 doi:10.20944/preprints202010.0048.v1

Article

Application of Machine learning for quality control of noise attenuation processes in seismic data imaging

Mohamed Mejri [1], Aymen Mejri [2] and Maiza Bekara [3]

[1] mohamed.mejri@gatech.edu
[2] aymen.mejri@telecom-paristech.fr
[3] maiza.bekara@pgs.com
* 1 - School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA,
2 - Image, Data and Signal department, Télécom ParisTech, Paris, France
3 - PGS Exploration Ltd, Weybridge KT13 0NY, United Kingdom

Abstract: Seismic imaging is the main technology used for subsurface hydrocarbon prospection. It provides an image of the subsurface using the same principles as ultrasound medical imaging. It is based on emitting a sound (pressure) wave through the subsurface and recording the reflected echoes using hydrophones (pressure sensors) and/or geophones (velocity/acceleration sensors). Contrary to medical imaging, which is done in real time, subsurface seismic imaging is an offline process that involves a huge volume of data and needs considerable computing power. The raw seismic data are heavily contaminated with noise and unwanted reflections that need to be removed before further processing. Therefore, the noise attenuation is done at an early stage and often while acquiring the data. Quality control (QC) is mandatory to give confidence in the denoising process and to ensure that a costly data re-acquisition is not needed. QC is done manually by humans and comprises a major portion of the cost of a typical seismic processing project. It is therefore advantageous to automate this process to improve cost and efficiency. Here, we propose a supervised learning approach to build an automatic QC system. The QC system is an attribute-based classifier that is trained to classify three types of filtering (mild = underfiltering, noise remaining in the data; optimal = good filtering; harsh = overfiltering, the signal is distorted). The attributes are computed from the data and represent geophysical and statistical measures of the quality of the filtering. The system is tested on a full-scale survey (9000 km^2) to QC the results of the swell noise attenuation process in marine seismic data. The results are encouraging and helped identify localized issues that were difficult for a human to spot.

Keywords: QC denoise automation; feature transformation techniques; classification methods

1. Introduction

For the oil and gas industry, localizing and characterizing economically worthwhile geological reservoirs is essential. Nevertheless, easy-to-exploit reservoirs are relatively rare, and target exploration techniques are becoming increasingly sophisticated. As a result, the requirements for high-quality seismic images in terms of both the signal-to-noise ratio (SNR) and the regularity and density of sampling are constantly increasing [1]. The seismic dataset is always contaminated by different types of noise and unwanted reflections; therefore, the noise attenuation is an important step in a typical seismic data processing sequence. As with every important step, performing quality control (QC) after each noise attenuation process is essential to ensure that noise has been sufficiently removed while no useful signal has been distorted. The QC process is time consuming and often requires precious human resources to perform a visual inspection of several seismic products, such as: (i) selected seismic gathers before and after filtering; (ii) 2D-3D stacks before and after filtering; and (iii) maps of generic seismic attributes (e.g., RMS amplitudes or coherency slices) computed before and after filtering. Geophysicists are becoming increasingly reliant on the assessment of global attribute maps that can give an overall picture of the performance of the filtering process; however, this picture is

 doi:10.20944/preprints202010.0048.v1

compressed and local problems can be easily missed. The simultaneous visual assessment of multiple attributes is difficult, empirical, and subjective. Improving the efficiency and the turnaround of seismic processing projects can be achieved by automating the QC process. Here, we investigate the use of several machine learning techniques to automate the quality control process of seismic data.

2. Context and related work

Early work on the use of statistical data mining for QC was reported by Spanos et al. [2], where statistical metrics or attributes computed from the seismic data are used in an unsupervised way to assess the quality of a filtering process. The assessment was binary (i.e., either the filtering is normal or it shows some anomalies and it was based on outlier detection). This methodology was an exploratory step towards the development of a data-driven automated QC platform that takes into account techniques from the fields of statistics and machine learning. The work of the authors in [3] further expands the initial work above and proposes the use of a supervised learning-based approach for training data to build an automatic classifier using a support vector machine (SVM) to predict the type of filtering (mild, optimal, or harsh).

2.1. Feature transformation and dimensionality reduction

Ensemble-based statistical attributes that measure the similarity between the filtered seismic image (output) and the residual image (output-input) are used to assess the quality of the denoising process. In the case of the optimal filtering, these attributes will not show any similarity (i.e., correlation) between the output and the difference. This assumption comes from the fact that the signal (i.e., output) has nothing in common with the noise (residual) in the case of ideal filtering. When there is signal distortion or residual noise, the level of similarity will increase as some noise is also present in the output or a signal is also present in the difference, and this will be picked up by the attributes. Authors in [3] proposed using principal component analysis (PCA) as a feature transformation technique to improve the separability of attributes, and hence, to ease the construction of the decision space. Although PCA decomposition allows the extraction of uncorrelated features, nonlinearity in the relationship between attributes and their high dimensionality are two potential limitations of this method. To overcome this issue, a new method for performing a nonlinear PCA was introduced in [4], which suggested computing principal components in high-dimensional feature space using integral operator kernel functions and was used for pattern recognition. Kernel-based PCAs, as well as linear PCAs, are unable to extract independent components since the data distribution along the attributes is not necessarily Gaussian. Independent component analysis (ICA) is a statistical technique that aims to find a linear projection of the data that maximizes an inter-dependency criterion [5].

