Brazilian Journal of Geophysics 43 (2025), 1, 1-19
2025 Brazilian Geophysical Society
ISSN 2764-8044
DOI: 10.22564/brjg.v43i1.2334

STRATEGIES FOR AUTOMATED CLASSIFICATION OF SEISMIC TRACE MEASURED BY OCEAN BOTTOM SEISMIC ACQUISITION SYSTEMS

Lucas Costa Lobato*, Nicolas Giovanardi, Josafat Ribeiro, and Stephan Paul

Universidade Federal de Santa Catarina - UFSC, Acoustic and Vibration Laboratory, Florianópolis, SC, Brazil

*Corresponding author email: lucascostalobato@gmail.com

ABSTRACT. Ocean bottom nodes (OBNs) are a recent technological solution used for seismic data acquisition. Despite of various advantages compared to conventional methods of measurement, the amount of data acquired in OBNs campaigns poses challenges to energy management and data transmission, ultimately limiting the time the device can acquire data on the seabed. To deal with these disadvantages, compression techniques and prediction models have been proposed in the literature and in both approaches the type of trace is an important information. In this work, strategies for developing seismic trace classifier models are assessed aiming to classify seismic traces from ocean bottom nodes into active, passive and microseism. The models were developed based on the machine learning algorithms decision tree and neural networks. Moreover, different features were used in the training process in order to analyze physical quantity dependent and agnostic classifier models. Five different datasets and thousands of traces were used for training and testing the models developed. Models outputs are explored in terms of confusion matrix, accuracy, precision and recall. Results have shown that the use of acceleration and velocity data for classification of microseism and passive traces led to a lower accuracy when compared to the use of sound pressure data. In addition, no relevant difference was found between the decision tree and neural networks for the classification task.

Keywords: machine learning in geophysics; machine learning models; neural networks for seismic analysis; decision tree classifier; seismic data compression.

INTRODUCTION

In the hydrocarbon industry, seismic data acquisition is the most widely used and well-known geophysical technique (Kearey et al., 2013). According to Kearey et al. (2013), the seismic reflection method, which generally consists in emitting an acoustic pulse and measuring the reflected sound and vibration, can reveal details about the geological structures from tens of meters up to the whole lithosphere. In underwater scenarios, the reflected sound can be measured by hydrophone systems, called streamers, at the surface of the sea, while sensors at the seabed can acquire both sound and vibration data. Ocean bottom based sensors systems such as Ocean Bottom Cables (OBCs), Permanent Reservoir Monitoring (PRM) systems and Ocean Bottom Nodes (OBNs) for seismic data acquisition present several advantages when compared to conventional streamer systems, such as measuring wider azimuth, possibility to characterize P and S waves, by means of sound pressure and acceleration or velocity, respectively, be less sensitive to noise produced by adjacent vessels and better repeatability (Hays et al., 2008; Zhang et al., 2021). While OBCs and PRM systems use sensors that are connected by cables to a hub that provides energy and allows data transfer, OBNs are autonomous sensor suites that rely on a battery

for its energy requirements and need to be collected to get access to the data collected by its integrated sensors.

However, for OBNs, the amount of data acquired limits the energy autonomy, the time of the OBNs residence on the seabed and hinders the communication among nodes and between nodes and the surface. Consequently, many recent publications aimed to assess different techniques to compress seismic data using both classical compress techniques and machine learning predictive models (Nuha et al., 2020, 2021; Helal et al., 2021; Lajús and Paul, 2023).

The time-continuous seismic data measured are normally segmented, being each segment called *trace* (Scales, 1995). Seismic traces can be classified into active, passive or passive microseismic ones. We consider active traces, in the context of seismic acquisition, to be traces that contain the seabeds and watercolumns response to active excitation by an acoustic source, generally an air gun. On the other hand, passive traces only contain background noise, while microseismic traces contain both background noise and the response of the watercolumn and seabed to excitations from microseisms (Artman, 2006). These different types of traces can present very different characteristics in terms of stationarity, amplitude, spectral content, etc. As an example, Figure 1 shows active and passive traces in terms of sound pressure and acceleration, measured on the seabed. In Figure 1, the differences between active and passive traces are graphically evident. It should be noted that active traces are composed by passive portions, before the arrival of the first wavefront from the air gun, and by an active portion after the arrival of the first wavefront, as can be seen in Figure 1.

As a consequence of these differences illustrated by the traces shown in Figure 1, it is less likely to obtain compression techniques and specially seismic data prediction models that are generic enough to be effective and accurate for both passive and active seismic data. The same reasoning can be extended to microseism traces. On the other hand, if there are differences that can be identified consistently, compression techniques and predictive models of seismic data can be optimized according to the physical quantity described by the trace. In this context, an autonomous seismic trace classifier is necessary. Such model is considered herein a *trace classifier model*.

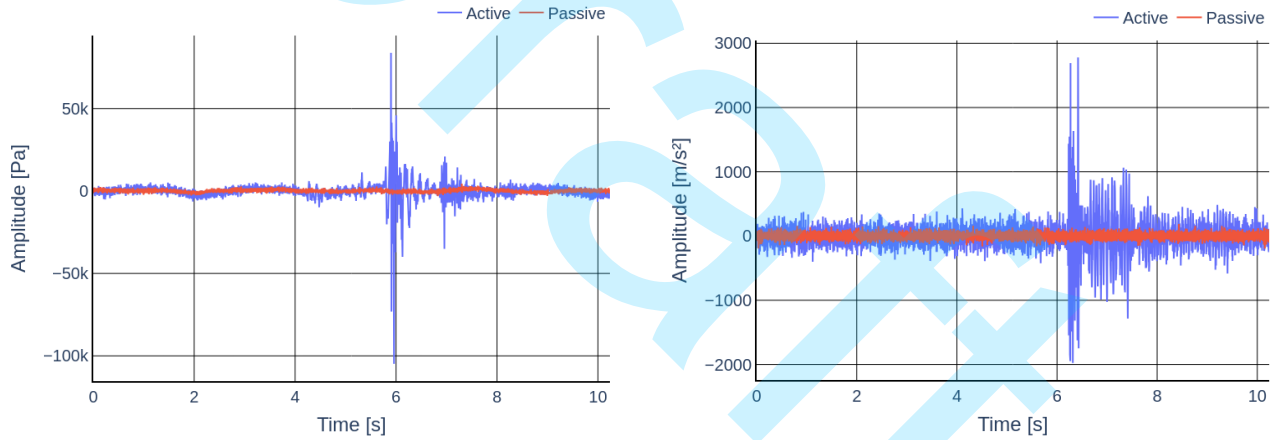
Studies reported in the literature have been focused on the detection of earthquakes (Li et al., 2018; White et al., 2023), volcano-seismic data (Bueno et al., 2020) and also the classification of wave dip in seismic reflection data (Geng and Wang, 2020). However, to our best knowledge, no work reporting on an autonomous classifier of seismic data (traces) into active, passive and passive with microseism (called henceforward as "microseism" only) traces was found in the literature. Therefore, this work aims to develop an approach to classify seismic traces into *active*, *passive* or *microseismic* data. It should be noted that the task of trace classification can be trivial for geophysicists. However, the goal of this work is to build a trace classifier model for seismic data measured by ocean bottom seismic acquisition systems in general (OBNs, OBCs and PRM). Consequently, not only sound pressure traces are considered but also acceleration and velocity (S-wave) traces should be classified. To achieve this goal, two consolidated machine learning algorithms are used and different features are tested. In Section Methods the experimental seismic data used to develop and test the trace classifier models are described as well as the feature engineering process and machine learning algorithms. Section Results presents the results obtained while Section Final Remarks addresses a discussion on the results, models applicability, future works and the final remarks.

