Is Machine learning the next leap in Subsurface Evaluations

Machine learning: the way forward for geophysical applications

- Introduction

- What is Machine Learning?

- Introduction Neural Networks
    - Input, Hidden, Output layers
    - Nodes and Biases
    - Activation Functions
    - Deep Learning
    - Forward and Backward Propagation

- Applications in Seismic Acquisition, Processing and Interpretation

- Future of Machine Learning in the Geosciences

- Discussion

Introduction

More and more Machine Learning will play a role not only in society in general but also in the geosciences.
In this domain often the word “Algorithms” is used to indicate that computer algorithms are used to obtain results.
Also, “Big Data” is mentioned, indicating that these algorithms need a fast amount of training data to produce useful results

Many scientists mention “Let the data speak for itself” when referring to Machine Learning, indicating that hidden or latent relationships between observations and classes of (desired) outcomes can be derived using these algorithms. Examples are not only in seismic acquisition, processing, and interpretation, but also in the non-seismic domain (gravity, magnetic, EM).

When no clear theoretical model, in the form of equations, can be formulated to describe a geophysical phenomenon, Machine Learning might find useful statistical relationships.

From a range of labelled data, we can derive a linear/nonlinear relationship (model in ML terminology) that predicts the label (supervised learning) of new data (instances in ML terminology).

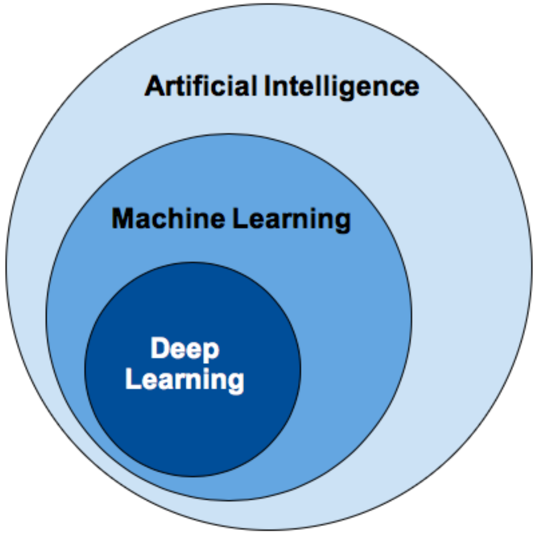
But sometimes it is already useful if an algorithm can define separate clusters, which then still need to be interpreted (unsupervised learning

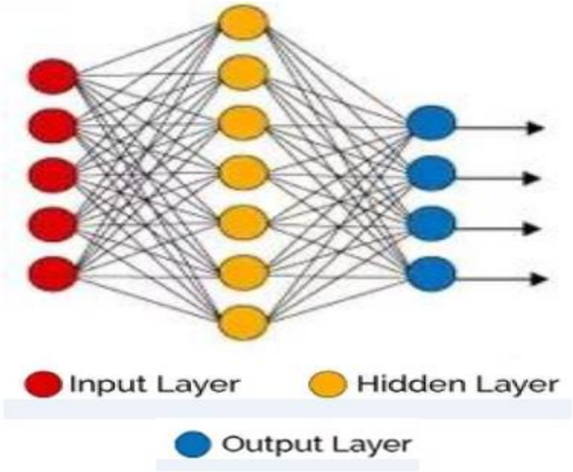
More sophisticated is Semi-supervised learning: labelled and unlabelled data together are clustered whereby the unlabelled data receives the label of the dominant class present in the cluster

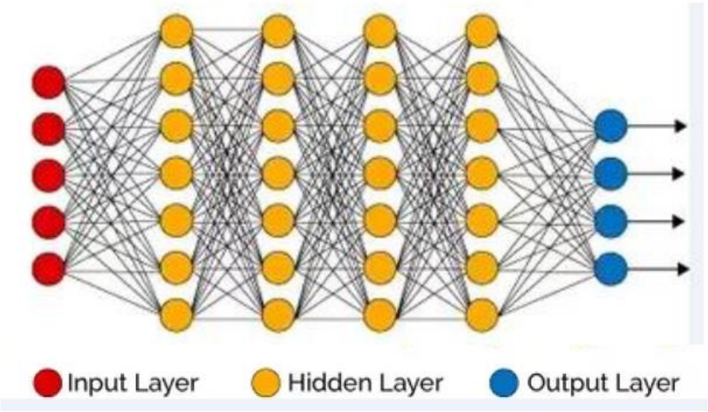
A trained algorithm can be used as a starting point for retraining on a different data set by updating the algorithm

Artificial Intelligence

- Artificial Intelligence > Machine Learning > Deep Learning
- Example of a fully connected Neural Net with 1 hidden layer with 7 nodes
- Each node in the hidden layer
  - Calculates a weighted sum of all inputs
  - Uses an activation function to determine its output to the next layer
- Hyperparameters and parameters
- Machine learning can be used to
  - Classify data using a labelled learning set
  - Cluster data
- In Deep Learning many more hidden layers are used

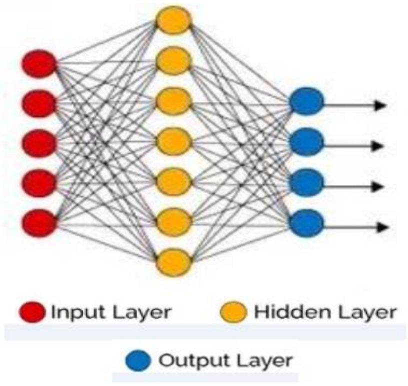


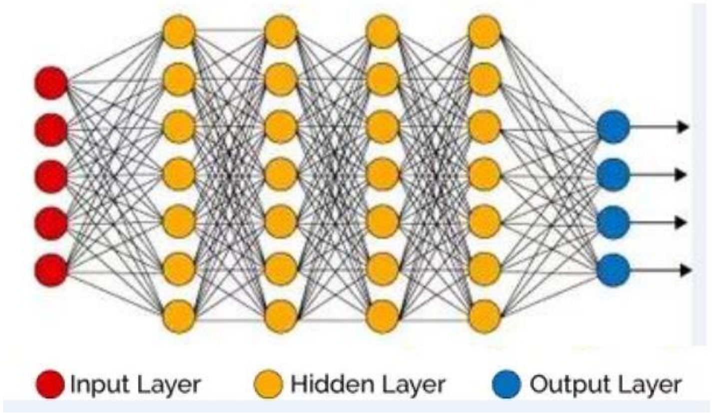




Neural Networks

- Machine Learning Algorithms:
  - Classification & Clustering
  - “Shallow” Learning: Nearest Neighbour, PCA, “you name it”
  - “Deep” Learning: Neural networks, Multilayer Perceptron, “you name it”





- Neural Net with 1 and 4 hidden layers (deep learning)

- Dense network: all nodes are fully connected between neighbouring layers with weights and biases.

- Many Neural Networks exist, for example
  - Convolutional Neural Networks (classification seismic images of different stratigraphic intervals).
  - Transferable Neural Nets: adjusting a network trained on one data set to be able to predict on a different data set.
  - Generative Adversarial Network (GAN): Two neural networks contest with each other. Given a training set, this technique learns to generate new data with the same statistics as the training set (Generator NN), while the other tries to discriminate whether new data is real or fake (Discriminator NN).
  - U-Nets: encoder-decoder: encoder reduces the dimensions of a data set (encoded information), decoder “restores” original data set from encoded form, allows some compute intensive applications to run on encoded condensed data with good results (e.g. FWI)

- How do Neural Networks (Deep Learning) learn?

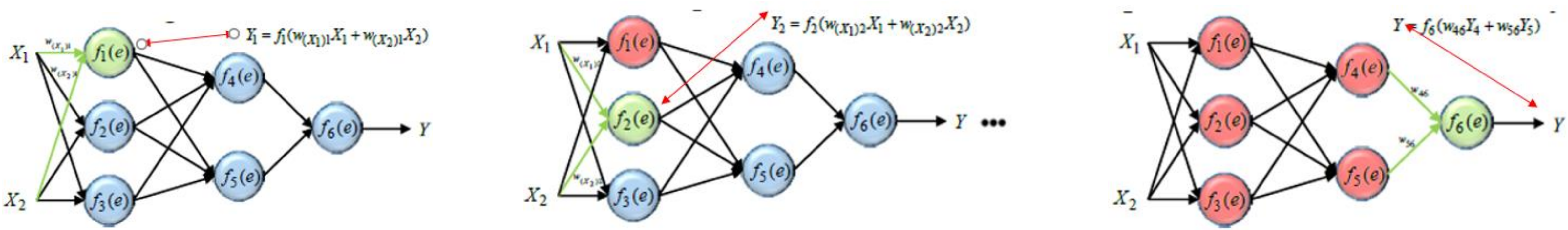
Forward Propagation

Forward propagation: each node receives input from all nodes of the previous layer and provides input to all nodes of the next layer

The weighted attributes are summed. Often also a bias term is added to the weighted sum: $f_1(w_{(X_1)}X_1+w_{(X_2)}X_2 +\text{bias})$.

