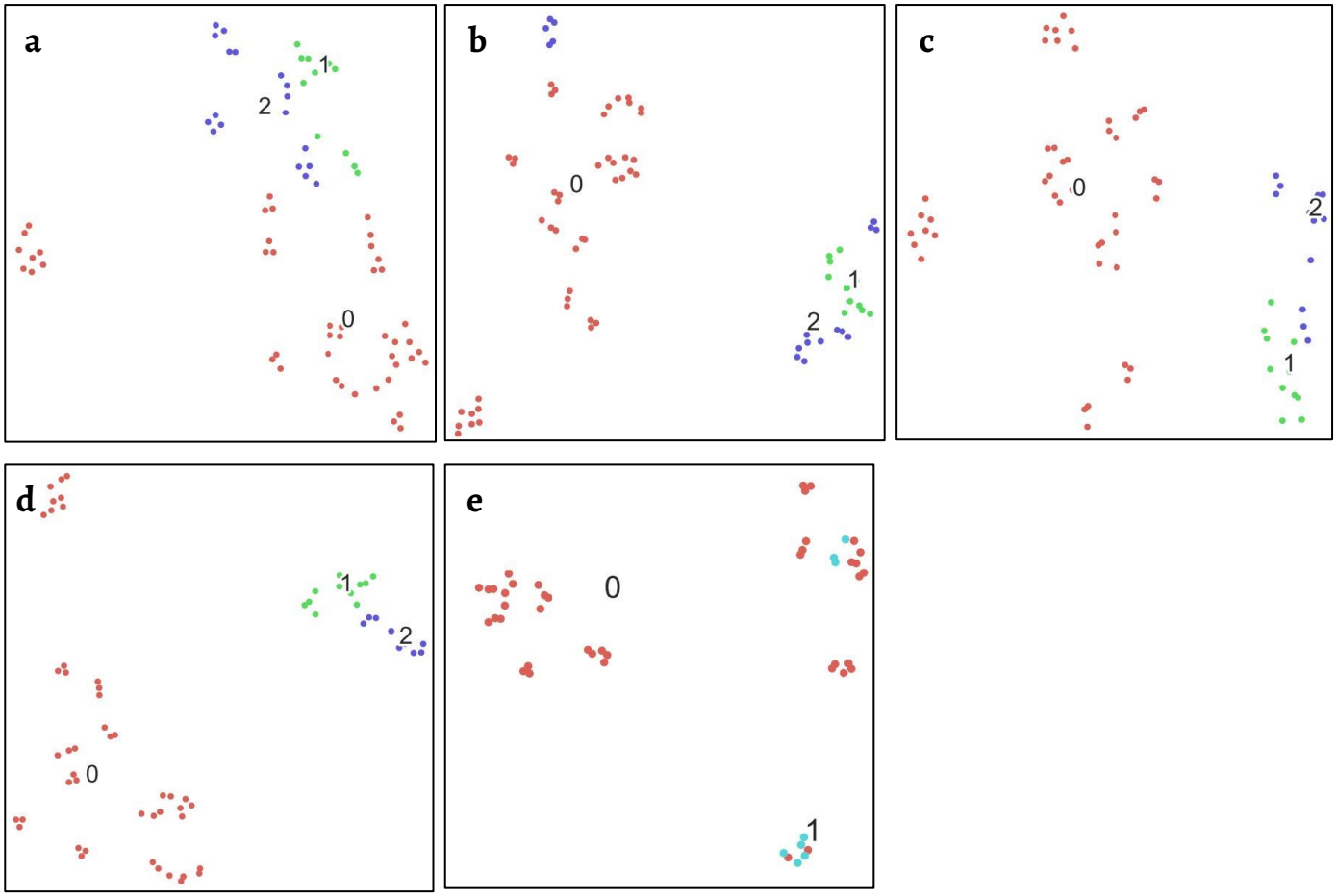

Figure 7. t-SNE visualization in a 2D space, datapoints from the same class (number) have the same color (0-red: ENG0; 1- green: ENG1; 2-blue: ENG2): *a)* engine classification using only 2 features - FFT picks and WPD coefficients; *b)* same but using power distribution between sensors as additional feature; *c)* data obtained without spark plugs are removed from ENG0; *d)* data obtained without spark plugs are removed from all the engines' data; *e)* visualization differentiating engines without (0) and with anomalies (1).