METHODS

This section details the methods used to develop the different trace classifier models that will be evaluated. Firstly, Section Seismic datasets describes the experimental datasets used in the training and testing procedures. Then, in Section Classification algorithms the two machine learning algorithms are briefly described. In Section Feature engineering, the features used in the classification models and the reason to use normalized features are presented. Finally, Sections Training and testings procedures and Metrics of performance define the training and testing procedures used to develop the trace classifier models and the metrics used to evaluate the models, respectively.

Seismic datasets

Five datasets were used in this work: set A (active), set B (passive), set C (microseism), set D (active) and set E (passive). All datasets were kindly provided by Petrobras S.A in .SEG-Y format and were acquired by a PRM system, between 2013 and 2014, at Jubarte field (Thedy et al., 2014). The datasets are composed by the raw measurement data, without any signal preprocessing. Figure 1 shows examples of passive and active seismic traces in terms of sound pressure and acceleration. For the following analyses, only the dead traces, i.e., the traces containing only zeros due to some error during the acquisition, were removed previously.



(a) Sound pressure

(b) Acceleration

Figure 1: Active and passive seismic traces in terms of (a) sound pressure and (b) acceleration.

Datasets A and B were used in the training and testing procedures. Datasets D and E were used in a second test procedure (better described in Section Training and testings procedures), using the models already developed, in order to assess the generalization of the built models. Since dataset C is the only one with microseims, this dataset was used in model training and testing procedures as well as in the second test procedure. It should be noted that all datasets were previously classified by geophysicists into active, passive and microseism and these labels are used as reference in the training and test steps. The classification of an active event can be considered a trivial task since the traces from active and non-active seismic events are quite different. However, the difference between pure passive or passive with microseism traces is not so clear. Therefore, there is a probability of mislabeling of passive and passive with microseism traces. Nevertheless, The error rate in trace labeling is considered low because the data was labeled based on events detected by experienced geophysi-

cists. Given their expertise, microseisms in the time-space domain are easily visualized, reducing the likelihood of incorrect labels in the provided dataset.

And while the Jubarte field's datasets in terms of sound pressure and three-dimensional acceleration (Thedy et al., 2014) were obtained using a permanent reservoir system, our study aims to create a classifier for ocean bottom seismic acquisition systems in general, so traces in terms of particle displacement velocity must also be taken into account. Therefore, the acceleration traces used in the training and testing processes were converted into velocity to form a more complete dataset including sound pressure, acceleration, and velocity traces. This transformation was achieved through numerical integration using the `cumsum()` method from the Numpy library s1.23.5 (Numpy, 2024), to perform comulative summation, along with a high-pass filter with a cutoff frequency of 0.1 Hz. Figure 2 shows an example of an active trace converted from acceleration to velocity, with and without filtering. Additionally, a close-up of the active segment (following the arrival of the first wavefront from the air gun) is provided. This figure emphasizes the significance of filtering and demonstrates that the conversion yields reliable results, as evidenced by the consistent behavior of the first wavefront arrival and reflections.

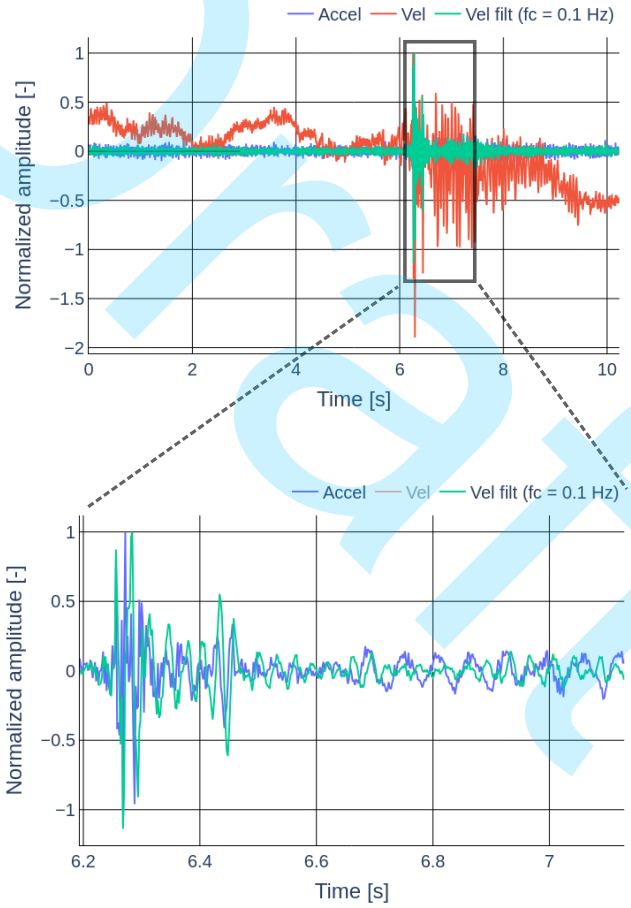


Figure 2: Example of an acceleration to velocity conversion, with and without filtering.

Classification algorithms

In this study, two distinct machine learning algorithms were used to develop classification models and evaluate their performance according to the algorithm employed. Certainly, there is a large number of machine learning techniques and algorithms that could be used to build a classifier model (Saravanan and Sujatha, 2018). Here,

a *decision tree* and a *neural network* techniques were chosen for the investigation and they are briefly described. All implementations were carried out using Python 3.10.

Decision tree (DT) is a supervised machine learning approach and was firstly presented by Quinlan (1986). It provides a hierarchical partition of the training data space Saravanan and Sujatha (2018). In this work, the class model `ExtraTreesClassifier()` was used (Scikit-Learn, 2023a). The `ExtraTreesClassifier()` model is an ensemble of randomized decision trees and we have used the implementation available in the open source library Scikit-learn s1.3.2. While the `ExtraTreesClassifier()` class has several parameterizable inputs (also known as hyperparameters), in our work the default values were used because we understand that the optimization of the hyperparameters requires a dedicated work. Thus, the number of trees was 100, and the Gini criterion was used for the trees' partition. For details, please see (Scikit-Learn, 2023a).

Neural Networks (NNs) are defined by mathematical models motivated by the behavior of neurons of the brain, which are the fundamental entity of a NN model (Chollet, 2017). The idea of a neuron model is to summarize an input vector $\mathbf{x}$ by the transformation applied by an activation function ϕ on the linear combinations of the inputs given by

$$y = \phi(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b), \tag{1}$$

being $\mathbf{w}$ and b the weights and biases of a given NN layer. Thus, the NN algorithm aims to obtain the weights $\mathbf{w}$ and biases b for using Equation 1 to achieve the desired output y. In this case, the input $\mathbf{x}$ vector is the set of features computed from a single seismic trace (please see Section Feature engineering) while the output y is a vector of the transformed input. Then, the number of neuron defines the size of the output. In our model, an output vector size of three is desired, wherein each output value expresses the probability of the input to belong to a given class (active, passive or microseism).

In this work the library Keras s2.15.0, with the class model `Sequential()` (Keras, 2023) was used. A feed-forward NN was built with four layers, being the number of neurons and activation function of each layer presented in Table 1. The input dimension of the first layer depends on the number of features used as input, as detailed in Section Feature engineering. Moreover, the Adam optimizer with learning rate of 0.001 and the *categorical_ crossentropy* loss function were used in the training process. The ReLU activation function was used due to the nature of the input, i.e., real non-negative values. The softmax activation function was used in the last layer to obtain an output that describes the probability of the input to belong a given class. Finally, the categorical_crossentropy loss function was selected once it is suitable for multi-class classification.