Forward Propagation:

> For each node the weighted sum of all nodes of the precious layer is the input to an activation function
> This function determines the values propagated to each node in the next layer
> And so on

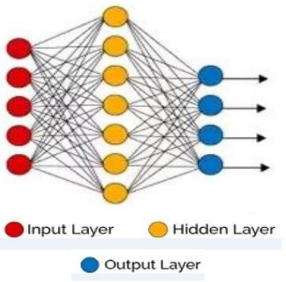


How many parameters (weights & biases) do we need to determine?

It is not uncommon that a NN has up to 1,000 parameters, hence needs up to 1,333 labelled data

Learning using back-propagation

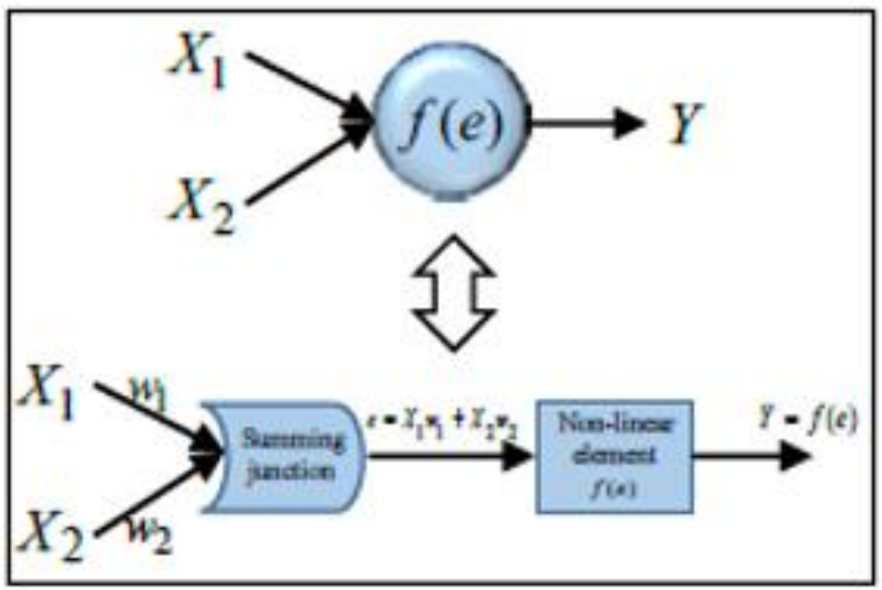
Three layer neural net



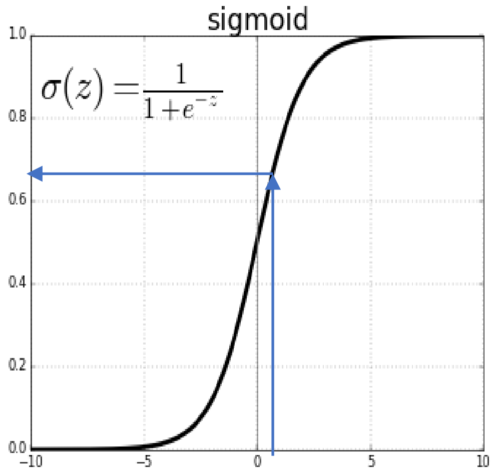
The aim is to find a relationship between the input data and the desired outcome.

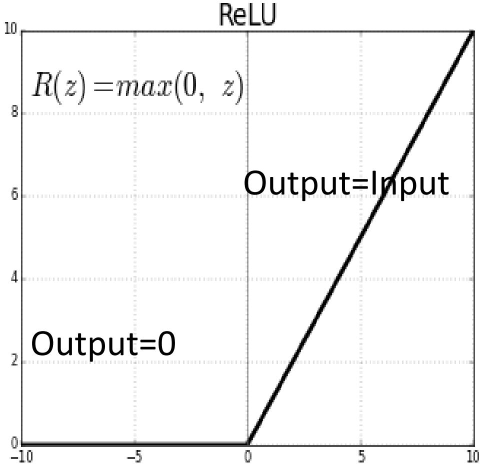
First the network structure needs to be defined (hyperparameters), then the weights and biases (parameters) have to be learned

A neuron consists of 2 parts: the first part adds the products of weighted (w_i) input signals (x_i), the second part determines the output to the nodes of the next layer depending on a non-linear activation function.



Output: y=σ(z)