2.2. Classification methods

The work proposed by the authors in [3] uses a support vector machine [6] with a polynomial kernel to predict the quality of the denoising process. Different approaches exist when it comes to using binary support vectors to perform multiclass prediction tasks. The work proposed in [7] described the one-against-all strategy as a set of one SVM per class trained to predict one class from a sample of all the remaining classes; the classification is performed according to the maximum output among all SVMs. In contrast, the one-against-one approach, also known as “pairwise coupling”, “all pairs”, or “round-robin”, consists of building one SVM for each pair of classes.The classification of one testing sample is usually done according to the maximum voting, where each SVM votes for one class. Although SVMs are efficient predictors, especially when it comes to binary classification, other models exist and they are more or less efficient depending on the classification task and the complexity of the training set. Random forest, introduced in [8], is one of them. According to [8], decision trees are very attractive for their high training speed; however, trees derived from traditional models cannot handle training sets with high complexity. In [8], authors built multiple trees in randomly selected subspaces of the feature space. Unlike SVMs, random forests are able to perform multiclass prediction tasks

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2020 doi:10.20944/preprints202010.0048.v1

without using multiple binary classifiers. Authors in [9] observed that the one-against-all random forest strategy achieves better classification performance than common random forest and is more robust to outliers.

Unlike all the classifiers already cited, the neural network architecture is very difficult to tune. Many autonomous deep neural network-based algorithms have been designed (e.g, SGNT [10], NADINE [11], ADL [12]) to manage the prediction of data streams. They aimed to automatically build a neural network architecture by performing growing and pruning operations for hidden layers and nodes, respectively, when it comes to underfitting and overfitting.

3. Methods

To automate quality control of the denoising process [3], ensemble-based (e.g., group of sensors) statistical attributes that measure the level of similarity between the output of the filtering and the residual (the difference between the output and the input) are used. In the case of optimal filtering, the residual consists only of the remaining noise after a filtering operation; therefore, these attributes will not show any similarity between the difference and the output (i.e., correlation). When signal leakage or residual noise is observed, the level of similarity will increase as some noise is also present in the output or some signal is also present in the difference, and this will be picked up by the attributes. The Pearson cross-correlation, Kendell cross-correlation, and mutual information are examples of attributes that capture the correlation between the filtering output and the difference [13]. The cross plots of five different attributes computed from three test lines are shown overlaid for the three types of filtering in Figure 1. A tricolor code is adopted for the display (mild = blue; optimal = green; and harsh = red). Some attributes (e.g., TLAMBD, in Figure 1) show a good level of visual separation between the different types of filtering, particularly the harsh one. The clusters of attributes for the mild and the optimal filtering are close as they reflect the observation made earlier about the subtle differences between the two types of filtering.

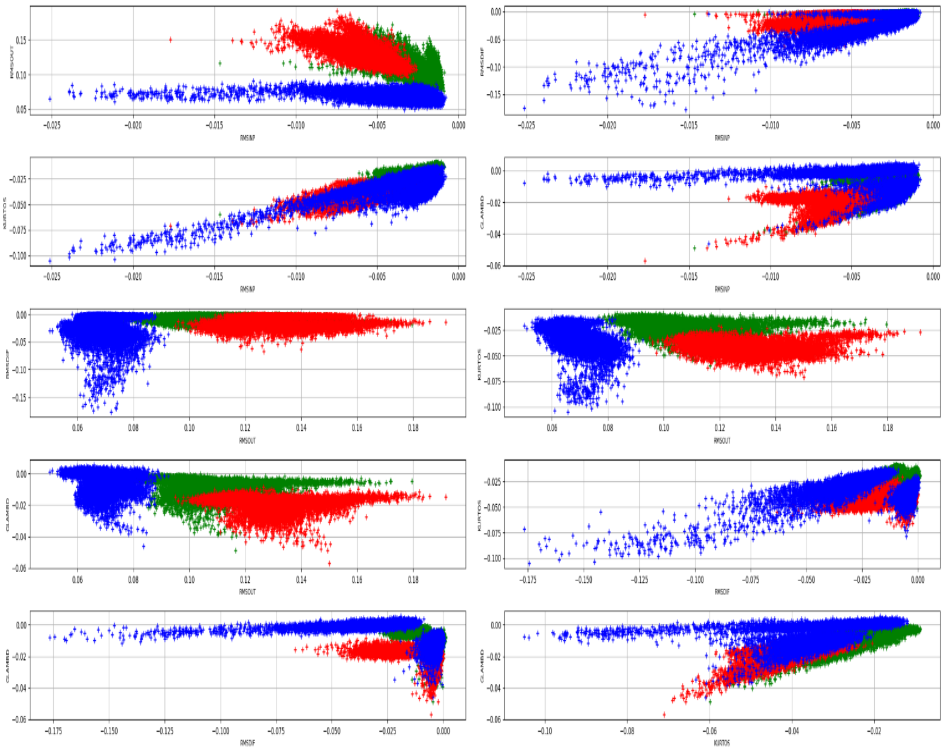


Figure 1. Scatter plot of five different correlation attributes.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2020 doi:10.20944/preprints202010.0048.v1

The environmental and acquisition constraints do not change significantly around a local spatial neighborhood and this implies that the quality of the denoising does not change either. As a result, we suppose that filtered signal quality (i.e., optimal, harsh, and mild) corresponding to the N successive shots in one swath of data is spatially co-located. Therefore, we decided to merge the attributes of the shot S_i of the swath L_j with the attributes of its $m = \frac{N}{2}$ previous neighbors and $m = \frac{N}{2}$ following neighbors. This technique allows for more robustness for harsh and mild outliers inside one swath of data. The curse of dimensionality generated by the use of a high number of statistical attributes might impact the accuracy of the QC denoise classification process. Therefore, a feature transformation and dimensionality reduction pre-process are required.

3.1. Feature transformation and dimensionality reduction

Feature transformation and dimensionality reduction techniques are essential to find a more separable transformed feature basis, to speed up the classification process, and to avoid overfitting. In the sections below, we detail two main feature transformation techniques: principal component analysis and independent component analysis.