The number of epochs was defined by a convergence analysis, as shown in Section Number of epochs of the Neural Networks models.

Table 1: Description of the neural network model.

ine **Layer**	**Number of neurons**	**Activation function**	**Input dimension**
ine 1	500	ReLU	Number of features used
ine 2	100	ReLU	-
ine 3	50	ReLU	-
ine 4	3	Softmax	-
ine			

It should be noted that, unlike the DT, models based on NN algorithms may not be discrete classifiers, depending on the activation function. Since the ReLu activation function is used, the NN algorithm yields a discrete classifier. In other words, the results are presented as scores indicating the prediction probabilities for each class. In this study, the NN results were discretized by selecting the class with the highest score as the discrete prediction. This was done to facilitate comparison between the results of DT and NN models.

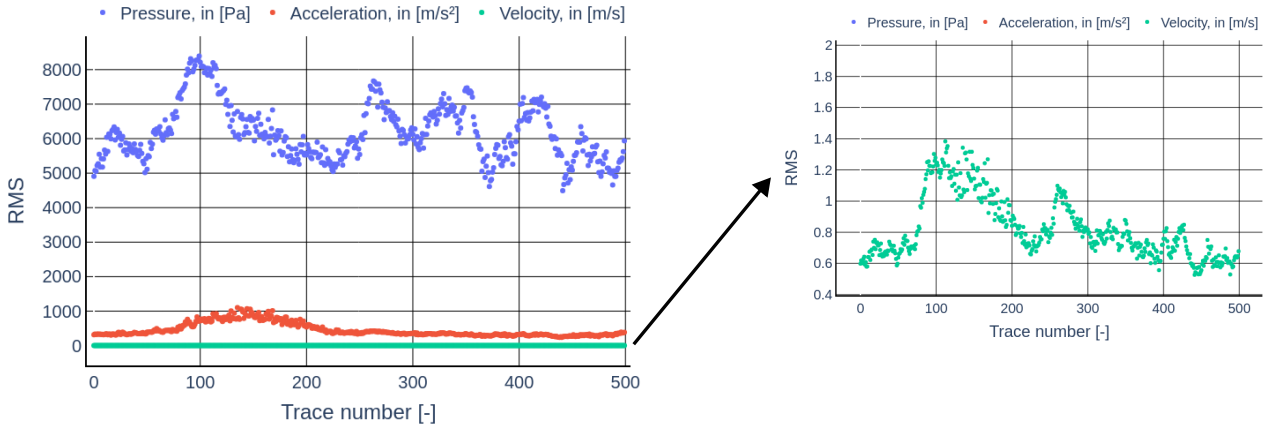
Feature engineering

Ocean bottom seismometers typically collect 4C data, including sound pressure and three components of acceleration (or velocity), like the seismic data utilized in this study (Thedy et al., 2014). Furthermore, this study aims to introduce a trace classifier model for ocean bottom seismometers in general, meaning that sound pressure, acceleration, and velocity should all be taken into account. However, it is important to note that the sound pressure, acceleration, and velocity traces vary significantly in magnitude, as illustrated in Figure 3. Therefore, in order to create a model that can be applied to all three physical quantities, the features were normalized based on the maximum value (also known as peak value) of each trace, resulting in normalized features. Figure 3 displays both the absolute and normalized features of RMS (root mean square) and RMS divided by the maximum value of each trace from a dataset of active traces. The large differences in sound pressure, acceleration, and velocity regarding their RMS magnitudes are clearly observed in Figure 3. Additionally, it is evident that the normalized features have the potential to be utilized in the development of a model that is independent of the physical quantity, capable of handling all types quantities simultaneously. Furthermore, we believe that classifiers based on normalized features tend to be less sensitive to different amplitudes measured by ocean bottom seismometers caused by changes in the propagation medium, different distances between source and receiver or different sources that can be used.

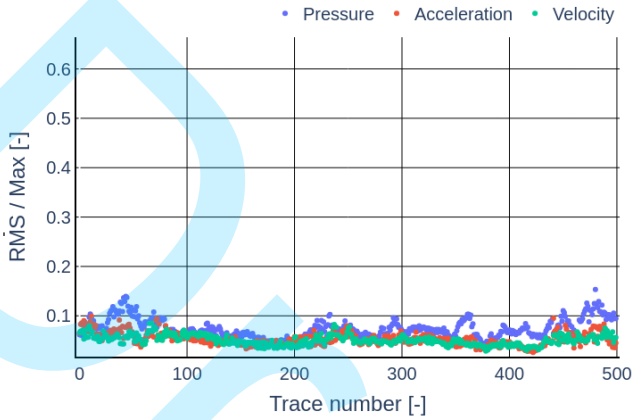
Five features are examined in this research, all calculated in the time domain and normalized by the maximum value of each trace: RMS (rms/max), percentile 80% (perc80/max), standard deviation (std/max), skewness (skew/max) and kurtosis (kurt/max). It should be noted that the features were calculated separately for each physical quantity (sound pressure, acceleration and velocity), being therefore independent from each other.

Figure 4 shows a contour plot using Kernel Density Estimate (KDE) of the five features, illustrating their relationship. The figure includes 700 traces of sound pressure from datasets A, B and C. In addition, a probability density function also using KDE for each feature is shown in the main diagonal of Figure 4. It can be seen that the active traces differ very well from the passive and microseism for rms/max, perc80/max, std/max and kurt/max features. For these four features, the passive and microseism traces are partially overlapped. On the other hand, for the skew/max feature the active, passive and microseism traces are overlapped. Moreover, the sub scatter plot correlating rms/max and std/max reveals an almost linear relationship between the two features, indicating redundancy in utilizing both features. The relationship among features, presented by the scatter plot in the off-diagonal of Figure 4 do not clearly show a potential of properly clustering the passive and microseism traces.

Figure 5 shows a contour plot of the same five features shown in Figure 4, but for one component of the acceleration traces. Contrary to the features observed for the sound pressure traces, the features of the acceleration traces show the microseism traces overlap partially the active and almost totally the passive ones.



(a) Absolute feature



(b) normalized feature

Figure 3: Absolute (a) and normalized (b) feature RMS from a dataset of active traces.

Additionally, note that for kurt/max feature, the passive data is too concentrated so that the active and microseism data are hidden. Figures 4 and 5 suggest the microseism and passive traces are better differentiated by features based on sound pressure traces than by those features based on acceleration traces. As a consequence of the observations from Figures 4 and 5, beyond the two different machine learning algorithms, two sets of features and two set of physical quantities are employed in the training phase, as elaborated in the subsequent section.

Training and testings procedures

The models were trained using two sets of features: the first, called s1, considers only rms/max and the second, called s2, considers the five features described in Section Feature engineering. Also, the models were trained considering only the sound pressure traces, identified by the superscript ^{p}, and also considering sound pressure, acceleration and velocity, identified by the superscript ^{all}. It should be noted that, for models with the superscript ^{all}, one shot is classified seven times: once using its sound pressure measurement, three times using its three-component acceleration measurements, and three times using the velocity integrated from the acceleration measurements. The distinction between models trained only with sound pressure or trained with sound pressure, acceleration, and velocity was made to analyze if the classification based on different physical quantities impacts the accuracy of the model.