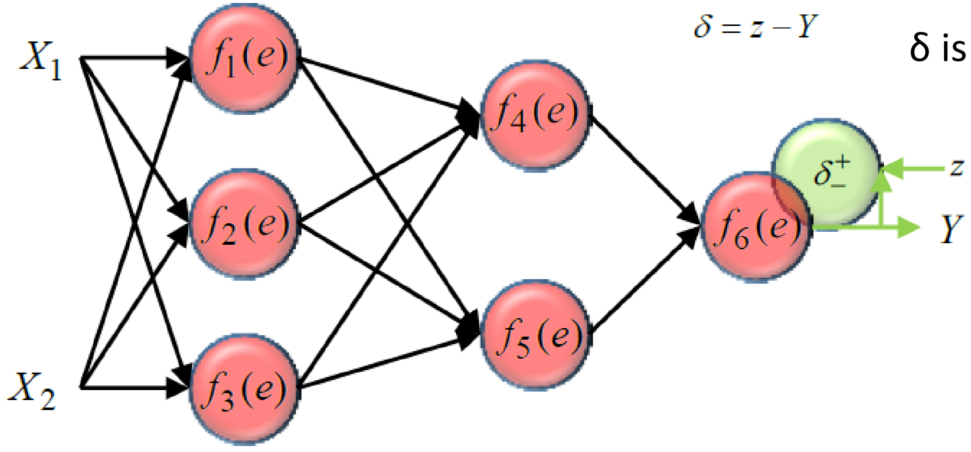
Output=Input

Output=0

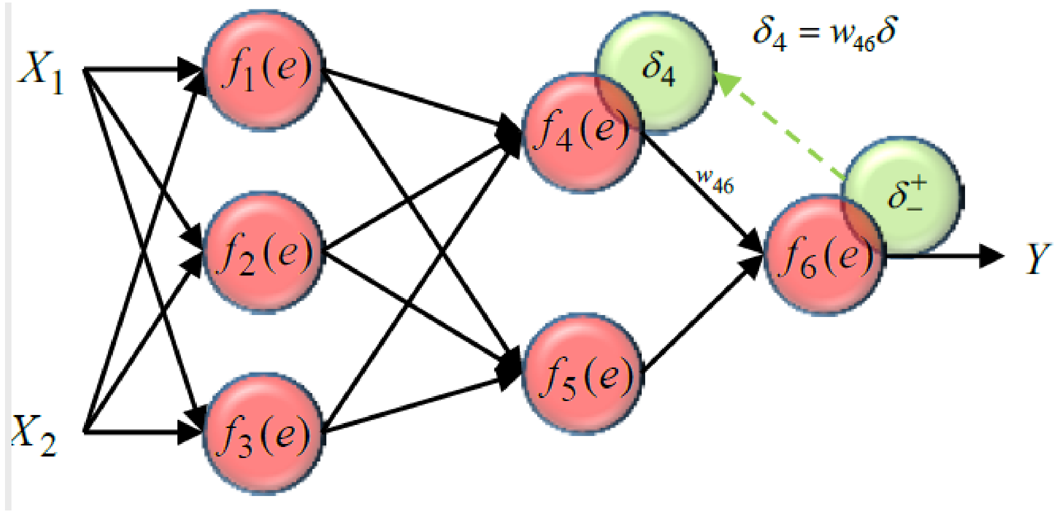
Input: $z = w_1X_1 + w_2X_2$

Calculation of the weights by Backpropagation

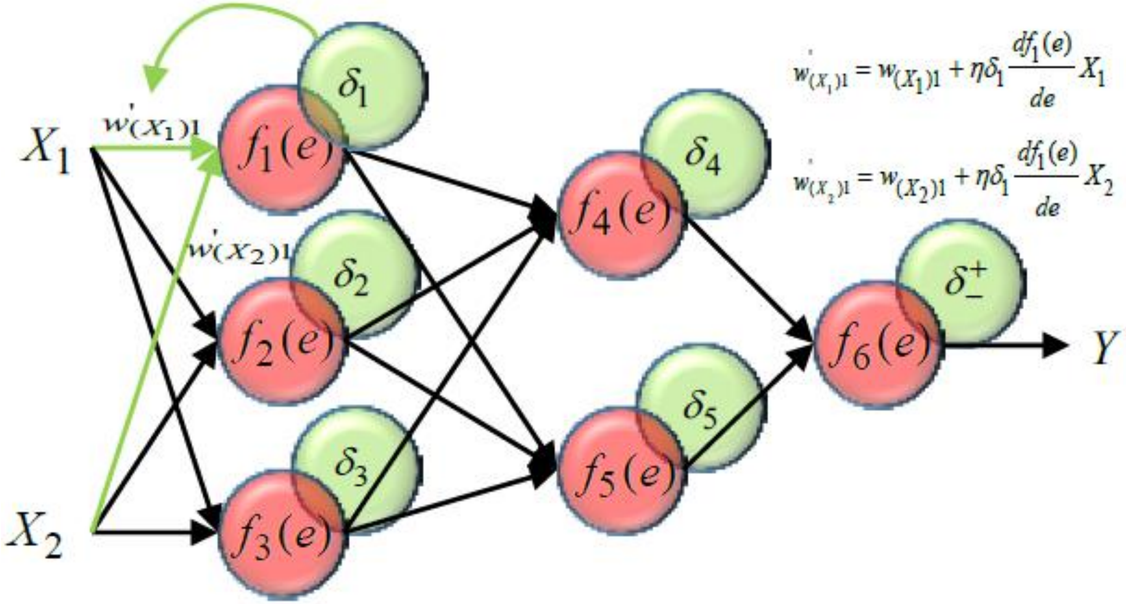
In the first epoch the weights are chosen randomly from [0,1] and the output (z) calculated.
The difference between the calculated (z) and the desired (Y) outcome is the error signal (δ=z-Y) :



δ is then back propagated to each neuron in the previous layer (e.g. $\delta_4 = w_{46}\delta$).



All weights and biases are updated according to:



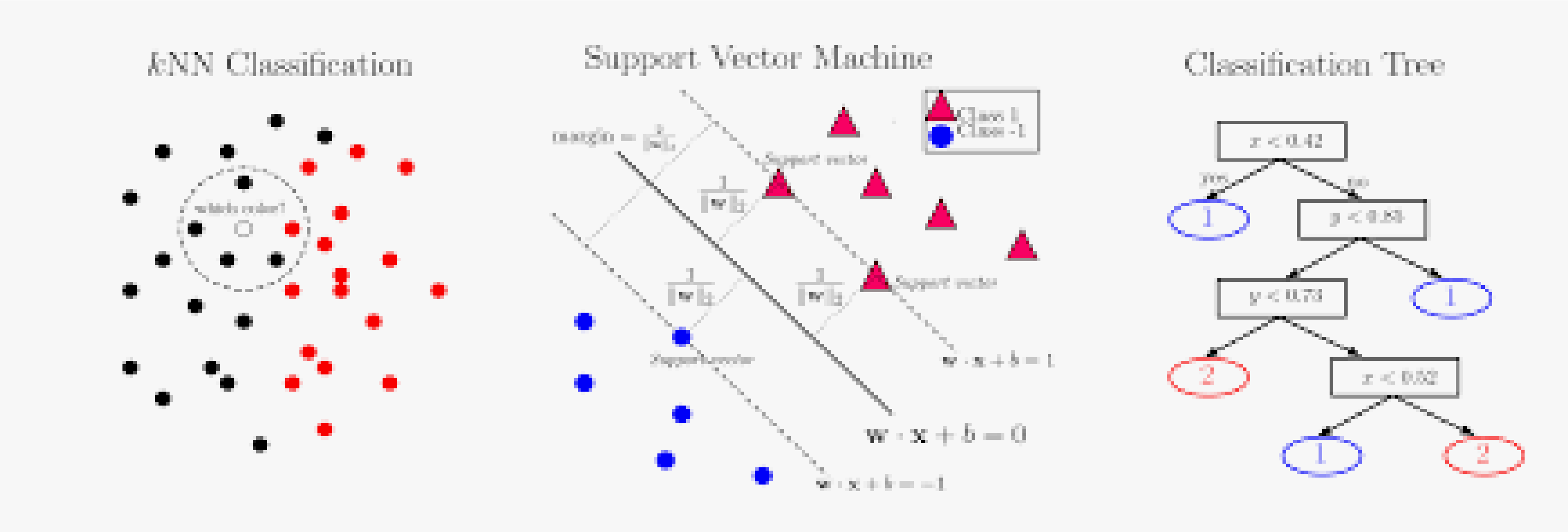
$$w'_{(X_1)1} = w_{(X_1)1} + \eta\delta_1 \frac{df_1(e)}{de} X_1$$

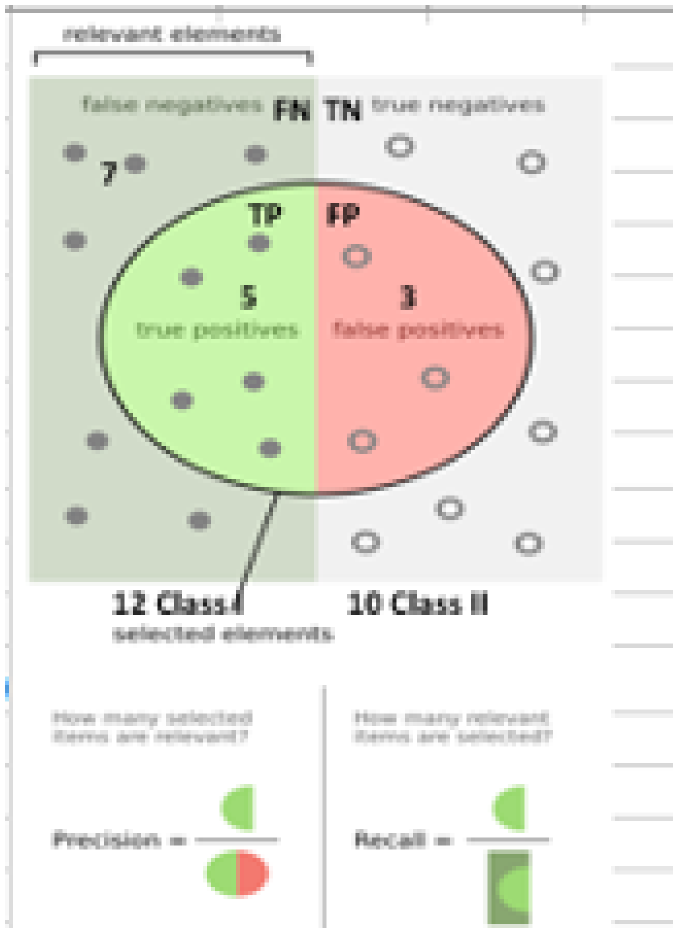
$$w'_{(X_2)1} = w_{(X_2)1} + \eta\delta_1 \frac{df_1(e)}{de} X_2$$

$df_i(e)/de$ is the derivative of the activation function (sigmoid or ReLU) and η is the learning rate, which gets smaller at later epochs

Classification

- Classification is done on the basis of a training set of data containing observations whose class membership is known.
- The individual observations are analysed using a set of properties, known as attributes or features
- Examples of such algorithms are
  - K-Nearest Neighbour
  - Support vector machines
  - Decision Trees