3.1.1. Principal Component analysis (PCA)

PCA decomposition is useful especially for building uncorrelated features ranked decreasingly according to their amount of energy (i.e., eigenvalues of the attributes correlation matrix). A whitening process is then applied to normalize the principal component's variance by rescaling the eigenvalues of PCs by their inverse value and performing regularization by adding a small constant to the scaling factor in the case of small eigenvalues to avoid numerical instability.

The scatter plots of the harsh, mild, and prod data points show three potential clusters with polynomial shape. A kernel-based decomposition [4] might be useful to generate more separable attributes and then build a more confident decision space.
We chose to apply a three-degree polynomial-based PCA decomposition [4], as it illustrates the shape of the cross plot of the dataset on its corresponding attributes. The transformed dataset is then feed-forward to a linear and not a polynomial-based support vector machine classifier. Computing the kernel matrix as well as extracting its singular values is fastidious and computationally expensive due to the relatively large dataset.

3.1.2. Independent Component analysis (ICA [5])

Independent component analysis decomposition [5] aims to find the independent source from the previously computed uncorrelated features. Multiple ICA algorithms exist and were detailed in the section above. For computational purposes, we chose to use the Fast-ICA algorithm as it is based on an iterative optimization process. Although performing dimensionality reduction by selecting a subset of independent features is possible, we decided to select relevant features based on their statistical energy (i.e., before source separation process), as we are unable to select relevant features from the independent components. Feature transformation and dimensionality reduction are essential pre-processing tasks to speed up the training and validation process and to prevent classification from overfitting caused by the curse of dimensionality. The choice of the classifier is crucial, especially when handling large datasets.

3.2. Classification methods

Different algorithm performances were analyzed, i.e., multilayer perceptron layers (MLP), instance-based K-nearest neighbor (K-NN), random forest [8], and support vector machine (SVM) [7]. In the sections below, we will detail the different classifier structure used to predict the quality of the seismic data denoising process.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2020 doi:10.20944/preprints202010.0048.v1

3.2.1. Instance-Based K-Nearest Neighbor

With the K-NN algorithm, it is easy to implement lazy algorithms. It is less time consuming when training but computationally expensive when testing. This strategy provides accurate results by effectively using a richer hypothesis space (i.e., it uses many local linear functions to form its implicit global approximation to the target function). As a result of resource limitations inside the vessels, instant prediction of the production dataset might be computationally unaffordable.

3.2.2. Random forests

Random forest [8] is a very powerful classifier. The trees of the forest, and more importantly their predictions, need to be uncorrelated. To achieve this requirement, we chose the bootstrap aggregation method to perform the distribution of both data points and features over the different trees.

3.2.3. Support Vector machines

We chose a polynomial kernel SVM based on visual assessment of the scatter plot of data points projected on its PCs. The degree of the polynomial kernel was optimized using a grid-search method and the validation accuracy as a metric for maximization.
Many approaches exist when it comes to performing a multiclass support vector machine. We decided to implement and analyze the classification performance and the computation expenses of both one-vs.-one and one-vs.-the-rest-based support vector machines.

3.2.4. Multilayer Perceptrons

MLPs are very efficient, but they are memory and time consuming classifiers, especially when it comes to adjusting architecture and tuning hyperparameters. As the architecture of the model becomes deeper, the risk of overfitting is higher, especially when training on low feature datasets. Unlike all classifiers discussed previously, which have relatively few tunable parameters, the architecture of MLPs should be adjusted to fit a specific project. This task can be done automatically. In [10–12], different strategies for building an autonomous deep learning model were proposed. As a result of computational limitations, these models are trained for only one epoch, which might be appropriate when learning from data streams. This approach does not allow for the optimization process and therefore might skew the architecture growth procedure. In addition, authors in [10–12] observed potential forgetting phenomena when evolving the model architecture at the pace of training. We cannot afford to lose information from distant but very informative shots; therefore, we decided to use a more common but secure MLP-building strategy. We proposed a heuristic greedy-based algorithm to find the best neural network structure. It is based on the following fundamental assumption: without regularization, an optimal MLP structure tends to overfit as its depth is increased.

The pseudocode of the binary MLP generator 18 is inspired by the Branch and Bound algorithm [14]. It aims to build the multilayer perceptron model with the least validation loss of l_{min}. At each recursion, the MLP structure grows by appending to its final hidden layer one of the candidate layers L_c listed in the $Children(L)$ set of layers: the $Children(L)$ are a set of 32 hidden layers, and each one consists of N perceptrons ($64 < N < 2048$) followed by a 10% dropout layer to avoid overfitting. If a layer is appended to the optimal MLP model (i.e., it is part of the classifier that achieves the least validation loss), it is called "visited" and is added to the V_L set. All other candidate layers are added to forbidden layers set F_L. Therefore, at each stage, the proposed algorithm explores only the children of the visited layer L_C^*. If the algorithm fails to build a better MLP classifier (i.e., $L_c^* \neq \emptyset$), it automatically breaks. each candidate MLP is followed by a fully connected layer with one perceptron.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2020 doi:10.20944/preprints202010.0048.v1

Algorithm 1 Greedy Binary Multilayer Perceptron Model Generator.

Initialize:
$l_{min} = +\infty V_L \leftarrow \emptyset\ F_L \leftarrow \emptyset\ L \leftarrow L_0$
procedure **BINARYMLP**$(L, V_L, F_L, l_{min}, x, y)$
if $L \notin (V_L \cup F_L)$ **then**
for $L_c \in Children(L)$ **do**
$\widetilde{y} \leftarrow MLP(V_L, x)$
if $h(y, \widetilde{y}) > l_{min}$ **then**
$F_L \leftarrow F_L \cup \{L_c\}$
else
$l_{min} = h(y, MLP(x))$
$L_c^* \leftarrow L_c$
$V_L \leftarrow V_L \cup \{V_c\}$
if $L_c^* \neq \emptyset$ **then**
$L \leftarrow L_c^*$
$L_c^* \leftarrow \emptyset$
BinaryMLP$(L, V_L, F_L, l_{min}, x, y)$
else
Break

Algorithm 11 details the steps of the one-vs.-all binary classification process: The training and validation sets were converted to N class binary sets. We then applied the previously described greedy binary MLP generator to find $\mathcal{D}_{c_i}$, which denotes a locally optimal binary (class c_i) decision space. The final decision space is a combination of the binary spaces built so far (i.e., the predicted class belongs to the most confident decision space).