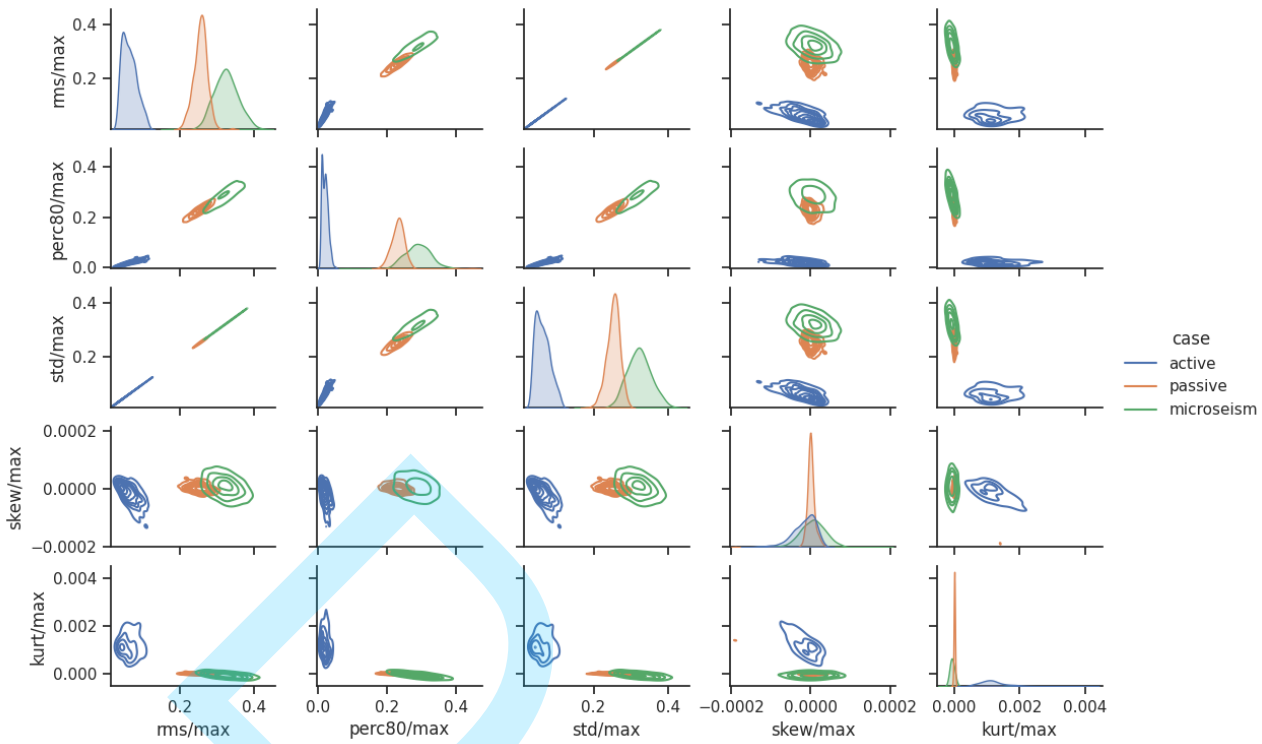


Figure 4: Contour plot of the normalized features considered in this study and computed using the sound pressure traces from datasets A, B and C.

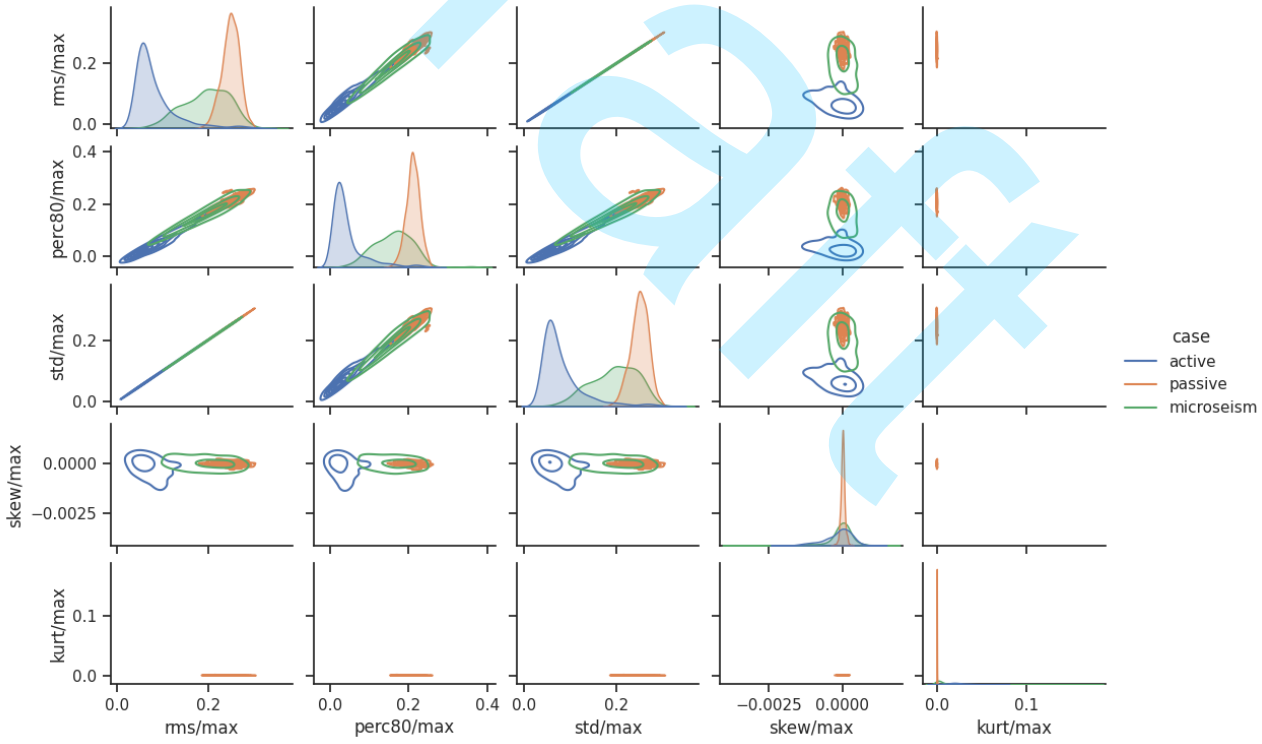


Figure 5: Contour plot of the normalized features considered in this study and computed using one component of the acceleration traces from datasets A, B and C.

In the validation process, the models based only on sound pressure traces have been trained with 500 traces of sound pressure from datasets A, B, and C, composing 1500 traces. The models based on all physical

Table 2: Summary of every model assessed, being ML machine learning and "all" the features described in Section Feature engineering. The number of traces is related to the total traces used in training and testing procedures. In this study, the number of traces was split 50% for training and 50% for testing.

ine ine Model's name	ML algorithm	Features	Physical quantities	Number of traces
ine DT $s1^{p}$	Decision tree	rms/max	sound pressure	1500
ine DT $s1^{all}$	Decision tree	rms/max	sound pressure, acceleration, velocity	10500
ine DT $s2^{p}$	Decision tree	all	sound pressure	1500
ine DT $s2^{all}$	Decision tree	all	sound pressure, acceleration, velocity	10500
ine NN $s1^{p}$	Neural network	rms/max	sound pressure	1500
ine NN $s1^{all}$	Neural network	rms/max	sound pressure, acceleration, velocity	10500
ine NN $s2^{p}$	Neural network	all	sound pressure	1500
ine NN $s2^{all}$	Neural network	all	sound pressure, acceleration, velocity	10500
ine ine				

quantities have been trained with 1500 traces of acceleration (500 for each Cartesian component), the velocity traces converted from these 1500 acceleration traces, and 500 sound pressure traces from datasets A, B, and C, composing 10500 traces. Table 2 summarizes all models evaluated in this study. Once the features were computed for these 10500 traces, the set of traces was randomly split using the `train_test_split()` (Scikit-Learn, 2023b) function from Scikit-learn s1.3.2 library. In this function, 50% of the datasets are defined for training and 50% for testing because we aimed to evaluate the model during the testing process with as many traces as possible. This test using the same dataset for training (but with different traces) is called here validation tests.