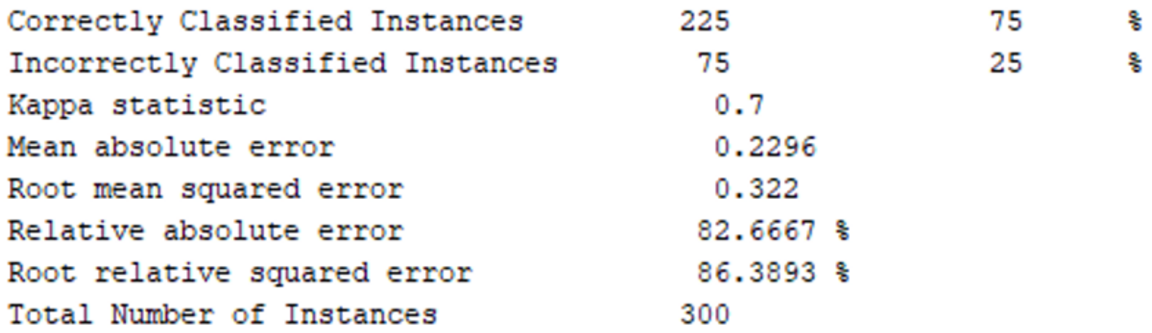




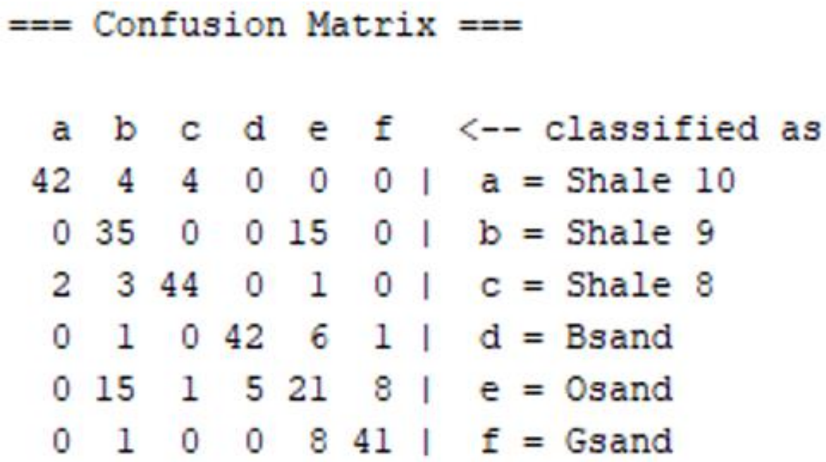
Precision, Recall, F-measure

Precision=TP/(TP+FP)
Recall=TP/(TP+FN)
F-measure=2(P x R)/(P + R)

Class I:
Precision=5(5+3)=0.63
Recall=5/(5+7)=0.42
F-measure=0.25



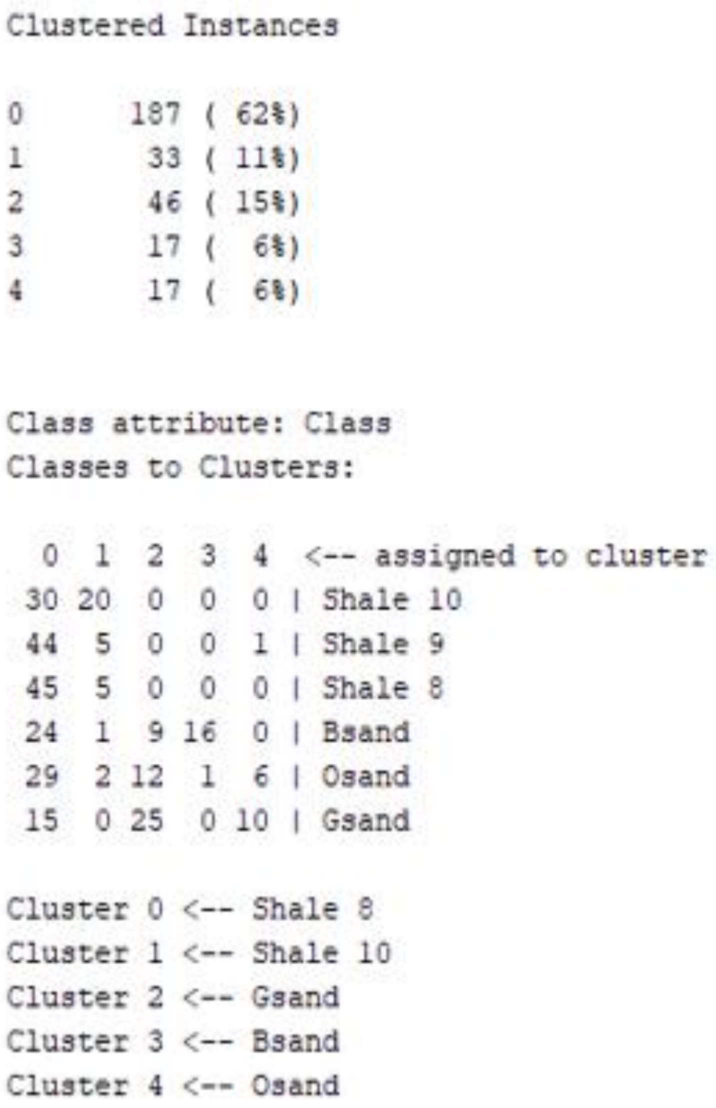
```
Correctly Classified Instances         225               75      %
Incorrectly Classified Instances        75               25      %
Kappa statistic                          0.7
Mean absolute error                      0.2296
Root mean squared error                  0.322
Relative absolute error                 82.6667 %
Root relative squared error             86.3893 %
Total Number of Instances              300
```



```
=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 42  4  4  0  0  0 |  a = Shale 10
  0 35  0  0 15  0 |  b = Shale 9
  2  3 44  0  1  0 |  c = Shale 8
  0  1  0 42  6  1 |  d = Bsand
  0 15  1  5 21  8 |  e = Osand
  0  1  0  0  8 41 |  f = Gsand
```

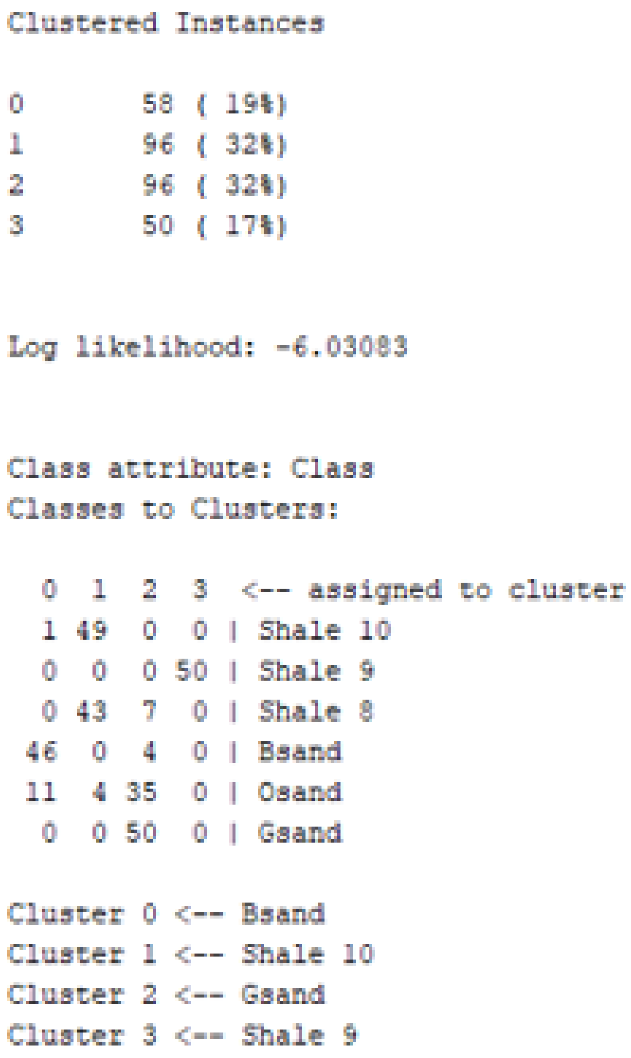
Clustering

Canopy



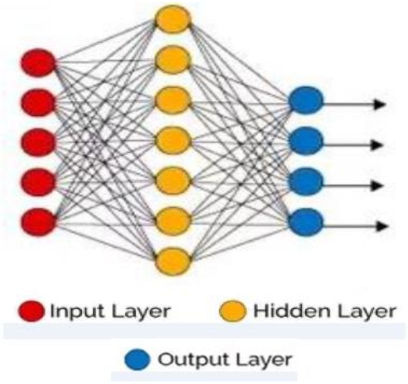
```
Clustered Instances

0       187 ( 62%)
1        33 ( 11%)
2        46 ( 15%)
3        17 (  6%)
4        17 (  6%)


Class attribute: Class
Classes to Clusters:

  0  1  2  3  4  <-- assigned to cluster
 30 20  0  0  0 | Shale 10
 44  5  0  0  1 | Shale 9
 45  5  0  0  0 | Shale 8
 24  1  9 16  0 | Bsand
 29  2 12  1  6 | Osand
 15  0 25  0 10 | Gsand

Cluster 0 <-- Shale 8
Cluster 1 <-- Shale 10
Cluster 2 <-- Gsand
Cluster 3 <-- Bsand
Cluster 4 <-- Osand
```

