Machine Learning Applications using Seismic Attributes – A Hands-On Course

Part 5c: Supervised Classification using PNN

AASPI Lab Exercises written by:

David Lubo-Robles
Thang Ha
Kurt J. Marfurt

The University of Oklahoma
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>2</td>
</tr>
<tr>
<td>Probabilistic Neural Networks (PNN)</td>
<td>2</td>
</tr>
<tr>
<td>Overview of Probabilistic Neural Networks (PNN)</td>
<td>4</td>
</tr>
<tr>
<td>The Machine Learning Toolbox</td>
<td>5</td>
</tr>
<tr>
<td>Step 1: Plot and define polygons</td>
<td>6</td>
</tr>
<tr>
<td>Step 2: Convert polygons to point sets</td>
<td>12</td>
</tr>
<tr>
<td>Step 3: Extract user-defined training data</td>
<td>15</td>
</tr>
<tr>
<td>Supervised Training Data Extraction</td>
<td>16</td>
</tr>
<tr>
<td>Step 4: Analyze supervised algorithms</td>
<td>17</td>
</tr>
<tr>
<td>Step 5: Create a supervised model</td>
<td>22</td>
</tr>
<tr>
<td>Step 6: Classify input data</td>
<td>25</td>
</tr>
<tr>
<td>Exhaustive PNN-based results</td>
<td>27</td>
</tr>
<tr>
<td>References</td>
<td>28</td>
</tr>
</tbody>
</table>

## Introduction

The goal of supervised learning is to mimic a skilled interpreter. The target seismic facies may be either a lithology (e.g., salt, dolomite, sandstone), a geomechanical rock type (brittle vs. ductile, porous vs. nonporous) or an architectural element (mass transport complex, point bar, karst collapse). Ideally, if an interpreter uses well control, geologic insight, and experience to define (by picking polygons) seismic facies on a suite of coarsely spaced inlines, crosslines, and time slices, the supervising learning algorithms should be able to use these labeled facies as training and subsequently assign every voxel to one of the desired classes.

In the previous section on unsupervised learning, we noted that a certain level of supervision exists through the selection of attributes and the location of the training data. Using these concepts, we introduced semi-supervised implementations of both the SOM and GTM algorithms.

In supervised learning we can carry this attribute bias one step further to quantitatively measure the optimum number and combination of attributes that best discriminate the seismic facies of interest from each other and from the background sedimentary matrix.

## Probabilistic Neural Networks (PNN)
Working with seismic attributes and well log data, Hampson et al. (2001) showed stepwise linear regression to be an effective means of picking the best number and collection of seismic attributes to predict a desire well log property. However, there are two limitations to this workflow: (1) it does not test all the possible combinations of seismic attributes, and (2) by searching for linear relationships between the attributes, it can miss non-linear relationships existing between them. For our problem, the limitation is slightly different; rather than predict a specific petrophysical property by correlating attributes to that property, we wish to differentiate one or more seismic facies from each other and the background pattern.

Because of artificial neural network’s capacity in exploring non-linear relationships, we evaluate whether a novel technique called exhaustive PNN which, coupling Probabilistic Neural Networks with an exhaustive search algorithm, can determine the best combination of seismic attributes to distinguish between salt vs. nonsalt seismic facies in a Gulf of Mexico 3D seismic survey.