Discussion

Although the dataset that has been used is quite small and heterogeneous (for instance, ENG0 sensor positioning differs from ENG1 and ENG2, the recordings in general are not all made under the same conditions), t-SNE allowed us to clearly identify some special cases and to isolate them. For instance, a sub-cluster of points was associated to engines data obtained with the removal of spark plugs. This is a good first result at the level of machine learning and confirms our hypothesis that classification seems to be a viable option for detecting anomalies in engines and motivate our next step. Nevertheless, it could be necessary to increase the quantity of extracted features from the signals for the clusters to be more distinct from each other.

Classification using machine learning

Libraries and metrics

LazyPredict [31] is a Python library that makes it easy to test a large number of models to determine their effectiveness on the data provided. The data is loaded into a matrix (or more precisely into an array of arrays) and the library takes care of testing the combinations.

The different datasets previously tested with t-SNE are also used to investigate how effectively the different classes could be predicted. With a dataset with a sample size of 50 labelled instances, 70 % of the data is used for training and 30 % for testing. This split has been selected experimentally. We needed at least 30 % of the data to obtain statistically significant results and as much as possible of the data for the training. As for the metrics used for evaluate the algorithms' performance we selected the accuracy, the balanced accuracy and the F1-Score.

Accuracy is a measure of how many predictions the model got correct out of the total number of predictions made (Equation 2). It is defined as the ratio of the number of correct predictions (true positives and true negatives) to the total number of predictions made:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (2)$$

Where, TP – number of true positives (number of samples from the positive class that are correctly classified as positive); TN – number of true negatives (number of samples from the negative class that are correctly classified as negative); FP – number of false positives (number of samples from the negative class that are incorrectly classified as positive); FN – number of false negatives (number of samples from the positive class that are incorrectly classified as negative).

The F1-score is a harmonic mean of precision and recall, two metrics that describe the classifier's performance with respect to each class separately (Equation 3).

$$F1\text{-}score = \frac{2 * (precision * recall)}{precision + recall} \qquad (3)$$

Where, precision = TP / (TP + FP) is the fraction of true positives among all the samples that the model classified as positive; recall = TP / (TP + FN) is the fraction of true positives that the model correctly classified as positive out of all the positive samples.

The F1-score is a useful metric when dealing with imbalanced datasets, where one class has fewer samples than the other. It provides a balanced measure of precision and recall that takes into account the class distribution.

Similarly, "balanced accuracy" is a performance metric that measures the accuracy of a classifier in a dataset that has imbalanced class distribution. It is defined as the average of the sensitivity and specificity of the classifier, which are the true positive rate and true negative rate respectively, calculated for each class (Equation 4). More formally, if we have a binary classification problem with classes A and B, and N_A and N_B are the number of samples in each class, the balanced accuracy can be calculated as:

$$balanced\_accuracy = (sensitivity_A + specificity_B)/2 \qquad (4)$$

Where: $sensitivity_A = TP_A / N_A$ is the true positive rate for class A, i.e., the fraction of samples from class A that are correctly classified as A; $specificity_B = TN_B / N_B$ is the true negative rate for class B, i.e., the fraction of samples from class B that are correctly classified as B; TP_A – number of true positives for class A; TN_B – number of true negatives for class B. In a multiclass classification problem, the balanced accuracy can be computed as the average of the per-class balanced accuracies.