Algorithm 2 Multiclass Multilayer Perceptron Layers (MLP) Model Generator.

procedure **MULTICLASS MLP GENERATOR**(x, y, C)
$Y = \{\emptyset\}$
for c_i in C **do**
$X_B, y_B =$ **Bag**(x, y)
$y_B^{c_i} = \delta_{(c=c_i)}(y_B)$
Initialize $l_{min} = +\infty V_L \leftarrow \emptyset\ F_L \leftarrow \emptyset\ L \leftarrow L_0$
$\mathcal{D}_{c_i}$= **BinaryMLP**$(L, V_L, F_L, l_{min}, x_B, y_B^{c_i})$
$Y \leftarrow Y \cup \mathcal{D}_{c_i}(x)$
$\widetilde{Y}$=$Argmax_C(Y)$
Output $\widetilde{Y}$

4. Experiments

In this section, we detail several experiments using feature extraction, dimensionality reduction techniques, and classification methods.

4.1. Seismic Dataset

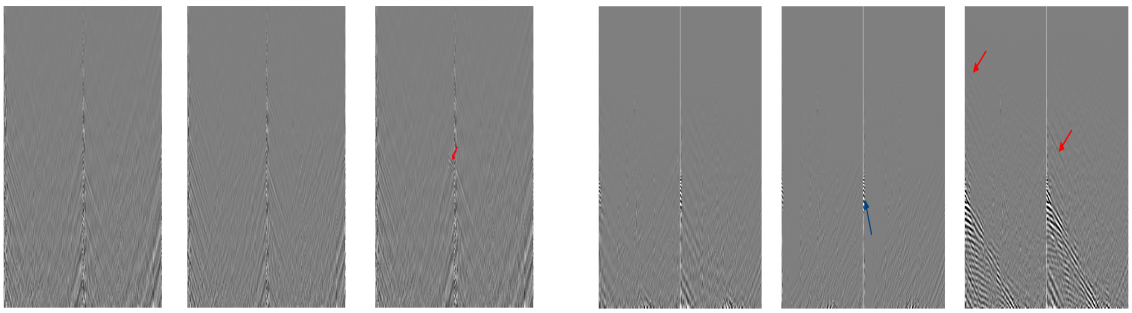
The original dataset provided by the off-shore production team consists of several projects (e.g., 4376exm, 4354pgs, 4050eni, 4169tot, and 3920pgs). Each one is a set of filtered data points aligned along different sail lines. To ensure correlation between successive shots in the same sail line, the attribute of each data point is merged with its *Winlen* previous and successive neighbor, ending up

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2020 doi:10.20944/preprints202010.0048.v1

with $(2 * Winlen + 1) * 5$ attributes per data point.

A cross-validation-based accuracy is computed after splitting the projects into five balanced folds in terms of class members to assess the performance of the tested classifiers.

During the testing phase, only two projects were used: 4376exm and 4354pgs. The 4376exm project consists of 40 sail lines (8889 shots): five sail lines for the training and validation process and 35 sail lines for the testing process. The 4354pgs project is composed of 19 sail lines (38,437 shots): five for the training and validation sets and 14 for the testing set.
Figures 2a,2b,3a show three examples of residual seismic data after optimal, harsh, and mild filtering processes of two projects: 4376exm and 4354pgs. The red arrows show a residual relevant signal (harsh denoise). The blue arrows illustrate DA artifacts which are not considered as residual noise.



(a) Residual seismic data of the 4376exm project. **(b)** Residual seismic data of the 4354pgs project.

Figure 2. Residual seismic data after prod, mild, and harsh filtering processes (from left to right) of two projects (4376exm and 4354pgs).

4.2. Feature transformation and dimensionality reduction

4.2.1. Principal Component Analysis

Principal component analysis decomposition was tested on four projects: 4111pgs, 4162pgs, 4374exm, and 4050eni with a *Winlen* equal to 15, 20, 12, and 25, respectively. As a result of computational limitations, kernel principal component analysis was only tested on three projects (4050eni, 4162pgs, and 4376exm) with a small *Winlen* value. Even with a small attributes number, performing kernel-PCA decomposition is memory and time consuming because of the large scale of the kernel matrix. To overcome this issue, all matrix operations were performed on a GPU (i.e, Nvidia Tesla V100 16 GB) through the scikit-Cuda library [15], and singular values were approximated using randomized singular value decomposition techniques [16].

4.2.2. Independent Component Analysis

Several techniques were tested to find the unmixing matrix that extracts the independent sources from the principal components. As a result of computational constraints, Informax- [17] and JADE-based [18] ICA decomposition algorithms could not be assessed (i.e., because of the very large correlation matrices). FastICA [5], which is an iterative-based algorithm, allows for a very fast and less time-consuming decomposition. This algorithm requires a fundamental assumption of non-Gaussianity of the original features. Therefore, a normal kurtosis hypothesis test was applied to the principal components of all the projects and very low p-value (between 0 and 1×10^{-6}) outcomes were obtained.