In the proposed exhaustive PNN workflow for attribute selection and supervised seismic facies classification (Lubo-Robles et al., 2021). First, we use our geological insight to choose a suite of candidate seismic attributes. Next, we apply 3D Kuwahara median filter to smooth and block the attributes (Qi et al., 2016), preconditioning them for subsequent classification. We also define a group of polygons for each facies which represent the training and validation datasets. Next, a robust scaling scheme is applied to avoid any bias associated with different units between the seismic attributes. To initialize the exhaustive PNN algorithm, we define our first seismic attribute combination and an initial smoothing parameter, \( r \). We test a suite of values ranging from \( 0.5 \leq r \leq 3.0 \) and compute the validation error \( E_v \) in each iteration. These combinations are ranked based on their \( E_v \) and the best combination of seismic attribute and smoothing parameter \( r \) is given by the smallest \( E_v \) while maintaining a balanced bias-variance trade-off. As an optional step, a first-order gradient optimization technique called Adam (Kingma and Ba, 2015) can be applied to the best combination to relax the fixed smoothing parameter \( r \) condition imposed by the exhaustive search algorithm and further minimize the validation error. Finally, using the exhaustive PNN attribute subset, we perform our supervised seismic facies classification, and we compute the probability of each class, which measures the confidence in the classification.
Overview of Probabilistic Neural Networks (PNN)

Probabilistic Neural Networks (PNNs) are feedforward neural networks that use Bayes’s criteria and Parzen windows to estimate the probability density function from random samples, and then classify an unknown variable into a certain class (Specht, 1995; Masters, 1995; Hajmeer and Basheer, 2002). According to Masters (1995), the most common kernel function used in the Parzen method is the Gaussian function. For this reason, given a set of training attributes \( \mathbf{a} \), the average estimated density function \( g_k(\mathbf{x}) \) is given by:

\[
g_k(\mathbf{x}) = \frac{1}{N_k} \sum_{n=1}^{N_k} \exp \left[ - \sum_{m=1}^{M} \frac{(x_m - a_{nm})^2}{r^2} \right]
\]  

(1)

where, \( N_k \) is the number of training samples associated with the \( k \)-th class defined by the training data, \( M \) is the number of input attributes, \( \mathbf{x} \) is a validation seismic attribute vector, and \( r \) is a smoothing parameter that requires careful selection through training.

Following Masters (1995), to classify an unknown sample, the PNN starts by computing the distance between the validation attributes and the training attributes. This difference is then input into the Gaussian activation function in equation 1 (Masters, 1995). In the summation layer, the PNN calculates the average estimated density function \( g_k(\mathbf{x}) \) for each class. Finally, in the output layer, the PNN assigns the unknown sample to class \( q \) where

\[
g_q(\mathbf{x}) \geq g_k(\mathbf{x}) \quad k = 1, 2, ..., K
\]  

(2)

PNN also provides confidence estimates of the classification (Masters, 1995) given by:

\[
P_k(\mathbf{x}) = \frac{1}{N_k} \sum_{n=1}^{N_k} \delta_{nk} \exp \left[ - \sum_{m=1}^{M} \frac{(x_m - a_{nm})^2}{r^2} \right]
\]

\[
\sum_{k=1}^{K} \left[ \frac{1}{N_k} \sum_{n=1}^{N_k} \delta_{nk} \exp \left[ - \sum_{m=1}^{M} \frac{(x_m - a_{nm})^2}{r^2} \right] \right]
\]  

(3)

where, \( P_k \) represents the normalized probabilities given by the estimated probability density function of each class \( k \), \( g_k(\mathbf{x}) \), divided by the sum of all the density functions of all \( K \) classes. The Kronecker delta \( \delta_{nk} \) is equal to 1 if the training case \( n \) belongs to class \( k \), and 0 otherwise.
Optimizing the smoothing parameter \( r \)

In order to optimize \( r \), we use an exhaustive search algorithm to test a range of values for \( r \), and select the one associated with the minimum continuous error \( e_k(x) \) defined by Masters (1995) as:

\[
e_k(x) = [1 - P_k(x)]^2 + \sum_{j \neq k} |P_j(x)|^2
\]  \hspace{1cm} (4)

Because we are interested in the error of all the samples, we define the global error as

\[
E = \frac{1}{H} \sum_{k=1}^{H} e_k(x_k)
\]  \hspace{1cm} (5)

where, \( H \) is the number of validation samples.