After the validation procedure, another test was applied to verify the models' generalization. This procedure is called here a generalization test. Thus, datasets C, D and E were used. Due to the lack of alternative microseismic datasets, dataset C was used for validation and generalization testing, but different trace subsets were used for each purpose. In the generalization test, 10500 traces were used, being 1500 acceleration traces (500 for each Cartesian component), 1500 velocity traces converted from these 1500 acceleration traces, and 500 sound pressure traces, similar to what was described above, but now from datasets C, D, and E. In summary, the classifiers were trained with 5250 traces and tested against 5250 traces in the validation test and against 10500 traces in the generalization test.

Metrics of performance

To evaluate the performance of the trace classifier models developed, the results are presented through confusion matrices, which show the relationship between the occurrences of the references and the predictions (Grandini et al., 2020). From a confusion matrix, various metrics can be calculated, and three of them are utilized to assess the multi-class classification models in this study: overall accuracy, precision, and recall (Grandini et al., 2020). Additionally, the Receiver Operating Characteristic (ROC) curve for each model is generated. All three metrics, along with the ROC curve, are based on the classifications of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP and TN indicate correct classification by the model, while FN

and FP represent incorrect classification.

The general accuracy, given by

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{2}$$

relates the sum of TP and TN to the total number of samples. Accuracy is a general measure of the model. On the other hand, precision and recall are measures of the model's performance related to a specific class. Therefore, for a specific class, precision is given as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{3}$$

and recall is given as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{4}$$

Precision is the percentage of samples classified as class x that truly belong to class x, while recall is the percentage of samples that truly belong to x that were classified as class x. Therefore, both precision and recall evaluate a model's performance from different points of view.

Additionally to the aforementioned measures the analysis of the ROC curve is a useful method to evaluate a classifier's performance (Fawcett, 2006). The ROC displays the TP rate (TPR) at the y-axis, which has the same definition of recall (see Equation 4), over the FP rate (FPR) at the x-axis, given as

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{5}$$

Since the ROC graph deals with binary classification scenarios, and in this work a multi-class scenario is considered, the results from the models developed here were binarized using the one-vs-rest approach. In other words, the ROC graphs are built with Recall and FPR of one class with relation to the others.

RESULTS

The results are organized in two sections: firstly, the results from the validation and cross-validation using k-fold tests are presented in Section Validation tests. Then, the best two models identified in the validation test are used in the generalization test and results are presented in Section Generalization tests.

Number of epochs of the Neural Networks models

The training procedure of the models based on NN algorithms depends on the number of epochs defined. In order to define the suitable number of epochs, a convergence analysis was performed. Figure 6 shows the convergence of the accuracy and the time required for the fitting process of model NN s2$^{\text{p}}$ for nine different numbers of epochs used in the training procedure. Results show that while the accuracy converges around 80 epochs, the fitting time increases almost linearly with the number of epochs. Note that only the fitting time was assessed since the prediction time and model size are not affected by the number of epochs used in the training

process. Therefore, for the remaining analyses presented in this study using NN models, 80 epochs were used in the training process.

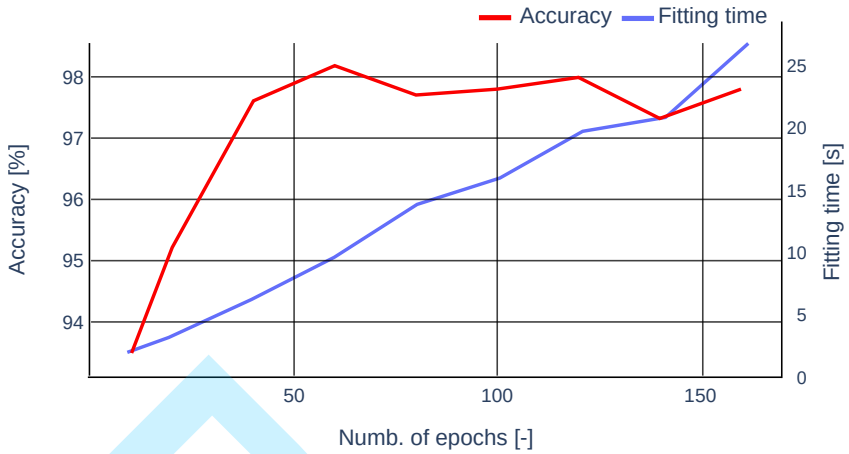


Figure 6: Convergence of the accuracy and fitting time of model NN $s2^{p}$ as a function of epochs used in the training procedure.

Validation tests

Figure 7 shows the confusion matrices of the results obtained from the validation test of the eight classifier models described in Table 2. In addition, Table 3 presents the accuracy, recall and precision of the models.

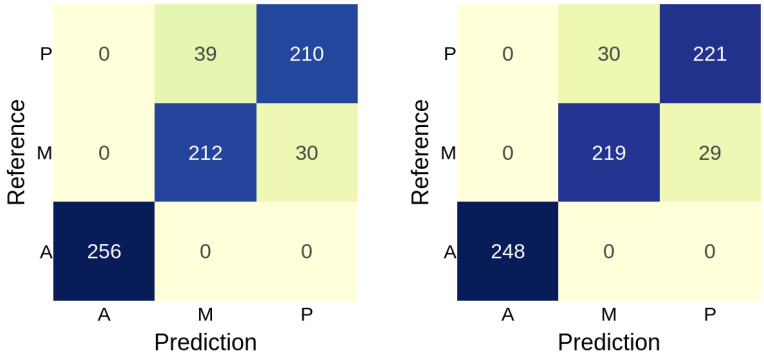
Decision tree vs Neural Networks

At first, the results are compared in terms of the machine learning algorithm used. Results presented in Figure 7 and Table 3 show that the use of Decision tree or Neural Network algorithms does not seem to impact the performance of the classifier. Comparing the accuracy of models based on DT and NN while varying the set of features and physical quantities considered, no considerable differences could be observed. For instance, the models based on both machine learning algorithms presented great performance in classifying active traces correctly but resulted in misclassifications between microseism and passive traces. However, it should be noted that the hyperparameters of the algorithms were not optimized. In other words, if the hyperparameters are adjusted, differences among the models based on different algorithms might be found.

One feature (v1) vs all features (v2)

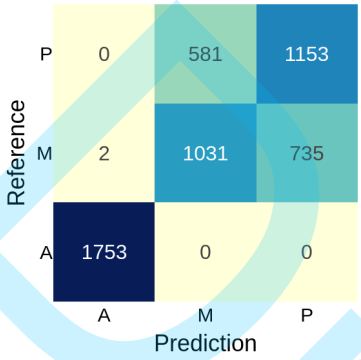
Results presented in Figure 7 and Table 3 show that the performance of the model for classifying the active traces was not affected by the set of features tested in this study. For the active traces, recall was 100% for all models while the precision was equal or higher than 99.8% for all models. However, for the classification of passive and microseism traces, the set of features used presented a considerable impact.