4.3. Classification methods

The number of nearest neighbors to consider in the instance-based nearest neighbor classifier is crucial. We applied a random search to find the k-value that minimizes the cross-validation error. As a

 doi:10.20944/preprints202010.0048.v1

result that the number of classes is odd (three classes), we decided to only consider k not divisible by three to avoid tied votes. We also decided not to use the weighted version of the K-NN algorithm, as the classification process is already robust to noisy data through the attribute reconstruction method. Although the instant-based algorithm is less time consuming when training, the prediction process is computationally expensive since one needs to scan all the training data to make a decision. Unlike lazy algorithms, the eager classifier training process is computationally less expensive.

The random forest [8] algorithm is one of them. Its hyperparameters were wisely chosen: We used the Gini metric [19] to measure the separability of data points over leaves. In [20], authors concluded that random forest classifiers should include at least 64 trees while not exceeding 128 trees to achieve a good balance between classification performance, memory usage, and processing time. We introduced the early-stopping criterion to choose the appropriate number of trees within the range previously stated. Although the pruning process is required to avoid overfitting, when it comes to using decision tree classifier, the random forest classification method already handles the overfitting issue through features and data bagging.

In contrast, support vector machines and more specifically kernel-based support vector machines are easy-to-train classifiers as they only require kernel space parameter tuning. A grid search-based approach showed that a third-degree polynomial kernel space fits the shape of the clusters of the projected data points. A multiclass support vector machine is based on a combination of binary support vector classifiers with different voting strategies. The one-vs.-one classification strategy is appropriate when handling imbalanced data which is not relevant since all our projects include three categories of filtered shots in the same proportion. Therefore, we decided to use the one-vs.-the-rest-based SVM classifier since it achieves good performance with balanced data and is less memory and time consuming (i.e., $(n+1)$ is the binary classifier for the one-vs.-all SVM classifier, whereas it is $\frac{n(n+1)}{2}$ for the one-vs.-one SVM classifier).

Unlike all the classifiers we have already analyzed, neural networks are less stable when training due to the relatively high number of tunable parameters and the disparity of the projects in terms of their mean value. The hyperparameters of the neural network have to be wisely chosen. Although RMSprop [21], Adadelta [22], and Adam [23] are very similar algorithms, authors in [23] show that Adam's bias-correction outperforms RMSprop towards the end of optimization as gradients become sparser. Therefore, in all the following experiments, we trained our MLPs using the Adam [23] optimizer with an initial learning rate tuned to 10^{-3}.

The greedy multilayer perceptron model generator described in the previous section is a project adaptive algorithm that builds a local optimal neural network structure. Each neural network candidate is trained for 20 epochs with a categorical cross-entropy loss.

Once the optimal structure was built, we retrained it for 200 epochs and we only saved the weights with a minimum validation error.

4.4. Assessment of the predicted quality control of seismic data after filtering

The QC denoise classification assessment was performed using a combination of three binary-classification evaluation tasks (i.e., harsh, mild, and optimal prediction). Misclassifying shots with signal leakage is considered to be a critical issue due to the high impact of harsh filtering on the relevant signal. As a result, we paid more attention to the harsh classification of a false negative. Undetected harshly filtered shots might be either the result of a highly confident but misleading mild or optimal classifier, a non-confident but correct harsh classifier, or both of them. Therefore, we decided to use two balanced F_β[24] to assess our binary classifiers: The first, with $\beta = 2$, which weighs sensitivity higher than specificity when assessing the harsh classifier performance; the second, with $\beta = 0.5$, which weighs specificity higher than sensitivity when assessing the mild or the optimal classification performance.

Despite the relevance of the F_β [24] and other related statistics-based metrics, visual assessment of the residual signal map by geophysicists remains the most reliable way to assert the seismic data

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2020 doi:10.20944/preprints202010.0048.v1

denoise quality control. Therefore, all the filtered test sail line shots, also known as the production lines, were verified by a team of geophysicists.

5. Results and discussion

In this section, we assess and discuss the different feature transformation and dimensionality reduction strategies. We then compare the classification strategies using the metrics so far described.

5.1. Feature transformation and dimensionality reduction methods

Table 1 shows the precision, the recall, and the unbalanced F1-score, as well as the computation time of the classification process (i.e., support vector machines) after applying different features and transformation techniques.

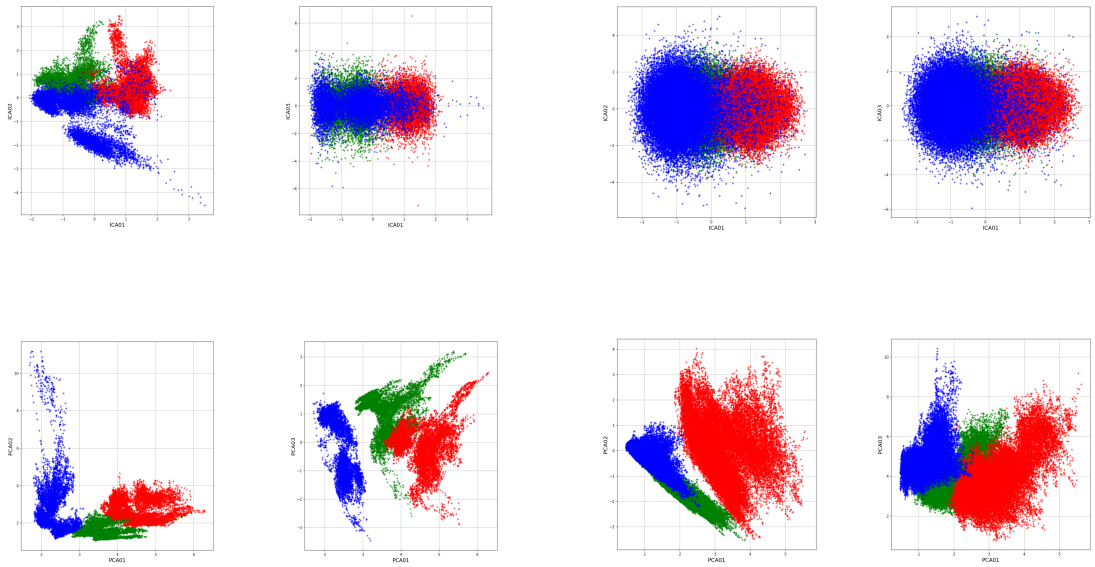
Table 1. Classification and time processing assessment of different feature transformation and dimensionality reduction techniques.