Predict which engine a record belongs to

It is in this scenario that the classification works best, as many ML algorithms are able to classify the engines and the AE signal with an accuracy of 1.0. As a baseline, we also note the score of 0.36 for the "Dummy Classifier", which simply predicts the most frequent class for all data points. Therefore, the classification seems to provide valuable results (Table 2).

Table 2. *Classification results for engine identification (output of the LazyPredict library). Results are sorted by "balanced accuracy".*

Model	Accuracy	Balanced accuracy	F1 score
LinearSVC	1.00	1.00	1.00
LogisticRegression	1.00	1.00	1.00
XGBClassifier	1.00	1.00	1.00

SGDClassifier	1.00	1.00	1.00
RidgeClassifierCV	1.00	1.00	1.00
RidgeClassifier	1.00	1.00	1.00
RandomForestClassifier	1.00	1.00	1.00
Perceptron	1.00	1.00	1.00
PassiveAggressiveClassifier	1.00	1.00	1.00
NearestCentroid	1.00	1.00	1.00
LGBMClassifier	1.00	1.00	1.00
ExtraTreesClassifier	1.00	1.00	1.00
BernoulliNB	1.00	1.00	1.00
BaggingClassifier	0.95	0.97	0.95
GaussianNB	0.95	0.96	0.96
CalibratedClassifierCV	0.95	0.96	0.95
LinearDiscriminantAnalysis	0.95	0.92	0.95
ExtraTreeClassifier	0.95	0.92	0.95
KNeighborsClassifier	0.91	0.88	0.91
AdaBoostClassifier	0.91	0.88	0.91
DecisionTreeClassifier	0.86	0.84	0.87
SVC	0.91	0.83	0.90
LabelSpreading	0.86	0.79	0.85
LabelPropagation	0.86	0.79	0.85
QuadraticDiscriminantAnalysis	0.64	0.54	0.59
DummyClassifier	0.36	0.32	0.29

Predict if an engine has an anomaly

The dataset is divided into two classes, one containing only signals without problems and the other containing signals with an anomaly. In this scenario, it does not matter which dataset the signal belongs to.

In contrast to the t-SNE results, the accuracy of the different models is very promising (Table 3), with the majority of classifiers outperforming the Dummy Classifier. Note that the amount of data in the class with anomalies is much smaller than in the class without anomalies, hence the high accuracy of the Dummy Classifier (which, as a reminder, always tries to predict the most frequent class).

Table 3. *Classification results for anomaly detection (output of the LazyPredict library). Results are sorted by "balanced accuracy".*

Model	Accuracy	Balanced accuracy	F1 score
AdaBoostClassifier	0.93	0.96	0.94
XGBClassifier	0.93	0.96	0.94
DecisionTreeClassifier	0.93	0.96	0.94
ExtraTreesClassifier	0.93	0.96	0.94
GaussianNB	0.87	0.92	0.88
LabelPropagation	0.87	0.79	0.87
PassiveAggressiveClassifier	0.87	0.79	0.87
NearestCentroid	0.87	0.79	0.87
BaggingClassifier	0.87	0.79	0.87
LabelSpreading	0.87	0.79	0.87
Perceptron	0.87	0.79	0.87
RandomForestClassifier	0.87	0.79	0.87
BernoulliNB	0.87	0.79	0.87
LinearDiscriminantAnalysis	0.67	0.67	0.70
GOClassifier	0.80	0.62	0.78
RidgeClassifier	0.80	0.62	0.78
SVC	0.80	0.62	0.78
RidgeClassifierCV	0.80	0.62	0.78
LinearSVC	0.80	0.62	0.78
LogisticRegression	0.80	0.62	0.78
KNeighborsClassifier	0.80	0.62	0.78
ExtraTreeClassifier	0.80	0.62	0.78
CalibratedClassifierCV	a.89	0.62	0.78
DummyClassifier	0.73	0.58	0.73
QuadraticDiscriminantAnalysis	0.80	0.50	0.71