When only the feature rms/max is used (set s1), models were unable to separate microseism and passive traces. This is evidenced by recall and precision for this classification attempt being lower or equal than 88.4% for all models. For instance, recall for passive trace classification by model NN $s1^{all}$ is only 34.2%. On the other hand, using all five features proposed (set s2), misclassification between the microseism and passive traces

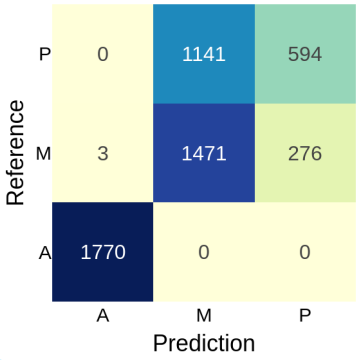


(a) DT, $s1^{P}$

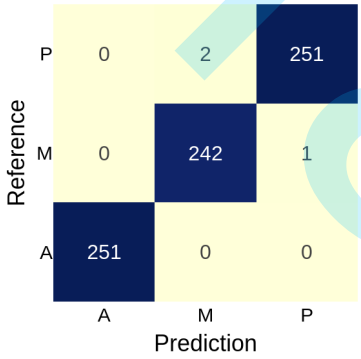
(b) NN, $s1^{P}$



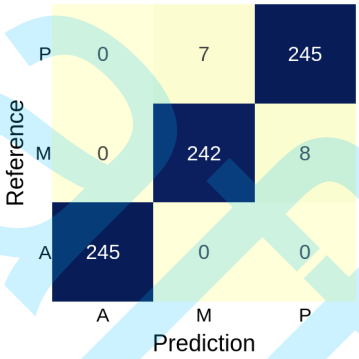
(c) DT, $s1^{all}$



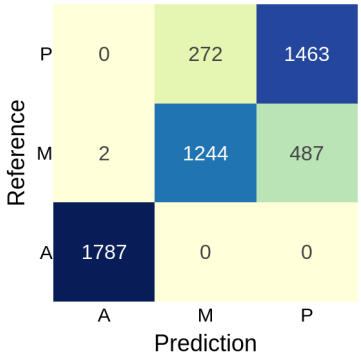
(d) NN, $s1^{all}$



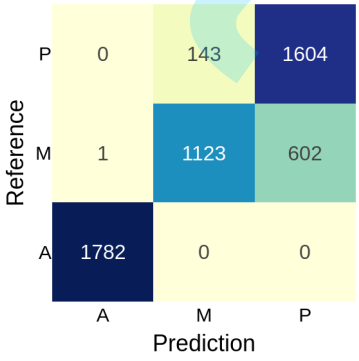
(e) DT, $s2^{P}$



(f) NN, $s2^{P}$



(g) DT, $s2^{all}$



(h) NN, $s2^{all}$

Figure 7: Confusion matrices of the results obtained from the validation tests of the eight trace classifier models described in Table 2, being: A - active; M - microseism; and P - passive.

was reduced. For instance, models based only on sound pressure traces, such as DT $s2^{p}$ and NN $s2^{p}$, showed precision and recall for the microseism and passive trace classification problem in excess of 96.8%.

For a better visualization of the impact of the set of features, results from models based on all physical quantities are shown by means of a ROC graph, in Figure 8. The ROC graph displays points corresponding to models DT $s1^{all}$ and NN $s1^{all}$, with the microseism and passive classes (blue/red square/diamond markers) being in close proximity to the dashed line, that corresponds to random classification (Fawcett, 2006). In the case of model DT $s1^{all}$, the misclassifications between the microseism and passive classes appear more random compared to model NN $s1^{all}$, as the ROC points are nearer to the (0.5, 0.5) coordinate. However, there was an improvement in the classification of microseism and passive traces when all features are used. The ROC graph illustrates that by the points associated with DT and NN $s2^{all}$ (x and hourglass markers) for the microseism and passive classes (blue and red markers) are closer to the (0, 1) coordinate when compared to DT and NN $s1^{all}$.

Table 3: Summary of the accuracy, class, recall and precision from all models in the validation test.

ine ine **Model**	**Accuracy**	**Class**	**Recall**	**Precision**
ine		Active	100%	100%
DT, $s1^{p}$	90.8%	Microseism	87.6%	84.4%
		Passive	84.3%	87.5%
ine		Active	100%	99.9%
DT, $s1^{all}$	74.9%	Microseism	58.3%	63.5%
		Passive	66.5%	61.1%
ine		Active	100%	100%
DT, $s2^{p}$	99.6%	Microseism	99.5%	99.2%
		Passive	99.2%	99.6%
ine		Active	100%	99.8%
DT, $s2^{all}$	85.5%	Microseism	71.8%	82.1%
		Passive	84.3%	75.0%
ine		Active	100%	99.8%
NN, $s1^{p}$	92.1%	Microseism	88.3%	87.9%
		Passive	88.0%	88.4%
ine		Active	100%	99.8%
NN, $s1^{all}$	73.0%	Microseism	84.1%	56.3%
		Passive	34.2%	68.3%
ine		Active	100%	100%
NN, $s2^{p}$	98.0%	Microseism	96.8%	97.2%
		Passive	97.2%	96.8%
ine		Active	100%	99.9%
NN, $s2^{all}$	85.8%	Microseism	65.1%	88.7%
		Passive	91.8%	72.7%

ine ine

Sound pressure vs all physical quantities

Regarding the classification of the active traces, the physical quantity of the trace does not affect the performance of the models in classifying these traces. However, for the classification of passive and microseism traces, the physical quantities used for the trace classification showed to have a significant impact on the performance of the models.

Figure 7 and Table 3 show that misclassifications are more likely to occur when the classification is based on acceleration or velocity traces compared to the classification based on sound pressure traces. For instance, while model DT $s1^{p}$ presented recall and precision for microseism and passive traces greater than 84%, model DT $s1^{all}$ that seeks to classify traces of acceleration and velocity too, achieves recall and precision for microseism and passive not larger than 66.5%. Similarly, model NN $s1^{p}$ presented recall and precision for microseism and passive traces around 88% while NN $s1^{all}$ only achieved recall and precision equal to 34.2% and 68.3%, respectively, for passive traces. The same was observed for the model trained with set of features s2. These results indicate that the distinctions between microseism and passive traces are more noticeable in the sound pressure traces if compared to the acceleration and velocity.

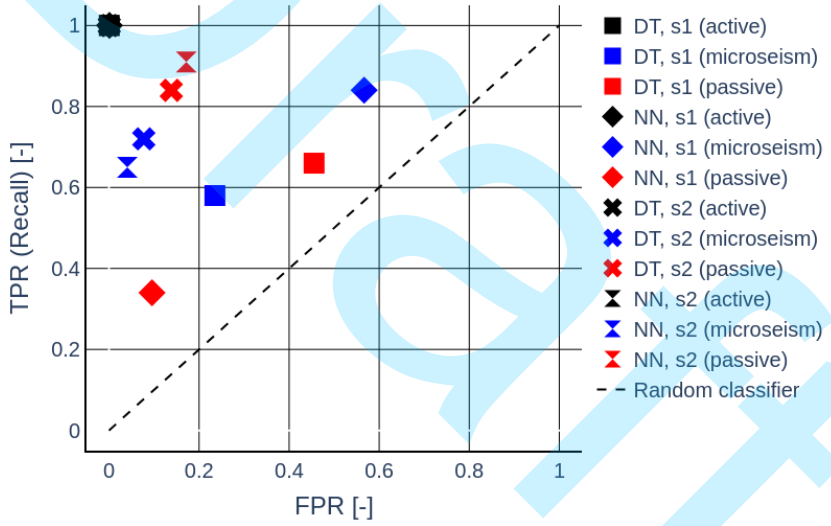


Figure 8: ROC graph from all models based on all physical quantities with relation to all one-vs-rest combinations from the validation test. The superscript "all" was omitted since it is common to all models presented in this figure. It should be noted that the same marker symbols are used for the same models and the same colors are used for the same classes.