---

Figure 1. Exhaustive PNN-based workflow for attribute selection (Lubo-Robles et al., 2021)

The Machine Learning Toolbox

The Machine Learning Toolbox GUI is launched from the Machine Learning Toolbox tab on the main aaspi_util GUI. The toolbox walks the user through a suite of steps useful in unsupervised and supervised classification.
Step 1: Plot and define polygons

In supervised classification, interpreters first pick a set of seismic voxels and assign it to specific facies of interest. The most intuitive way to assign a set of seismic voxels to facies is to pick a polygon on a 2D display (inline profile, crossline profile, or time slice). Using (1) the **plot and define polygons** module, interpreter can pick polygons of different facies on either the original seismic amplitude volume or on a blended image of multiple attribute volumes.

In this example, we will co-render a seismic amplitude volume with a coherence volume in order to pick salt facies. To do so, (2) browse the base layer to the seismic amplitude volume, (3) set the color bar to be *red_white_blue.alut*, and then for scaling, check (4a) statistical scaling, (4b) set the range to be between the (4b) 5 and 95 percentiles, and (because seismic amplitude exhibits both peaks and troughs), do not place a check mark in for (4c) **All positive values**:
Browse the 2\textsuperscript{nd} layer and (5) select the coherence volume, (6) set its color bar to *monochrome black*, select (7) the opacity curve type to be *high values transparent, low value opaque*, and (8) define its scale parameters:

![Image of a GUI interface for a seismic processing tool]  

Note that you can also plot a single attribute in the base layer and perform polygon picking, without the need to corender multiple attributes. However, we recommend co-rendering two to three attributes together to better delineate different facies.

Finally, (9) modify the plot title as needed, (10) define the geometric plot parameters.
Press the *Execute corender* button (green arrow) after which you should obtain an inline image like this one:

**Warning!**

When the polygon mode is enabled, you cannot move to next display panel. To move to a different line, uncheck the polygon mode box first. In this manner, each polygon is associated with one and only one vertical or time slice.
To pick a polygon, (11) place a checkmark before Polygon mode. The facies name text field and the three other buttons enclosed in the red box above will be enabled.

Specify (12) facies name and (13) start clicking to define a polygon around the facies of interest. For now, this polygon tool does not allow the user to edit the coordinates of the picked vertices, so if you make a mistake in picking, (14) you will need to click the clear polygon button and start over again. After you finish picking a polygon, (15) click on the Save polygon to file... button to (16) save the picked polygon to an AASPI text-based polygon file (*.plg). For your reference, you can also (17) load and display a previously picked polygon using Load polygon from file... button. However, keep in mind that the display configuration of the corender program must match that of the polygon file (e.g., you cannot display a time-slice and load a polygon that was picked on an inline profile).
Below is a sample of what an AASPI polygon file looks like in a text editor:
I can open or display the Salt_Line_270 file using my favorite editor and find that the first 6 lines of a polygon file list the:

- Facies name,
- Panel type (in this case, inline),
- 1\textsuperscript{st} column (in this case, crossline),
- 2\textsuperscript{nd} column (in this case, time),
- Panel number (in this case, the inline no. on which the polygon was picked), and
- Number of picked polygon vertices.

The next 13 lines consist of 2 values defining a polygon vertex in the order in which they were picked. These polygons will be converted to sets of points (i.e., voxel coordinates) in the next step of the AASPI Machine Learning Toolbox: \textit{Convert polygons to point sets}. Before doing so, I pick a polygons on a few lines to define the NoSalt seismic facies:
Step 2: Convert polygons to point sets