Project	FT^1 and DR^2 methods	F_β score [24]			Processing time (s)
		Optimal	Harsh	Mild	
4376exm	whitening	93.1	99.3	98.4	130
	PCA	**93.8**	**100**	**98.2**	**100**
	Kernel PCA	91.5	98.3	97.3	5300
	ICA	87.1	99.0	97.3	540
4354pgs	whitening	95.7	91.2	90.3	1432
	PCA	**96.8**	**93.6**	**90.5**	**1054**
	Kernel PCA	91.5	91.7	88.9	10321
	ICA	95.0	90.9	84.3	2630

[1] Feature transformation, [2] dimensionality reduction.

Table 1 illustrates the processing time of different feature transformation techniques and shows the classification performance of an SVM classifier trained on those transformed data. Principal component analysis was found to be less time consuming and generate more separable features. The whitening process consisting of a simple standard scalar operation achieves classification performance close to the PCA decomposition since it encompasses more informative data. This justifies the small computational time of the PCA decomposition compared to the whitening process. The processing time metric includes feature transformation pre-processing and classification time. The PCA decomposition itself is more time consuming than a simple standard scalar operation. As a result, principal component analysis significantly reduces the classification time which confirms our intuition. Unexpectedly, the independent components analysis decomposition [5] led to a relatively poor classification performance. Figure 3a and 3b illustrate the non-correlation between the features of independence and class separability which impacts the decision boundaries generation.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2020 doi:10.20944/preprints202010.0048.v1



(a) Cross plots of ICs and PCs of the 4376exm project. **(b)** Cross plots of ICs and PCs the 4354pgs project.

Figure 3. Cross plots of the first, second, and third independent and principal components of the 4376exm and 4354pgs projects.

Principal component analysis decomposition led to the best classification performance while limiting the computational expense (i.e., processing time). Therefore, PCA pre-processing was applied at the beginning of each QC prediction experiment.

5.2. Classification algorithms

In this section, we evaluate the efficiency of instance-based nearest neighbor, the support vector machines, the random forest classifier, and the MLP predictor in terms of classification performance and computational costs.

Table 2. Classification performance of different models applied to two projects (4376exm and 4543pgs).

Project	Classifier	F_β score [24] (%)		
		Optimal	Harsh	Mild
4376exm	Support Vector Machine	95.1	93.7	93.3
	Random Forest	93.4	94.7	**99.3**
	Multilayer Perceptron	**95.5**	**95.6**	98.0
	Instance-based Nearest Neighbors	71.6	89.7	82.4
4354pgs	Support Vector Machine	**93.6**	**99.7**	**90.9**
	Random Forest	91.7	97.8	83.7
	Multilayer Perceptron	93.5	99.2	90.9
	Instance-based Nearest Neighbors	87.6	92.1	91.4

According to Table 2, the instance-based nearest neighbor classifier performed relatively poorly in terms of classification compared to the eager algorithms. Multilayer perceptron outperformed the support vector machine and random forest for the harsh-vs.-the-rest and optimal-vs.-the-rest classification problem of project 4376exm, while the support vector machine achieved the best classification performance of the 4354pgs project. Although the classification assessment metrics (precision, recall, F1-score) are crucial to evaluate and compare the performance of the models, the computational costs are a key parameter, especially when it comes to processing the quality control of the filtered seismic data inside the vessels. Therefore, we only considered the models with relatively low training and prediction latency.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2020 doi:10.20944/preprints202010.0048.v1

Table 3. Processing time and memory usage of different classification strategies of two projects (4354pgs and 4376exm).

Project	Classifier	Processing Time		Memory Usage (kB)
		Training Phase	Prediction Phase	
4376exm	Support Vector Machines	**15.19**	2.98	**355.5**
	Instance-based Nearest Neighbor	96.8	57.06	6900.5
	Random Forest	15.57	**1.68**	5223.6
	Multilayer Perceptron	930.4	3.6	31,291.8
4354pgs	Support Vector Machines	633.10	27.88	**983.41**
	Instance-based Nearest Neighbor	3880.5	27.88	29,606
	Random Forest	**128.79**	**7.69**	21,197
	Multilayer Perceptron	9304	6.5	69,350.6

The processing time of one classification approach includes the model hyperparameters and structure tuning procedure as well as the learning process. Multilayer perceptrons and instance-based nearest neighbor are relatively more time consuming than the other classifiers (i.e., support vector machine and random forest) which are affected by their tuning and learning process: The multilayer perceptron structure generation is fastidious due to the high but necessary number of epochs per iteration (20 for tuning and 200 for learning). Indeed, we observed that when decreasing the number of epochs, the MLP candidate was underfitted, which impacts the relevance of the optimal number of perceptron selection for each layer. The data fitting process was relatively less time consuming due to the low number of batches: 544 batch for the 4543pgs project and 276 batches for the 4376exm project. Each batch consists of 50 data points. The training phase of the instance-based nearest neighbor classifier was unexpectedly more time consuming than the prediction phase due to the nearest neighbors considered in the tuning process. The memory usage observations confirm the conclusion derived from the analysis of the Processing time of both the training and prediction phase of each classifier: To find the suboptimal MLP structure, the generator algorithm tested several architecture candidates, each one consisting of between 4099 and 1,082,371 trainable parameters. We ended up with hundreds of millions of generated parameters and hence a very memory consuming process. Given the results stated in Table 3, the support vector machine achieved the best trade-off between computational expenses reduction and good classification performance. Therefore, we chose it for the testing phase. We had already statistically assessed the feature transformation and classification techniques and decided to proceed with principal component analysis decomposition while keeping only 80% of the total energy as a pre-processing step, as well as choosing the SVM model with a third-degree polynomial kernel for the classification process. In the next section, we assess the efficiency of our candidate strategy on a massive test set of filtered seismic data. This step was fully assessed by a geophysicist.