EM



```
Clustered Instances

0        58 ( 19%)
1        96 ( 32%)
2        96 ( 32%)
3        50 ( 17%)


Log likelihood: -6.03083


Class attribute: Class
Classes to Clusters:

  0  1  2  3  <-- assigned to cluster
  1 49  0  0 | Shale 10
  0  0  0 50 | Shale 9
  0 43  7  0 | Shale 8
 46  0  4  0 | Bsand
 11  4 35  0 | Osand
  0  0 50  0 | Gsand

Cluster 0 <-- Bsand
Cluster 1 <-- Shale 10
Cluster 2 <-- Gsand
Cluster 3 <-- Shale 9
```

Geophysical Applications

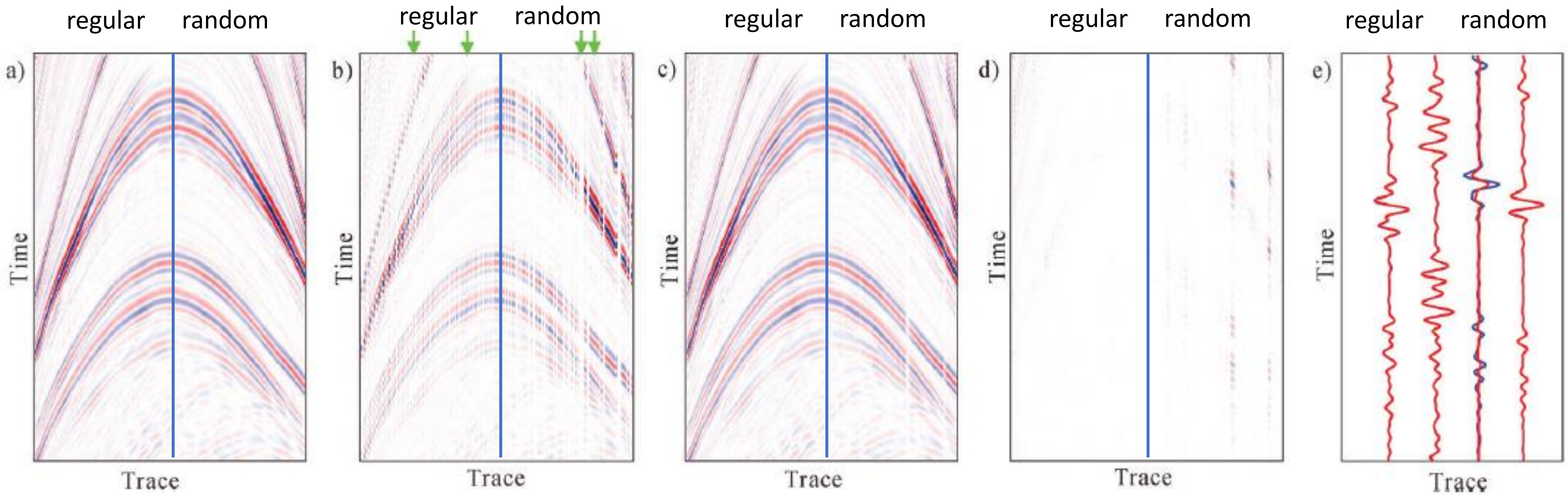


- Machine Learning Applications for Seismic data:
  - Interpolation missing acquisition traces
  - First Arrival picking
  - Ground-roll attenuation
  - Data reduction (encoder-Processing/FWI- decoder NN)
  - Stratigraphic Interpretation
  - Reservoir Characterisation
  - ………

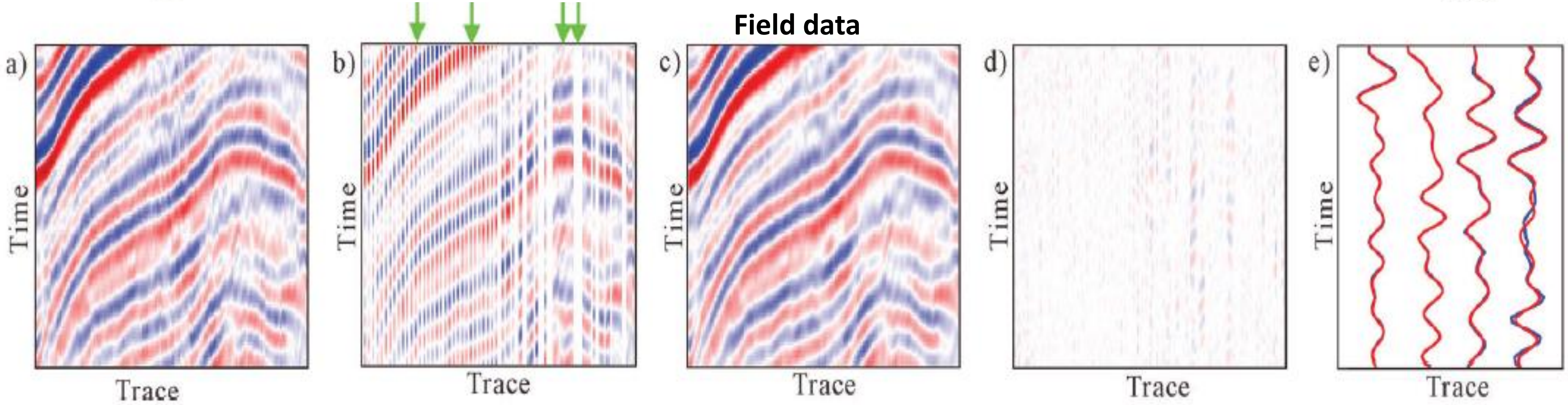
- Machine Learning Applications for Non-Seismic data:
  - Interpolation missing data
  - Data reduction (encoder-Processing/Inversion- decoder NN)
  - Classification of areas based on Satellite Spectral response
  - Determination of different ore bodies using Gravity, Magnetics and IP
  - …….

- Machine Learning Applications for Well data:
  - Interpolation missing data
  - Prediction of missing (shear) logs
  - Prediction of facies
  - …….

Seismic Acquisition: Trace Interpolation

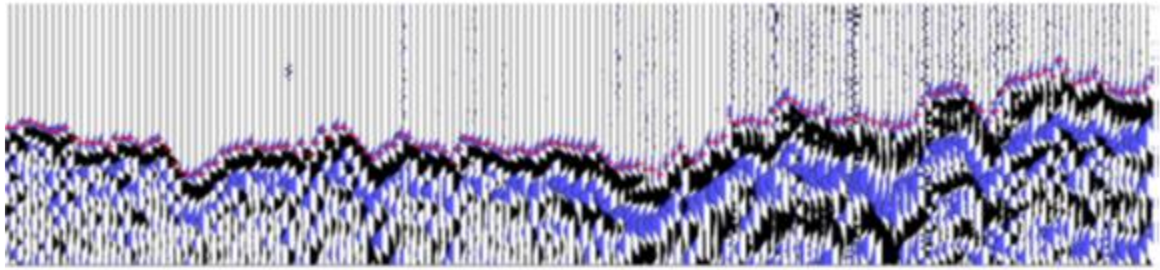


Field data

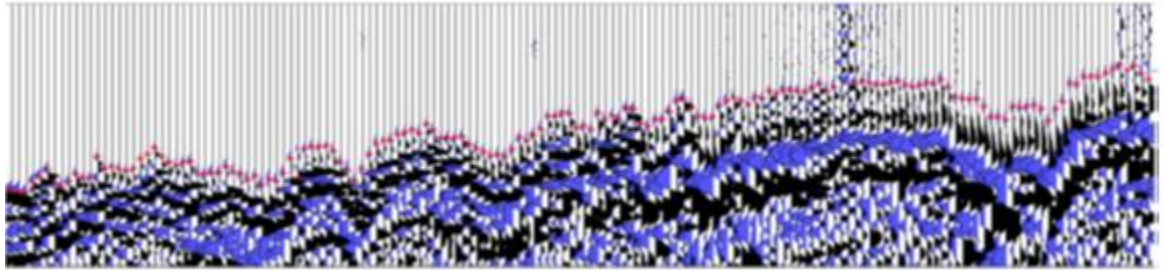


Notice: poorer interpolation for large gaps (on the right) and for steep dips

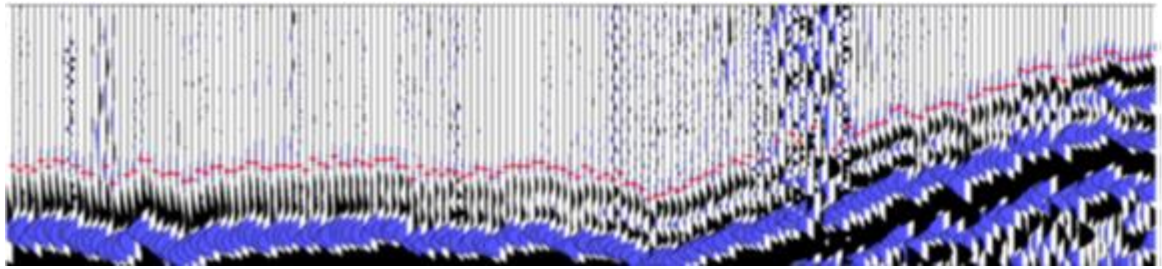
Seismic Processing: First arrival picks using U-nets