In supervised machine learning workflow, after you defined different polygons for different facies of interest in step 1, the next step is to convert those polygons to point sets. Each point in a point set is defined by inline, crossline, and time/depth. To convert polygons to point sets, go to the Machine Learning Toolbox, and select (1) **Convert polygons to point sets**.
Click (2) **Browse and add to current list** to select which polygons you wish to (3) convert to points and hit OK. The (4) list of input polygon files will be updated, and (5) the associated facies names will be automatically loaded from input polygon files. If for some reason, you need to change facies name, double click on the facies name table, and start typing. The changes of facies names will be automatically saved when you execute this program, but if you want to save the polygon’s facies name right away, you can (6) click on **Save modified facies names to polygon files**.
Next, (7) browse to the AASPI volume that you used for picking the polygons. The geometric information of this volume (sampling interval, number of samples, crosslines, inlines, etc.) are critical for polygon-to-points conversion. Specify (8) the Unique project name and Suffix. By default, (9) the point set converted from a polygon will be decimated at 5 inlines x 5 crosslines x 5 vertical samples. You can change these default parameters to different values. This decimation will be beneficial if the machine learning algorithm of choice is computationally intensive (such as Probabilistic Neural Network - `pnn_facies`). The computational cost increases geometrically with the number of points in a point set.

After all parameters are set, Execute (green arrow).

Program **Convert polygons to point sets** will always generate the following output files:

<table>
<thead>
<tr>
<th>Output file description</th>
<th>File name syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point set name</td>
<td><code>polygon_file_name_unique_project_name_suffix.pnt</code></td>
</tr>
<tr>
<td>Program log information</td>
<td><code>polygon_to_points_unique_project_name_suffix.log</code></td>
</tr>
<tr>
<td>Program error/completion information</td>
<td><code>polygon_to_points_unique_project_name_suffix.err</code></td>
</tr>
</tbody>
</table>

Below is a sample of what a converted point set file looks like in a text editor:

The (11) first 4 lines of a converted point set file is: Facies name, 1\(^{st}\) column (time/depth), 2\(^{nd}\) column (crossline number), and 3\(^{rd}\) column (inline number). Each subsequent line consists of 3 values (12) defining the geometric coordinate of a point within a seismic volume: time/depth, crossline, and inline values.

Since AASPI can only define a polygon in either inline, crossline, or time slice display, AASPI point set exhibits a 2D structure. However, if you can define a 3D point set from an external program, you can mimic the structure of AASPI point set file to create a 3D AASPI point set.
These geometric coordinates will be used to extract data value from seismic attribute in the next step of AASPI machine learning workflow: **Extract user-defined training data**

**Step 3: Extract user-defined training data**

Recall that for unsupervised classification, the first step is to generate a subset of input attributes via decimation (i.e., skipping while reading data at a specific interval, such as 5 inlines x 5 crosslines x 5 vertical samples). In contrast, for supervised classification, after you convert polygon to point sets, the next step is to extract input attributes at those points to generate training data. Go to **Machine Learning Toolbox** and select (1) **Extract user-defined training data**.
Click (2) *Browse and add to current list* to select input attributes for training data extraction. These attributes should be preconditioned via Kuwahara filtering (see Appendix A in these course notes) to reduce noise and give a more realistic, geological classification, **UNLESS** you want to perform fine-scaled reservoir characterization using inversion products (density, impedance, \(v_p/v_s\) ratio, etc.) as input attributes.

Choose (3) whether you want dead traces and mute zones to be ignored in extraction and whether to extract the coordinates of data points as spatial attributes. If the data are in time domain, specify a constant velocity to approximately convert time to depth. As with most AASPI applications (4) define a *Unique project name* and a *Suffix*.

**Supervised Training Data Extraction**

For supervised classification, (5) click on the *Point set* tab. Browse to the (6) AASPI-formatted point set files, which were converted from polygons in the previous step of AASPI machine learning workflow. A (7) white box displays detected facies names from the browsed point sets. This list is NOT editable and is for informational purpose only. However, the order of facies names can be changed by (8) *Move selected facies UP/Down* buttons.