5.3. Geophysical assessment of QC denoise prediction process

After performing spatial augmentation, feature transformation, dimensionality reduction, and then classification, we predicted the QC map of the remaining testing sail lines of the 4376exm project. Figure 4 shows the final QC-map after achieving the pseudo-labeling and the retraining strategy described in the previous section.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2020 doi:10.20944/preprints202010.0048.v1

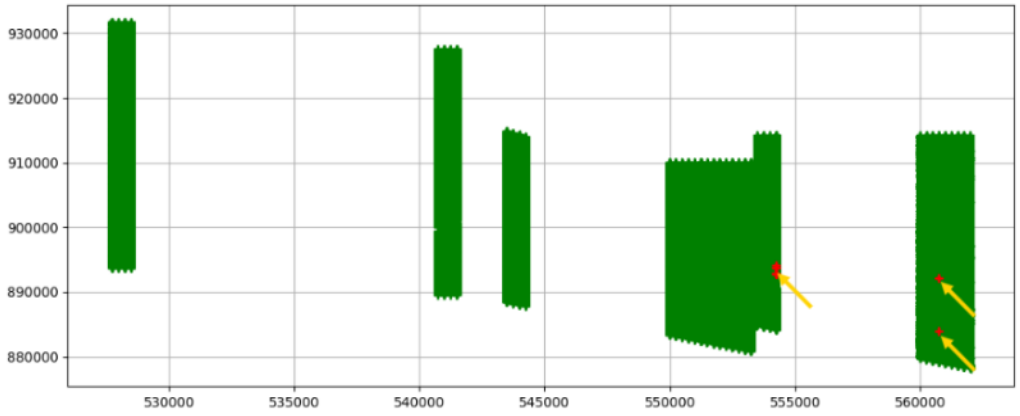
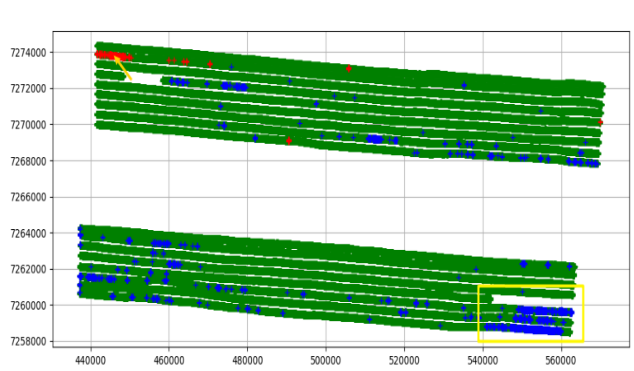
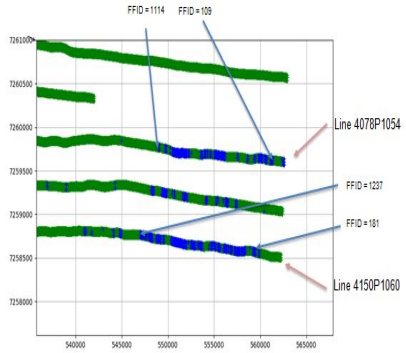


Figure 4. Quality control (QC) map of the 4376exm project.

The QC map was validated by the production team. According to the geophysicists, the shots in the lower right corner of Figure 5a (as shown by the yellow arrows) were misclassified: The detected leakage was a false alarm. We predicted the residual noise (mild filtering) in some production sail lines of the 4354pgs project; these are shown inside the yellow box in the figure below, which can be seen zoomed in in Figure 5b. The shots denoted by the yellow arrows and predicted as harshly filtered are false alarms.



(a) QC map of the 4354pgs project.



(b) Zoom (x3) of the sail lines contaminated with mild noise.

Figure 5. Assessment of the QC prediction of the 4354pgs project.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2020 doi:10.20944/preprints202010.0048.v1

Our predictions of mild noise in some shots, especially those belonging to the 060 sail line, were validated by the geophysicists of the production team. The red arrows inside Figure 6 highlights the mild noise inside one seismic image of the 60 sail line.

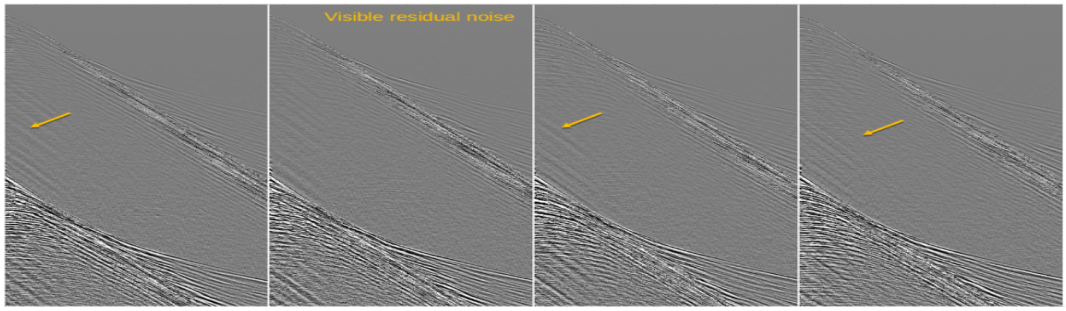


Figure 6. Filtered seismic image of the 60 sail line.