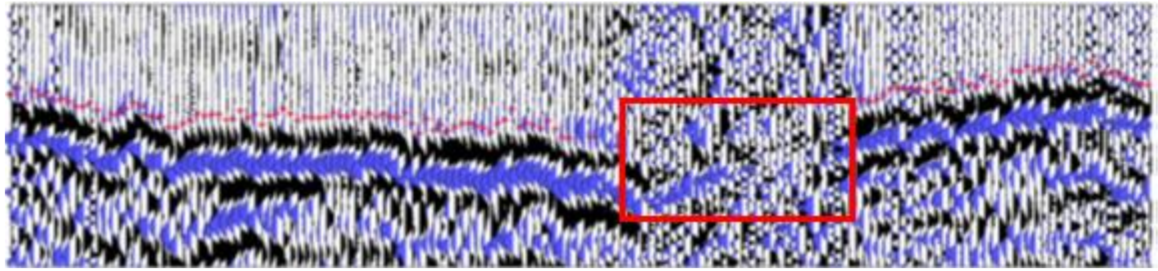
High quality data



Poor quality data



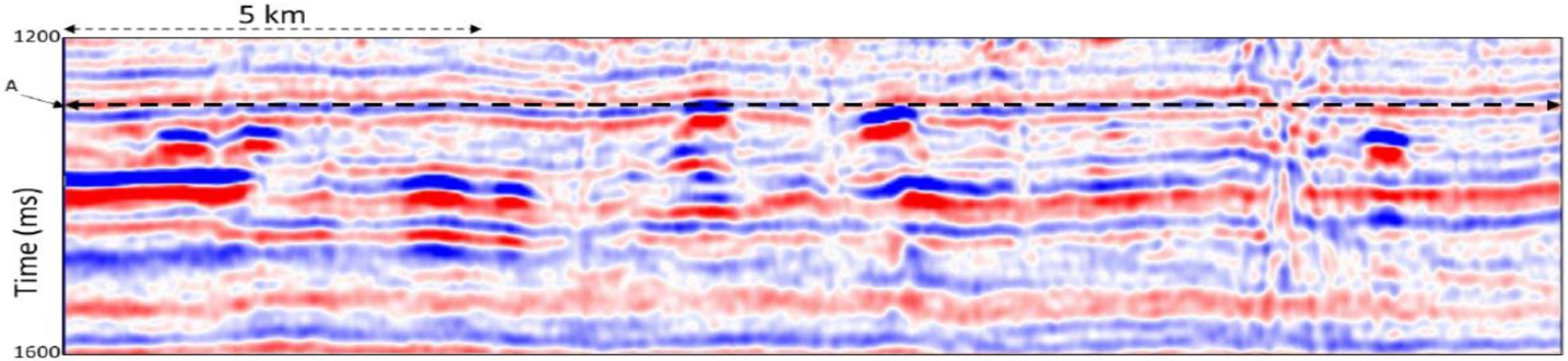
Poor quality data



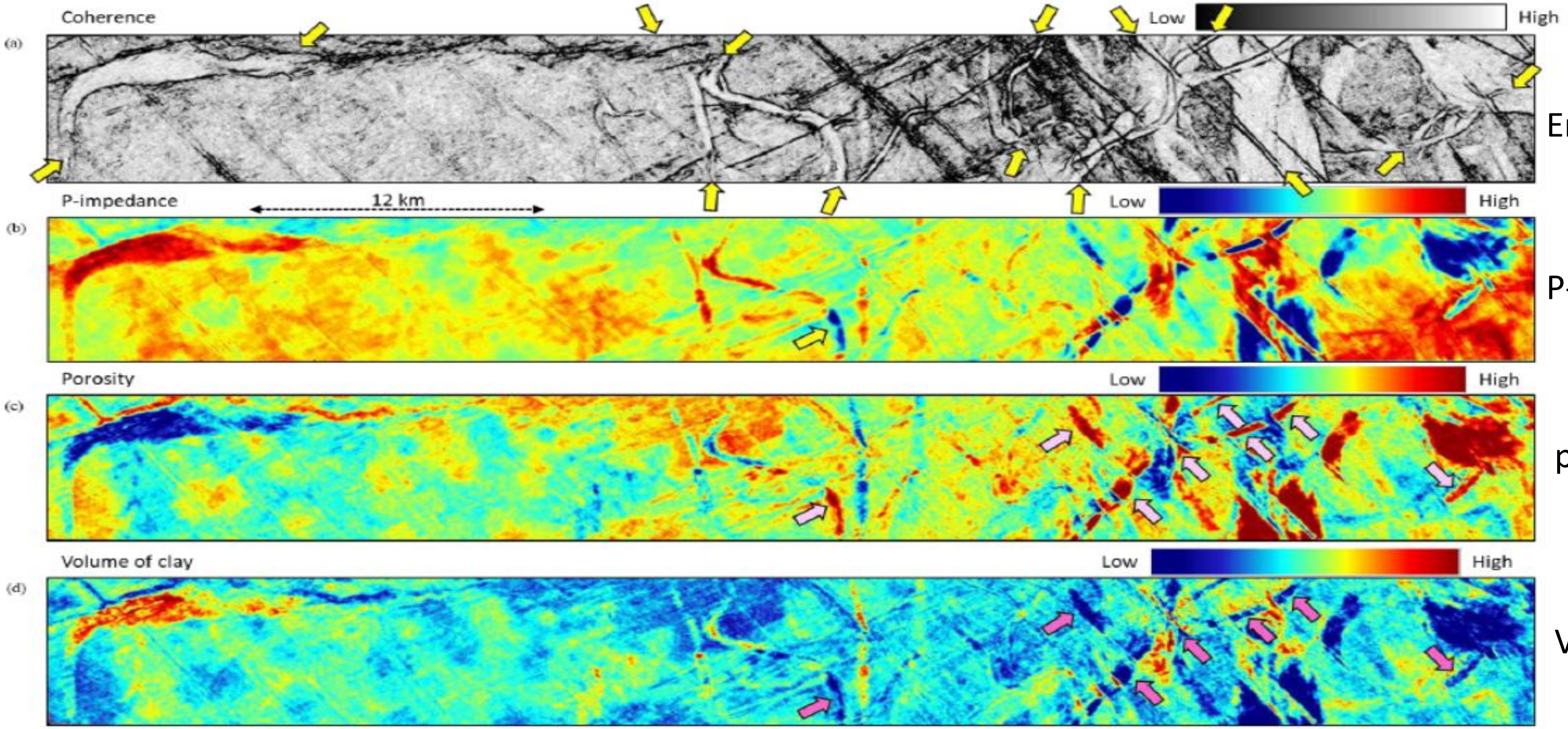
Poor quality data

Picks that can not be recognized by U-net are classified as non-breaks (see red box)