After browsing point sets, press (9) *Extract training data from point sets (green arrow).*
Program **Extract user-defined training data** will always generate the following output files:

<table>
<thead>
<tr>
<th>Output file description</th>
<th>File name syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training data</td>
<td>training_data_point_set_file_name_attribute_name.H</td>
</tr>
<tr>
<td>Program log information</td>
<td>polygon_to_points_unique_project_name_suffix.log</td>
</tr>
<tr>
<td>Program error/completion information</td>
<td>polygon_to_points_unique_project_name_suffix.err</td>
</tr>
</tbody>
</table>

These training data files, and text-based lists will be used to analyze input and create model for AASPI machine learning workflow in subsequent steps.

**Step 4: Analyze supervised algorithms**

To understand the significance of each input attribute, point set, and modeling parameter towards the final result, it is essential to analyze the input training data and test different parameters so users can determine the most optimal attributes, point sets, and modeling parameters for classification. To analyze input training data, go to Machine Learning Toolbox, and select (1) **Analyze supervised algorithms**

Click on the (2) **Input Training Data** tab. To quickly browse all training data generated in step 3, click (3) **Load, and append a list from a text file** button, then select the extracted list text file. This list contains all extracted training data files from generate training data step. If you decide to remove some training data files from the list, make sure to (4) click **Check for missing training data files and number of samples consistency** button. If there is no error, you can proceed. It is not recommended to remove training data files directly this way, but rather removing all training data belonging to a specific attribute or point set.

Click on (5) the **Attributes** tab to view the automatically detected attribute list associated with the input training data. You can move attribute up/down in the list. To remove some attributes and all of theirs associated training data files, (6) first highlight them in the list by clicking on them, and then (7) click **Remove selected attributes from current list** button. If you want to reset to the default attribute list, (8) click on the **Rescan Attribute** button. Since 2020, a new function transformation panel (red rectangle) was implemented in AASPI, and it allows the user to apply any type of scaler algorithm to the input data. For this example, we use the Automatic
transformation type that applies a lognormal scaling or a Z-score normalization if the distribution of the seismic attribute is close to a Gaussian.

Click on (9) the Point sets tab to view and define training point sets and validation point sets. Currently AASPI support two methods of training/validation data distribution (10):

- **User-defined**: the user will decide which point sets to be used for training and which for validation. To transfer some training point sets to the validation list, first select the training point sets you want to move by (11) clicking on them, then (12) clicking on the transfer button to swap them to the validation list (and vice versa). The analysis is then performed just on that particular training-validation setting.

- **One-point-set cross-validation**: The program will loop through all training point sets and choose one for validation. If there are \( Q \) training point sets, there will be \( Q \) analyses. Transfer buttons and validation list will be disabled for this method.
To remove a point set and all associated training data files, highlight them and (13) click on \textit{Remove selected point sets from current list} button.

![Image of software interface with numbered steps](image)

Click on (14) the \textit{Facies} tab to view and edit the facies list associated with the input training data.

\textbf{Warning!}

PLEASE, DO NOT modify facies names, unless you are trying to merge multiple training data sets with different facies lists. The analysis depends on facies names to match a training data file to a particular seismic facies, and if it cannot find a matching facies name, it will discard the training data file from the analysis! In case you do need to modify the facies list, you can (15) insert a blank row and then double click on the blank row to define the new facies name.

Click on (16) \textit{Run time parameters} tab to define unique project name and suffix for an analysis. These should be automatically loaded after user browse input training data.

Finally, (17) click on \textit{Parallelization parameters} tab to define MPI parameters (i.e., number of parallel processors).
In the lower section, (18) click on the desired algorithm you want to perform analysis.
For PNN algorithm, there is only one modeling parameter: the smoothing parameter \( r \). Currently AASPI support analysis of a constant \( r \) for all attributes. Specify (19) min, max, and increment values for the analysis. The \textit{pnn\_facies} program allows the user to perform a bulk scaling using the same parameters for all facies or apply different scaling parameters per facies (Ha et al., 2021). For this example, please (20) uncheck the Normalize input training data PER FACIES option. Finally, select the \textit{Execute pnn\_facies button (green arrow)}. 