LGBMClassifier	0.80	0.50	0.71

Table 4. *Classification of both anomalies and engines (output of the LazyPredict library). Results are sorted by "balanced accuracy".*

Model	Accuracy	Balanced accuracy	F1 score
RidgeClassifier	0.73	0.84	0.75
PassiveAggressiveClassifier	0.73	0.84	0.75
ExtraTreesClassifier	0.73	0.84	0.75
LinearSVC	0.73	0.84	0.75
SOClassifier	0.67	0.77	0.66
RandomForestClassifiep	0.67	0.77	0.68
LabelPropagation	0.67	0.70	0.70
RidgeClassifierCV	0.67	0.70	0.67
LogisticRegression	0.67	0.70	0.70
LinearDiscriminantAnalysis	0.60	0.70	0.59
LabelSpreading	0.67	0.70	0.70
DecisionTreeClassifier	0.67	0.69	0.68
SVC	0.53	0.64	0.46
CalibratedClassifierCV	0.53	0.64	0.41
BaggingClassifier	0.60	0.63	0.61
NearestCentroid	0.60	0.63	0.63
Perceptron	0.53	0.56	0.57
XGBClassifier	0.53	0.51	0.53
GaussianNB	0.53	0.49	0.54
BernoulliNB	0.40	0.43	0.33
ExtraTreeClassifier	0.40	0.39	0.39
KNeighborsClassifier	0.33	0.36	0.31
AdaBoostClassifier	0.13	0.14	0.03
DummyClassifier	0.07	0.14	0.04
LGBMClassifier	0.13	0.14	0.03

Predict both the presence of an anomaly and the engine

In the results presented below, the machine learning algorithms aim to predict which engine a dataset belongs to, whether the engine in question had its spark plugs or not, whether it had an anomaly or not, and which anomaly it was. The different classes are summarized the Table 4.

The results should be considered preliminary due to the small amount of data (150 samples distributed in 9 classes) available per class, as we could see in the same experiment with t-SNE. In such scenarios, a bias in the data could strongly affect the results (positively or negatively).

First, we note the base value of the Dummy Classifier, which is very low given the large number of classes and the fair distribution of the data, as well as the relatively good performance of most of the algorithms. Despite the limitations of the dataset, the results seem to confirm that it is possible to establish a link between the data and the classes, which is quite promising and encourages further exploration.

Conclusion

This paper presents the use of data-driven approaches to visualize and classify the condition of historic engines. Combined with acoustic emission data, these techniques appear to be a viable solution for assessing the condition of an engine in a non-destructive manner. In fact, the chosen data analysis method allows us to classify the measurements in an objective, human-independent manner. However, data is key to the success of machine learning. The amount of data currently available is limited, with significant differences between datasets. A larger database should be created and used to improve the significance of the results and allow the exploration of other techniques. After the acquisition of additional data, the next steps will be

the integration of a deep recurrent neural network for automatic thresholding of the signal and the study of an algorithm for multi-class anomaly detection solution.