6. Conclusion

Here, we demonstrated that machine learning can automate the QC denoising process following a common pipeline of feature transformation and a dimensionality reduction step; we then predicted the decision space of harsh, mild, and optimal filtering. Principal component analysis decomposition allows for the best feature separability while reducing the feature space dimensionality. The support vector machine achieved the best trade-off between computational cost reduction and classification performance. More attention must be paid to neural networks to reduce their memory usage and time consumption by improving the efficiency of the automotive neural network generation strategy. The spatial correlation between successive shots provided valuable information about the QC denoising process of a sequence of shots; therefore, this must be further explored and evaluated.

Acknowledgments: The authors would like to thank the crew of RAMFORM STERLING who acquired the data used in this test, and PGS for permission to publish these results.

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
PCA	Principal component analysis
ICA	Independent component analysis
MLP	Multilayer perceptron
K-NN	K-nearest neighbors
SVM	Support vector machines

Acknowledgments: We would like to thank the crew of Ramform Sterling who acquired the data used in this test, and PGS for the permission to publish these results.

References

1. Mandelli, S.; Lipari, V.; Bestagini, P.; Tubaro, S. Interpolation and Denoising of Seismic Data using Convolutional Neural Networks, 2019, [arXiv:cs.NE/1901.07927].
2. Spanos, A.; Bekara, M. Using Statistical Techniques to Improve the QC Process of Swell Noise Filtering **2013**. doi:https://doi.org/10.3997/2214-4609.20130884.
3. Bekara, M. Automatic Quality Control of Denoise Processes Using Support-Vector Machine Classifier **2019**. *2019*, 1–5. doi:https://doi.org/10.3997/2214-4609.201900845.
4. Scholkopf, B.; Smola, A.; Müller, K.R. Kernel principal component analysis. ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING. MIT Press, 1999, pp. 327–352.
5. Hyvärinen, A.; Oja, E. Independent component analysis: algorithms and applications. *Neural Networks* **2000**, *13*, 411 – 430. doi:https://doi.org/10.1016/S0893-6080(00)00026-5.

6. Hearst, M.A.; Dumais, S.T.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE Intelligent Systems and their Applications* **1998**, *13*, 18–28.
7. Milgram, J.; Cheriet, M.; Sabourin, R. "One Against One" or "One Against All": Which One is Better for Handwriting Recognition with SVMs? **2006**.
8. Breiman, L. Random forests. *Machine learning* **2001**, *45*, 5–32.
9. Adnan, M.N.; Islam, M.Z. One-vs-all binarization technique in the context of random forest. Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2015, pp. 385–390.
10. Inoue, H.; Narihisa, H. Efficiency of Self-Generating Neural Networks Applied to Pattern Recognition. *Mathematical and Computer Modelling* **2003**, *38*, 1225–1232. doi:10.1016/S0895-7177(03)90124-5.
11. Pratama, M.; Za'in, C.; Ashfahani, A.; Ong, Y.S.; Ding, W. Automatic construction of multi-layer perceptron network from streaming examples. Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 1171–1180.
12. Ashfahani, A.; Pratama, M. Autonomous Deep Learning: Continual Learning Approach for Dynamic Environments. *Proceedings of the 2019 SIAM International Conference on Data Mining* **2019**, p. 666–674. doi:10.1137/1.9781611975673.75.
13. Chen, C.C.; Chu, H.T. Similarity Measurement Between Images. 2005, Vol. 2, pp. 41 – 42 Vol. 1. doi:10.1109/COMPSAC.2005.140.
14. Kolesar, P.J. A branch and bound algorithm for the knapsack problem. *Management science* **1967**, *13*, 723–735.
15. Givon, L.E.; Unterthiner, T.; Erichson, N.B.; Chiang, D.W.; Larson, E.; Pfister, L.; Dieleman, S.; Lee, G.R.; van der Walt, S.; Menn, B.; Moldovan, T.M.; Bastien, F.; Shi, X.; Schlüter, J.; Thomas, B.; Capdevila, C.; Rubinsteyn, A.; Forbes, M.M.; Frelinger, J.; Klein, T.; Merry, B.; Merill, N.; Pastewka, L.; Liu, L.Y.; Clarkson, S.; Rader, M.; Taylor, S.; Bergeron, A.; Ukani, N.H.; Wang, F.; Lee, W.K.; Zhou, Y. scikit-cuda 0.5.3: a Python interface to GPU-powered libraries, 2019. http://dx.doi.org/10.5281/zenodo.3229433, doi:10.5281/zenodo.3229433.
16. Martinsson, G.; Gillman, A.; Liberty, E.; Halko, N.; Rokhlin, V.; Hao, S.; Shkolnisky, Y.; Young, P.; Tropp, J.; Tygert, M.; others. Randomized methods for computing the Singular Value Decomposition (SVD) of very large matrices. *Works. on Alg. for Modern Mass. Data Sets, Palo Alto* **2010**.
17. Nadal, J.P.; PARGA, N. Sensory coding: information maximization and redundancy reduction. In *Neuronal Information Processing*; World Scientific, 1999; pp. 164–171.
18. Rutledge, D.N.; Bouveresse, D.J.R. Independent components analysis with the JADE algorithm. *TrAC Trends in Analytical Chemistry* **2013**, *50*, 22–32.
19. Dagum, C. Decomposition and interpretation of Gini and the generalized entropy inequality measures. *Statistica* **1997**, *57*, 295–308.
20. Oshiro, T.; Perez, P.; Baranauskas, J. How Many Trees in a Random Forest? 2012, Vol. 7376. doi:10.1007/978-3-642-31537-4_13.
21. Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* **2012**, *4*, 26–31.
22. Zeiler, M.D. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* **2012**.
23. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**.
24. Baeza-Yates, R.; Ribeiro-Neto, B. Modern Information Retrieval **2011**. *2011*, 327–328.