Program \textit{Analyze supervised algorithms} will always generate the following output files:

<table>
<thead>
<tr>
<th>Output file description</th>
<th>File name syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute combinations</td>
<td>pnn_facies_report_dropout_point_set_combination_#.txt</td>
</tr>
<tr>
<td>program log information</td>
<td>machine_learning_analyze_input_unique_project_name_suffix.log</td>
</tr>
<tr>
<td>program error/completion</td>
<td>machine_learning_analyze_input_unique_project_name_suffix.err</td>
</tr>
</tbody>
</table>

where the values in red are defined by the program GUI. The errors we anticipated will be written to the \*.err file and be displayed in a pop-up window upon program termination. These errors, much of the input information, a description of intermediate variables, and any software trace-back errors will be contained in the \*.log file.

Analysis results are in the form of text reports. Each combination of input attributes per each “dropout” validation point set will have a separate report containing the best error, corresponding to \( r \), as well as the predicted facies and the facies probability values.
Step 5: Create a supervised model

After input training data are analyzed and users know the exact selection of attributes and point sets, they want to use for classification, the next step is to build a model for machine learning. Go to the Machine Learning Toolbox tab and choose (1) **Create a supervised model**.

Next, (2) click on the *Input Training Data* tab. To quickly browse all training data generated by the *Extract user-defined training data* module, click the (3) *Load and append a list from a text file* button, then (4) select the extracted list text file. This list contains all extracted training data files from generate training data step. If you decide to remove some training data files from the list, make sure to click the *Check for missing training data files and number of samples consistency* button. If there is no error, you can proceed. It is not recommended to remove training data files directly this way, but rather removing all training data belonging to a specific attribute or point set.

Next, (5) click on the *Attributes* tab to view the automatically detected attribute list associated with the input training data. You can move attribute up/down in the list. To remove some attributes and all their associated training data files, (6) first highlight them in the list by clicking on them, and then click the *Remove selected attributes from current list* button. **From the results**
obtained during step 4, we know that the best combination of attributes is given by coherence and most-positive curvature ($k_1$) with $r=1.9$. Therefore, please select and remove the other attributes on the list.

If you want to reset to the default attribute list, (8) click on the Rescan Attribute. Finally, select the Automatic transformation type (red rectangle).

Next, (9) click on the Point sets tab to view and define training point sets and validation point sets. For model creation, currently AASPI only supports a (10) user-defined training/validation data distribution where the user will decide which point sets to be used for training and which for validation. To transfer some training point sets to the validation list, (11) first select the training point sets you want to move by clicking on them, then (12) click on the transfer button to swap them to the validation list (and vice versa). A model is created just on that particular training-validation setting. For the PNN algorithm, model creation only depends on training point sets.

To remove a point set and all associated training data files, highlight them and (13) click on Remove selected point sets from current list button.
Finally, (14) change the smoothing parameter value to 1.9 for both attributes in the classification, (15) uncheck the Normalize input training data PER FACIES option, and press **Execute pnn_facies** (green arrow).

Program **Create a supervised model** will always generate the following output files:

<table>
<thead>
<tr>
<th>Output file description</th>
<th>File name syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML model</td>
<td>machine_learning_model_{algorithm_unique_project_name_suffix}.H</td>
</tr>
<tr>
<td>Program log information</td>
<td>machine_learning_create_model_{unique_project_name_suffix}.log</td>
</tr>
<tr>
<td>Program error/completion information</td>
<td>machine_learning_create_model_{unique_project_name_suffix}.err</td>
</tr>
</tbody>
</table>

This model will be an input for the last step of AASPI machine learning workflow: **Classify input data**
Step 6: Classify input data

The last step of AASPI machine learning workflow is to apply a model to the input attribute volumes and classify each voxel. Return to the Machine Learning Toolbox tab, and (1) select Classify input data.