REFERENCES

1. Ashton, J.; Hallam, D., 'The conservation of functional objects – An ethical dilemma', *AICCM Bulletin* **16**(3) (1990) 19-26, https://doi.org/10.1179/bac.1990.16.3.003.
2. Lemos, M.; Tissot, I., 'Reflections on the conservation challenges of scientific and technological objects', *Conservar Património* **33** (2020) 24-31, https://doi.org/10.14568/cp2018044.
3. Wain, A., 'The importance of movement and operation as preventive conservation strategies for heritage machinery', *Journal of the American Institute for Conservation* **56**(2) (2017) 81-95, https://doi.org/10.1080/01971360.2017.1326238.
4. Newey, H., 'Conservation and the preservation of scientific and industrial collections', *Studuies in Conservation* **45**(sup1) (2000) 137-139, https://doi.org/10.1179/sic.2000.45.Supplement-1.137.
5. Pye, E., 'Challenges of conservation: working objects', *Science Museum Group Journal* **6** (2016), https://dx.doi.org/10.15180/160608/001.
6. Scruby, C. B., 'An introduction to acoustic emission', *Journal of Physics E Scientific Instruments* **20**(8) (1987) 946-953, https//doi.org/10.1088/0022-3735/20/8/001.
7. Delvecchio, S.; Bonfiglio, P.; Pompoli, F., 'Vibro-acoustic condition monitoring of internal combustion engines: a critical review of existing techniques', *Mechanical Systems and Signal Processing* **99** (2018) 661-683, https://doi.org/10.1016/j.ymssp.2017.06.033.
8. Kaul, B. C.; Lawler, B.; Zahdeh, A., 'Engine diagnostics using acoustic emissions sensors', *SAE International Journal of Engines* **9**(2) (2016) 684-692, https://doi.org/10.4271/2016-01-0639.
9. Douglas, R. M.; Steel, J. A.; Reuben, R. L., 'A study of the tribological behaviour of piston ring/cylinder liner interaction in diesel engines using acoustic emission', *Tribology International* **39**(12) (2006) 1634-1642, https://doi.org/10.1016/j.triboint.2006.01.005.
10. Nivesrangsan, P.; Steel, J. A.; Reuben, R. L., 'Source location of acoustic emission in diesel engines', *Mechanical Systems and Signal Processing* **21**(2) (2007) 1103-1114, https://doi.org/10.1016/j.ymssp.2005.12.010.
11. Wu, J.-D.; Liu, C.-H., 'An expert system for fault diagnosis in internal combustion engines using wavelet packet transform and neural network', *Expert Systems with Applications* **36**(3) (2009) 4278-4286, https://doi.org/10.1016/j.eswa.2008.03.008.
12. Johansson, S.; Nilsson, P. H.; Ohlsson, R.; Rosén, B.-G., 'Experimental friction evaluation of cylinder liner/piston ring contact', *Wear* **271**(3-4) (2011) 625-633, https://doi.org/10.1016/j.wear.2010.08.028.
13. Wei, N.; Gu, J. X.; Gu, F.; Chen, Z.; Li, G.; Wang, T.; Ball, A. D., 'An Investigation into the acoustic emissions of internal combustion engines with modelling and wavelet package analysis for monitoring lubrication conditions', *Energies* **12**(4) (2019) 640-659, https://doi.org/10.3390/en12040640.
14. Łukomski, M.; Bratasz, Ł; Hagan, E.; Strojecki, M.; Laudato Beltran, V., 'Acoustic emission monitoring for cultural heritage', Getty Conservation Institute, Los Angeles (2020).
15. Thickett, D.; Cheung, C. S.; Liang, H.; Twydle, J.; Maev, R. G.; Gavrilov, D., 'Using non-invasive nondestructive techniques to monitor cultural heritage objects', *Insight - Non-destructive testing and Condition Monitoring* **59**(5) (2017) 230-234, https://doi.org/10.1784/insi.2017.59.5.230.
16. Le Conte, S.; Vaiedelich, S.; Thomas, J. H.; Muliava, V.; de Reyer, D.; Maurin, E., 'Acoustic emission to detect xylophagous insects in wooden musical instrument', *Journal of Cultural Heritage* **16**(3) (2015) 338-343, https://doi.org/10.1016/j.culher.2014.07.001.
17. Chalançon, B., *Les mesures d'émission acoustique appliquées aux moteurs d'automobiles de collection patrimoniale comme outil de diagnostic avant la remise en fonctionnement*, Master dissertation, Haute Ecole Arc Conservation-restauration (HES-SO), Neuchâtel (2019) https://doc.rero.ch/record/327368 (accessed 2023-03-30).
18. Roda Buch, A.; Cornet, E.; Rapp, G.; Chalançon, B.; Mischler, S.; Brambilla, L., 'Development of a diagnostic tool based on acoustic emission techniques', in *500 ans de tribologie*, Presses des Mines, Paris (2021) 147-152.
19. Roda Buch, A.; Cornet, E.; Rapp, G.; Chalançon, B.; Mischler, S.; Brambilla, L., 'Fault detection and diagnosis of historical vehicle engines using acoustic emission techniques', *Acta Imeko* **10**(1) (2021) 77-83, https://doi.org/10.21014/acta_imeko.v10i1.853.
20. Brambilla, L.; Chalançon, B.; Roda Buch, A.; Cornet, E.; Rapp, G.; Mischler, S., 'Acoustic emission techniques for the detection of simulated failures in historical vehicles engines', *The European Physical Journal Plus* **136**(6) (2021) 641, https://doi.org/10.1140/epjp/s13360-021-01611-9.
21. Chalançon, B.; Roda Buch, A.; Cornet, E.; Rapp, G.; Weisser, T.; Brambilla, L., 'Acoustic emission monitoring as a non-invasive tool to assist the conservator in the reactivation and maintenance of historical vehicle engines', *Studies in Conservation* (2023), https://doi.org/10.1080/00393630.2023.2183808.
22. Carrino, S.; Guerne, J.; Dreyer, J.; Ghorbel, H.; Schorderet, A.; Montavon, R., 'Machining quality prediction using acoustic sensors and machine learning', *Proceedings* **63**(1) (2020) 31, https://doi.org/10.3390/proceedings2020063031.
23. Li, Z.; Li, J.; Wang, Y.; Wang, K., 'A deep learning approach for anomaly detection based on SAE and LSTM in mechanical equipment', *The International Journal of Advanced Manufacturing Technology* **103** (2019) 499-510, https://doi.org/10.1007/s00170-019-03557-w.