Cross-validation using k-fold

In this section, a k-fold cross-validation approach (Wong, 2015) is applied to evaluate the performance of the best two classifier observed in previous sections. Here, 10 k-fold were used to split the training and testing data for models DT $s2^{p}$ and NN $s2^{p}$. Thus, the classifiers are trained and tested 10 times with different separation of training and testing data. This analysis allows us to verify the accuracy of the classifiers over different training and test data separation, avoiding a possible optimistic accuracy from only one training and test running (also known as leave-one-out cross-validation) (Wong, 2015). The results of the 10 k-fold cross-validation in terms

of accuracy's mean and standard deviation of the models DT $s2^p$ and NN $s2^p$ are shown in Table 4. The low standard deviations observed – 2.2% for DT $s2^p$ and 2.9% for NN $s2^p$ – indicate a very stable accuracy of the classifiers for the different folds of training data.

Table 4: Accuracy's mean and standard deviation of the models DT $s2^p$ and NN $s2^p$ obtained from 10 k-fold cross-validation.

ine ine **Model**	**Mean**	**Standard deviation**
ine DT $s2^p$	97.5%	2.2%
ine NN $s2^p$	95.5%	2.9%
ine ine		

Hyperparameters grid search

Section Validation tests showed that models DT $s2^p$ and NN $s2^p$ outperformed models DT $s2^{all}$ and NN $s2^{all}$. In other words, classification of acceleration and velocity traces is considerably less accurate than the classification of sound pressure traces. For this reason, this section presents a fine tuning approach of the hyperparameters of the models to strive for an optimal model for enhanced classification of acceleration and velocity traces. To this end, a grid search approach was applied to find a combination pf hyperparameters that provide the highest possible accuracy.

For the decision tree model DT $s2^{all}$ the following hyperparameter space was evaluated: number of trees = {20, 40, 60, 80, 100, 120, 140, 160, 180, 200}; criterion = {gini, entropy, logloss}; and maximum tree depth = {2, 5, 10, 20, 50}. On the other hand, for the neural network model NN $s2^{all}$ other three characteristics were varied: the activation function of the layers = {relu, elu, tanh, sigmoid}; the loss function used in the training process = {mean absolute error, poisson, categorical crossentropy}; and the architecture of the neural network (i.e., the number of layers and the number of neurons of each layer). The architectures tested were: {100, 50, 3} (#1 - three layers), {500, 100, 50, 3} (#2 - four layers), {600, 300, 150, 75, 25, 3} (#3 - six layers) and {1000, 750, 600, 500, 300, 250, 150, 50, 3} (#4 - nine layers), wherein each value refers to the number of neurons of each layer.

For models DT $s2^{all}$ and NN $s2^{all}$, all combinations in the hyperparameter space were tested. The accuracy was measured in each run. In the end, the best hyperparameters, i.e., the hyperparameters related to the highest accuracy were identified as well as the mean, standard deviation, max and min accuracy for all runs. Table 5 summarizes the statistics for the accuracies from the grid search fine tuning. Results show that the maximum accuracy observed for model DT $s2^{all}$ was only 0.6% above that observed in Section Validation tests. For model NN $s2^{all}$, the maximum accuracy was precisely the same as observed in Section Validation tests, with the hyperparameters being slightly different. In other words, although several combinations of hyperparameters were tested, the performance of the models could not be considerably improved when compared to the models validated in Section Validation tests.

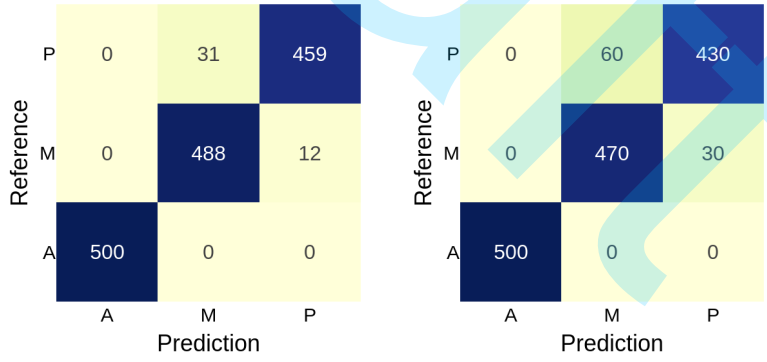
Table 5: Accuracy's mean, standard deviation, Max and Min of the models DT $s2^{all}$ and NN $s2^{all}$ obtained from the grid search fine tuning, being N_t the number of trees, M_d the maximum depth, C the criterion, loss the loss function, ϕ the activation function and arch the neural network architecture.

ine ine **Model**	**Mean**	**Standard deviation**	**Max**	**Min**	**Best hyperparameters**
ine DT $s2^{all}$	80.9%	5.2%	86.1%	69.8%	$N_t = 60$, $M_d = 2$, C = gini
ine NN $s2^{all}$	72.4%	16.6%	85.8%	33.3%	loss = categorical crossentropy, ϕ = elu, arch = #3
ine ine					

Generalization tests

The results of the validation tests presented in Section Validation tests have shown that models using the set of features s2 outperformed those using the set of features s1. In addition, models trained only with sound pressure traces outperformed those trained with sound pressure, acceleration and velocity traces. Therefore, models DT $s2^p$ and NN $s2^p$, the best performing models according to the validation tests, were further evaluated in a generalization test. Figure 9 shows the confusion matrices of the results obtained in this generalization test. In addition, Table 6 presents the metrics of performance of each model obtained in the generalization test.

It should be noted that, due to the lack of microseism data, the same dataset C used in the validation test was used in the generalization test. Therefore, the generalization of the model regarding microseism traces could not be analyzed properly. From the generalization test, in general, it was observed that the models' performance in classifying active and passive traces was not considerably reduced when different datasets are considered. For the active traces specifically, recall and precision were 100%, as the observed in the validation test. However, for the passive traces, it was observed that recall and precision were reduced, being the reduction a little more pronounced for model NN $s2^p$.



(a) DT $s2^p$ (b) NN $s2^p$

Figure 9: Confusion matrices of the results obtained in the generalization tests of the trace classifier models DT $s2^p$ and NN $s2^p$, being: A - active; M - microseism; and P - passive traces.