(2) Click to browse to an AASPI-formatted model file. By default, the (3) Unique project name, Suffix, classification type, classification algorithm, and (4) input attribute list are automatically loaded from the model history information.

You can change the seismic attributes if needed. HOWEVER, be careful when changing the input attribute list, as the order of the attributes must be kept the same, and all input attributes must have the same geometric configuration (i.e., same number of samples, crosslines, and inlines). Normally you do not want to modify the detected attribute list, UNLESS you perform training data generation from a cropped version of your data, and now want to apply the model to the entire data. In that case, you can either make a separate text file listing the new input attribute volumes (in correct order) and (5) load it to the table, or (6) modify each item in the list individually and save the modified list for later use.

If you want to reset to the (7) default attribute list, click the Rescan button. Also, select (8) the optional outputs for the algorithm. For this example, we will compute the seismic facies prediction volume by default and the individual facies’ probability volumes.
Click (9) the Vertical Limit tab to define the area of interest.

Next, go to Parallelization parameters tab and (10) set up the desired parallel processing parameters. Finally, press Execute classification (green arrow).
Program **Classify input data** will generate the following output files:

<table>
<thead>
<tr>
<th>Output file description</th>
<th>File name syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>PNN facies prediction</td>
<td>pnn_facies_prediction_&lt;algorithm&gt;_unique_project_name_suffix.H</td>
</tr>
<tr>
<td>PNN facies probability</td>
<td>pnn_facies_probability_&lt;facies&gt;_algorithm_unique_project_name_suffix.H</td>
</tr>
<tr>
<td>PNN facies maximum probability</td>
<td>pnn_facies_max_probability_&lt;facies&gt;_algorithm_unique_project_name_suffix.H</td>
</tr>
<tr>
<td>PNN facies modulated probability</td>
<td>pnn_facies_modulated_probability_&lt;facies&gt;_algorithm_unique_project_name_suffix.H</td>
</tr>
<tr>
<td>Program log information</td>
<td>machine_learning_perform_classification_&lt;pnn_facies&gt;_unique_project_name_suffix.log</td>
</tr>
<tr>
<td>Program error/completion information</td>
<td>machine_learning_perform_classification_&lt;pnn_facies&gt;_unique_project_name_suffix.err</td>
</tr>
</tbody>
</table>

**Exhaustive PNN-based results**

The following results are reported in Lubo-Robles et al. (2019):

We show the results obtained after applying the Exhaustive PNN in all the Eugene Island seismic survey using the coherence and most-positive \( k_1 \) curvature attributes and \( r \) of 1.9. Analyzing the PNN facies prediction co-rendered with the seismic amplitude along inline 391 (Figure 2a),

![Image of PNN classification interface](image-url)
we note that the neural network correctly classifies between salt (red arrow) and nonsalt seismic facies. Also, we compute the PNN Salt probability volume which provide the confidence of the classification (Figure 2b). The algorithm correctly classifies the extracted orange facies as salt with very high probabilities ranging from 75 to 80%. Note that some orange facies visible on the top of the seismic volume are associated with missing or noisy data in the edges of the survey with little interpretational value.

![Figure 2. PNN seismic facies predictions at inline 391 and t = 1.78 s](image)

In Figure 2c, we show the PNN Facies prediction co-rendered with the seismic amplitude volume at time slice 1.78 s. We observe that the Salt 1 (red arrow) diapir is still correctly classified by the Exhaustive PNN algorithm. Moreover, the Salt 2 diapir used as a blind test during the training of our neural network, is also correctly classified by our algorithm as a salt facies (red arrow). Finally, Figure 2d shows that both Salt 1 and Salt 2 diapirs (red arrows) show a high probability of being salt ranging from 75 to 80%.

References
Hajmeer M., and I. Basheer, 2002, A probabilistic neural network approach for modeling and classification of bacterial growth/no-growth data: Journal of Microbiological Methods, 51,