Data



time slice at 1280 ms



Faults & Channels

Energy-ratio coherence

P-impedance

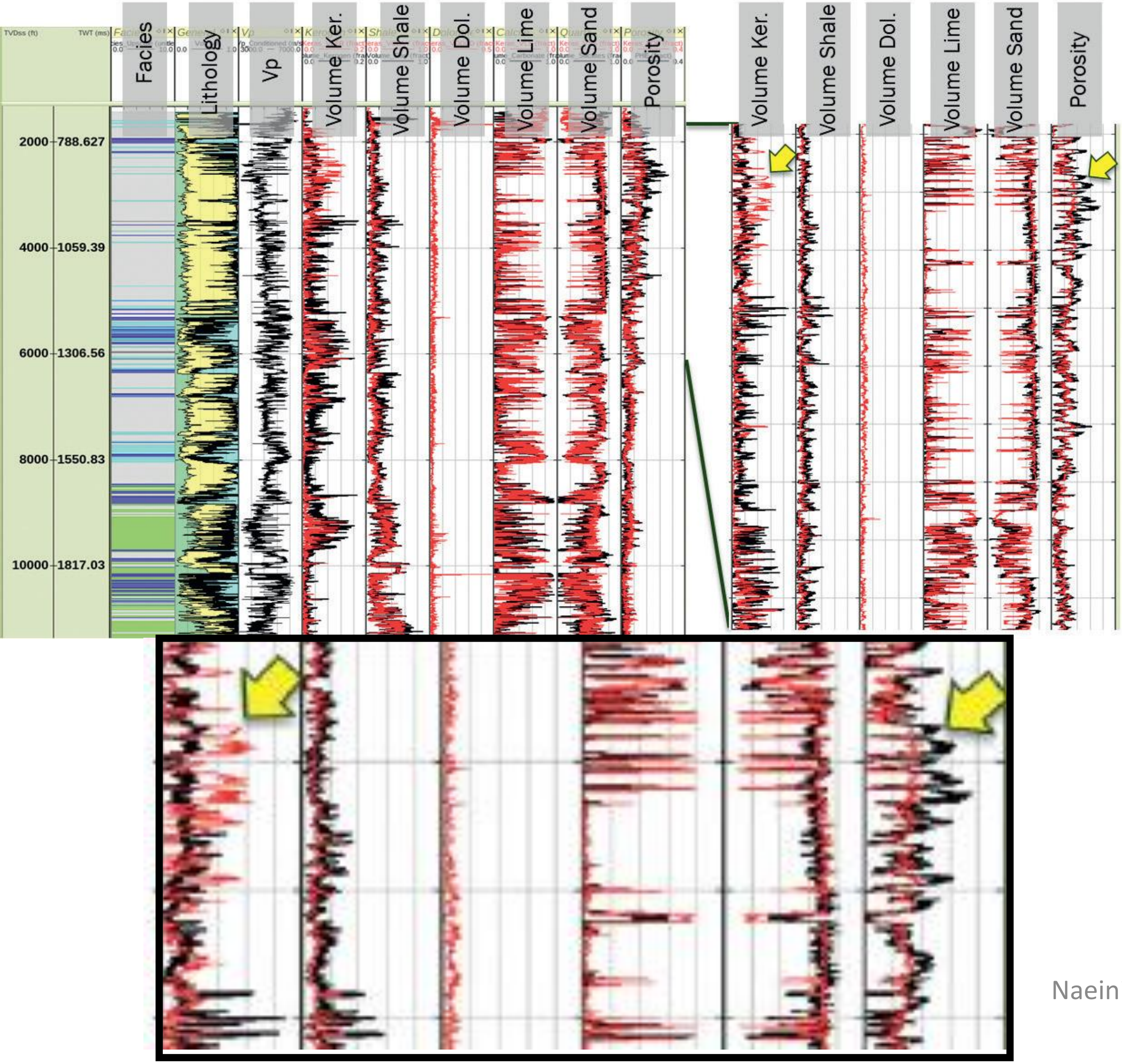
porosity

V_{clay} volume

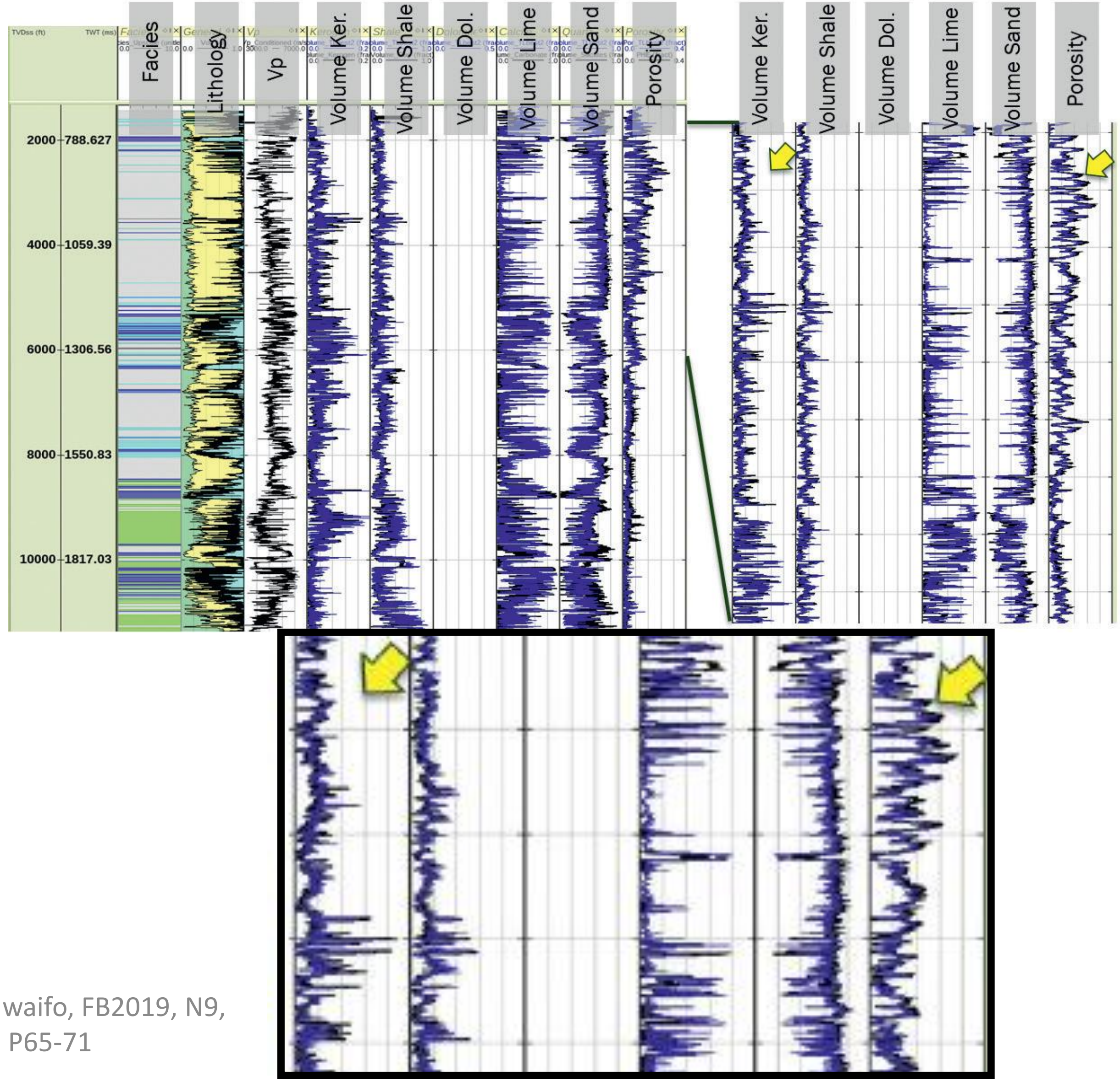
Chopra & Marfurt, SEG2018, P2056-2060

Blind well predictions: Manual versus DNN versus Transfer learning

DNN model without Transfer Learning



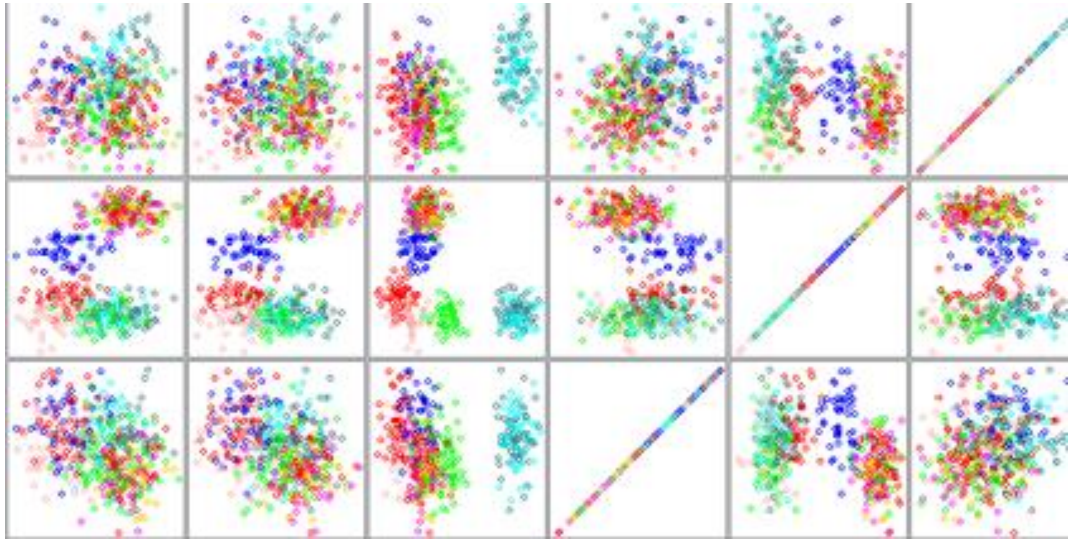
DNN model with Transfer Learning

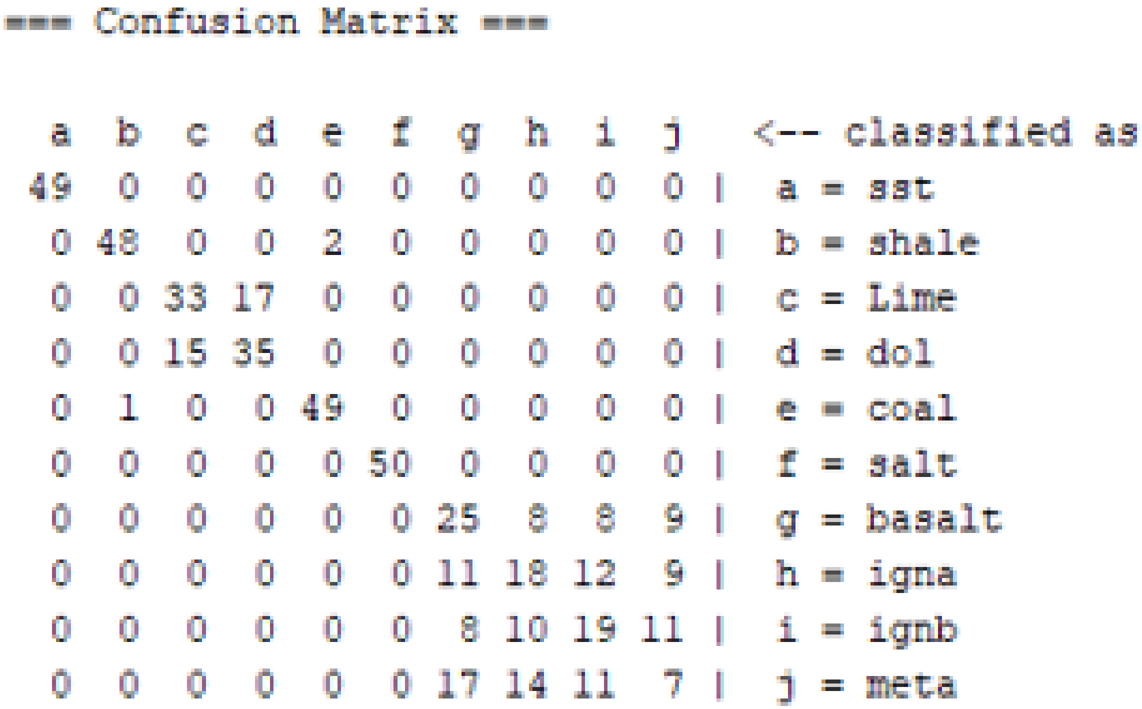


Naeini & Uwaifo, FB2019, N9, P65-71

Machine Learning for Non-seismic data

- For 10 rock types 8 measurements were chosen (2 seismic & 6 non-seismic) for classification





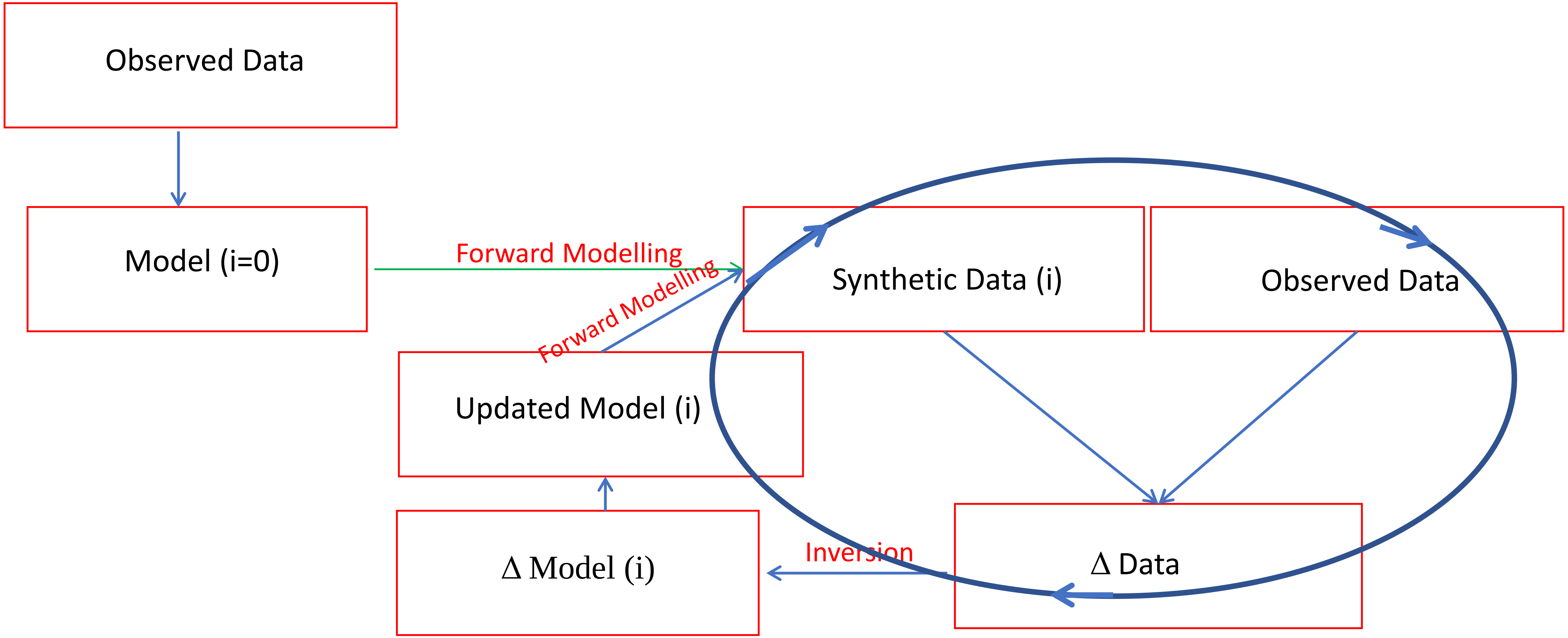
```
=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j   <-- classified as
 49  0  0  0  0  0  0  0  0  0 |  a = sst
  0 48  0  0  2  0  0  0  0  0 |  b = shale
  0  0 33 17  0  0  0  0  0  0 |  c = Lime
  0  0 15 35  0  0  0  0  0  0 |  d = dol
  0  1  0  0 49  0  0  0  0  0 |  e = coal
  0  0  0  0  0 50  0  0  0  0 |  f = salt
  0  0  0  0  0  0 25  8  8  9 |  g = basalt
  0  0  0  0  0  0 11 18 12  9 |  h = igna
  0  0  0  0  0  0  8 10 19 11 |  i = ignb
  0  0  0  0  0  0 17 14 11  7 |  j = meta
```

- From the confusion matrix:
  - Sandstone and shale were classified almost perfectly
  - Limestone and Dolomite were difficult to separate
  - Coal and Salt were easy to separate
  - Basalt, igneous and metamorphic rocks were difficult to separate

- On the whole, the confusion matrix has a dominant main diagonal

Geophysical Inversion

How to obtain the subsurface properties from the observed data



Observed Data

Model (i=0)

Forward Modelling

Synthetic Data (i)

Observed Data

Forward Modelling

Updated Model (i)

Δ Model (i)

Inversion

Δ Data

Future of ML in Geosciences

Machine Learning/Artificial Intelligence will certainly not be the “silver bullet”

ML will be of great “assistance” to the geoscientist by finding, hidden or latent “relationships” between data and objectives

ML will be able to cluster and classify data sets “objectively” (“consistently”)

When a clear and well understood physical model (equations) exists Inversion is the tool, when not it could be Machine Learning

Also, knowledge can be stored in algorithms, which then are the so-called shoulders on which new generations of geophysicist/geoscientists can stand and continue to expand the geosciences

This presentation was sponsored by www.breakawaylearning.nl