FINAL REMARKS

Although numerous studies have classified seismic data (Li et al., 2018; Bueno et al., 2020; White et al., 2023), to the best of our knowledge, no dedicated work on classifying ocean bottom seismometer data into active, passive, and microseism data has been found in the literature. In this study, eight models, utilizing Decision

Table 6: Summary of the accuracy, class, recall and precision from the two best models in the generalization test.

ine ine **Model**	**Accuracy**	**Class**	**Recall**	**Precision**
ine		active	100%	100%
DT s2^{p}	97.1%	microseism	97.6%	94.0%
		passive	93.7%	97.4%
ine		active	100%	100%
NN s2^{p}	94.0%	microseism	94.0%	88.7%
		passive	87.7%	93.5%

ine ine

Tree and Neural Networks machine learning algorithms with two different sets of features, were developed and compared. All models used normalized features, enabling classification of traces in terms of normalized sound pressure, acceleration and velocity. While the features analysis initially showed no clear distinction between active, passive, and microseism traces, results achieved by using ML models indicated that models utilizing feature set s2 (comprising five features) outperformed those with feature set s1 (containing only one feature). Also, the classification of microseism and passive traces based on the acceleration or velocity measurements led to a lower accuracy when compared to classifications based on sound pressure. The analysis of the features in terms of sound pressure and acceleration showed the sound pressure traces provide a better distinction between microseism and passive then the acceleration traces. Furthermore, there was no significant difference in the classification performance between the DT and NN algorithms developed herein.

While the classifier described in this work is intended to be used in a seismic data compression system, such seismic trace classifiers have several other applications. For example, an energy management system for ocean bottom seismometers might be created to activate recording only when active traces are identified. Furthermore, an autonomous communication system may want to transmit a specific type of trace. In all these scenarios, a trace classifier is essential, and the models discussed in this study can be beneficial.

Even though different models were developed and analyzed, and good results were achieved in this work, some further investigations are still needed. A critical point is the misclassification of passive into microseism data and vice versa when classifications are based on acceleration or velocity data. Additionally, although it was not possible to evaluate the generalization of the model regarding microseism traces, a good model generalization related to the microseism class is less likely. While the signature of active seismic data due to excitation of an air gun (the source commonly used in the seismic reflection method) is well defined, holding a low variability Landrø et al. (2013), data from microseismic excitation can vary considerably since the source is not controlled (Longuet-Higgins and Jeffreys, 1950). Therefore, future works should assess pre-processing techniques in order to achieve better classifications regarding passive and microseism data.

In addition to the possible improvements in the data pre-processing step, investigations can be carried out specially in terms of alternative architectures. Both decision tree and neural network algorithm allow several hyperparameters and architectures by means of number of trees, number of layers, number of neurons, optimizer, epochs used in the training process, activation functions and so on. Therefore, an optimization related to the

hyperparameters of the trace classifier based on neural networks will be considered in a near future.

ACKNOWLEDGMENTS

The authors would like to thank to Petrobras S.A. for the financial support.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

Artman, B., 2006, Imaging passive seismic data: Geophysics, **71**, SI177–SI187, doi: 10.1190/1.2209748.

Bueno, A., L. Zuccarello, A. Díaz-Moreno, J. Woollam, M. Titos, C. Benítez, I. Álvarez, J. Prudencio, and S. De Angelis, 2020, Picoss: Python interface for the classification of seismic signals: Computers Geosciences, **142**, 104531, doi: https://doi.org/10.1016/j.cageo.2020.104531.

Chollet, F., 2017, Deep learning with python: Manning.

Fawcett, T., 2006, An introduction to roc analysis: Pattern Recognition Letters, **27**, 861–874, doi: https://doi.org/10.1016/j.patrec.2005.10.010. (ROC Analysis in Pattern Recognition).

Geng, Z., and Y. Wang, 2020, Automated design of a convolutional neural network with multi-scale filters for cost-efficient seismic data classification: Nature Communications, **11**, doi: https://doi.org/10.1038/s41467-020-17123-6.

Grandini, M., E. Bagli, and G. Visani, 2020, Metrics for multi-class classification: an overview: Presented at the . doi: 10.48550/arXiv.2008.05756.

Hays, D., K. Craft, P. Docherty, and F. Smit, 2008, *in* An Ocean Bottom Seismic node repeatability study: 55–59.

Helal, E. B., O. M. Saad, A. G. Hafez, Y. Chen, and G. M. Dousoky, 2021, Seismic data compression using deep learning: IEEE Access, **9**, 58161–58169, doi: 10.1109/ACCESS.2021.3073090.

Kearey, P., M. Brooks, and I. Hill, 2013, An introduction to geophysical exploration: Wiley.

Keras, 2023, The sequential class. (https://keras.io/api/models/sequential/ [Accessed: 2024-08-18]).

Lajús, F. C., and S. Paul, 2023, A near-lossless alternative for obn data compression through block-sorting and decimation: Presented at the 18 th International Congress of the Brazilian Geophysical Society.

Landrø, M., L. Amundsen, and J. Langhammer, 2013, Repeatability issues of high-frequency signals emitted by air-gun arrays: Geophysics, **78**, P19–P27, doi: 10.1190/geo2013-0142.1.

Li, W., N. Narvekar, N. Nakshatra, N. Raut, B. Sirkeci, and J. Gao, 2018, Seismic data classification using ma-

chine learning: 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService), 56–63. doi: 10.1109/BigDataService.2018.00017.

Longuet-Higgins, M. S., and H. Jeffreys, 1950, A theory of the origin of microseisms: Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences, **243**, 1–35, doi: 10.1098/rsta.1950.0012.

Nuha, H., B. Liu, M. Mohandes, A. Balghonaim, and F. Fekri, 2021, Seismic data modeling and compression using particle swarm optimization: Arab J Geosci, **14**, doi: 10.1007/s12517-021-08675-y.

Nuha, H. H., A. Balghonaim, B. Liu, M. Mohandes, M. Deriche, and F. Fekri, 2020, Deep neural networks with extreme learning machine for seismic data compression: Arab J Sci Eng, **45**, 1367–1377, doi: 10.1007/s13369-019-03942-3.

Numpy, 2024, Cumsum. (https://numpy.org/doc/stable/reference/generated/numpy.cumsum.html [Accessed: 2024-08-18]).

Quinlan, J. R., 1986, Induction of decision trees: Machine Learning, **1**, 81–106, doi: 10.1007/BF00116251.

Saravanan, R., and P. Sujatha, 2018, A state of art techniques on machine learning algorithms: A perspective of supervised learning approaches in data classification: 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 945–949. doi: 10.1109/ICCONS.2018.8663155.

Scales, J., 1995, *in* Introduction to seismic migration: Springer Berlin Heidelberg, 1–22.

Scikit-Learn, 2023a, Extra tree classifier. (https://scikit-learn.org/stable/modules/generated/sklearn.tree.ExtraTreeClassifier.html [Accessed: 2024-08-18]).

Scikit-Learn, 2023b, Train test split. (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.htmll [Accessed: 2024-08-18]).

Thedy, E. A., R. Filho, W. L. Johann, S. N. Seth, S. C. Souza, and P. E. Murray, 2014, *in* Jubarte Permanent Reservoir Monitoring – Installation and First Results: 918–920.

White, M. C. A., K. Sharma, A. Li, T. K. S. Kumar, and and, 2023, Classifying seismograms using the fastmap algorithm and support-vector machines: Communications Engineering, **2**, doi: https://doi.org/10.1038/s44172-023-00099-8.

Wong, T.-T., 2015, Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation: Pattern Recognition, **48**, 2839–2846, doi: https://doi.org/10.1016/j.patcog.2015.03.009.

Zhang, D., C. Tsingas, A. A. Ghamdi, M. Huang, W. Jeong, K. K. Sliz, S. M. Aldeghaither, and S. A. Zahrani, 2021, A review of OBN processing: challenges and solutions: Journal of Geophysics and Engineering, **18**, 492–502, doi: 10.1093/jge/gxab024.

Lobato, L.: conceptualization, models development, data analysis and writing; **Leal J.:** data analysis and reviewing; **Giovanardi N.:** data analysis and reviewing; **Paul S.:** conceptualization, data analysis and reviewing.