24. Malhotra, P.; Vig, L.; Shroff, G.; Agarwal, P., 'Long short term memory networks for anomaly detection in time series', in *Proceedings of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN*, i6doc.com publ., Bruges (2015) 89-94, https://www.esann.org/proceedings/2015 (accessed 2023-07-03).
25. Kim, D.; Antariksa, G.; Handayani, M. P.; Lee, S.; Lee, J., 'Explainable anomaly detection framework for maritime main engine sensor data', *Sensors* **21**(15) (2021) 5200, https://doi.org/10.3390/s21155200.
26. Kim, J-M.; Baik, J., 'Anomaly detection in sensor data', *Journal of Applied Reliability* **18**(1) (2018) 20-32, https://koreascience.kr/article/JAKO201815565837435.pdf (accessed 2023-07-03).
27. Gokhale, M. Y.; Khanduja, D. K., 'Time domain signal analysis using wavelet packet decomposition approach', *International Journal of Communications, Network and System Sciences* **3**(3) (2010) 321-329, https://doi.org/10.4236/ijcns.2010.33041.
28. Yu, Y.; Si, X.; Hu, C.; Zhang, J., 'A review of recurrent neural networks: LSTM cells and network architectures', *Neural computation* **31**(7) (2019) 1235-1270, https://doi.org/10.1162/neco_a_01199.
29. Shensa, M. J., 'The discrete wavelet transform: wedding the a trous and Mallat algorithms', *IEEE Transactions on Signal Processing* **40**(10) (1992) 2464-2482, https://doi.org/10.1109/78.157290.
30. Van der Maaten, L.; Hinton, G., 'Visualizing data using t-SNE', *Journal of Machine Learning Research* **9**(11) (2008) 2579-2605, https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf (accessed 2023-07-05).
31. 'Shankarpandala - lazypredict', in *Github*, https://github.com/shankarpandala/lazypredict (accessed 2023-07-03).

RECEIVED: 2023.1.30
REVISED: 2023.2.22
ACCEPTED: 2023.5.19
ONLINE: 2023.9.22