dGB Earth Sciences

ML Fault Prediction Challenge

July 2021

Challenge

Can we improve Machine Learning Fault Prediction in OpendTect?

Current best results:

1. Pre-process the input seismic; favorite filter is Edge Preserving Smoother (EPS) in Faults & Fractures plugin
2. Apply pre-trained 3D Unet Fault Predictor

Pros:

- Very fast application (especially on GPU)
- Large-scale faults are better delineated than Thinned Fault Likelihood (TFL)

Cons:

- Dependency on pre-processing filter (which could potentially introduce artifacts?)
- Results cannot be improved with manual interpretations (needs 3D examples of 128x128x128 samples)
- Does not predict fault likelihood dip and azimuth which implies that output cannot be thinned like TFL; Output is currently thinned in Faults & Fractures with a (slightly inferior) Skeletonization algorithm

Boundary Conditions New Workflow

1. No pre-processing (or only a mild filter such as a dip-steered median filter with small stepout)

2. Human interpreter can create new examples to train (new, transfer, continue) a model

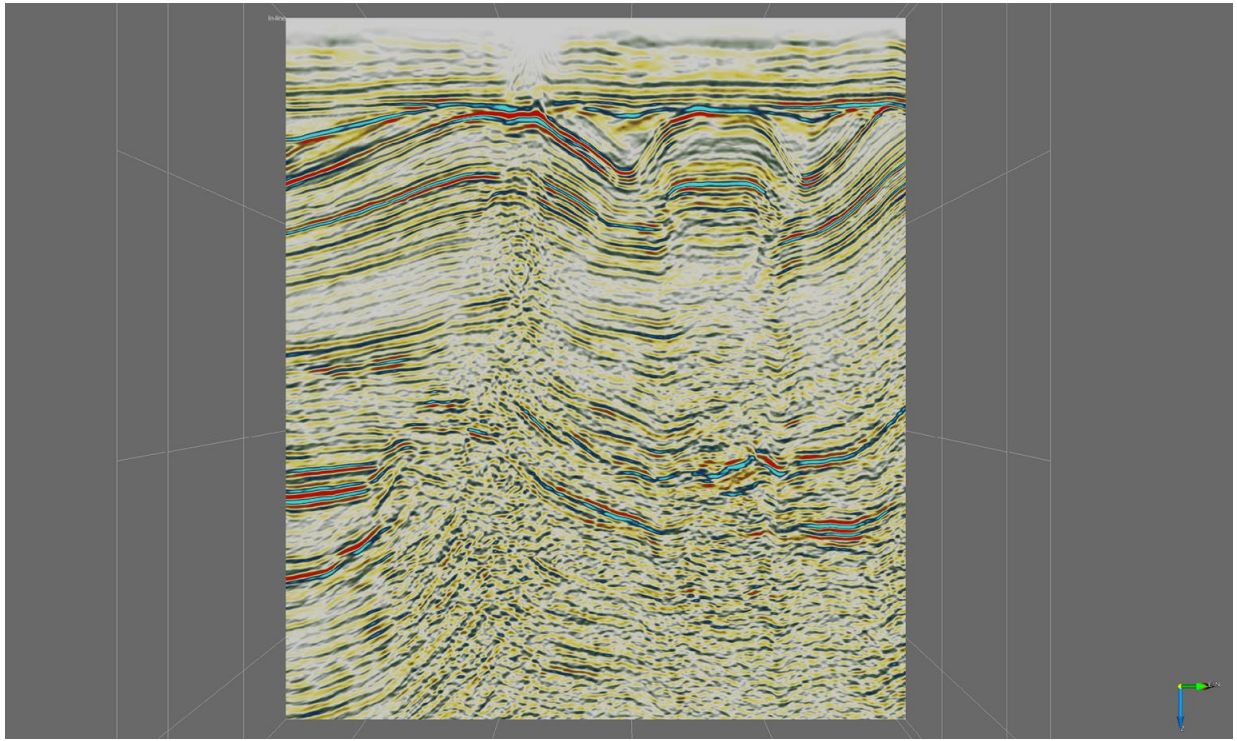
Regarding 2) we can already do this in OpendTect but before we improve the workflow (make it user-friendlier & more interactive) we want to see prove that this generates better results than the current best workflow

Next slides

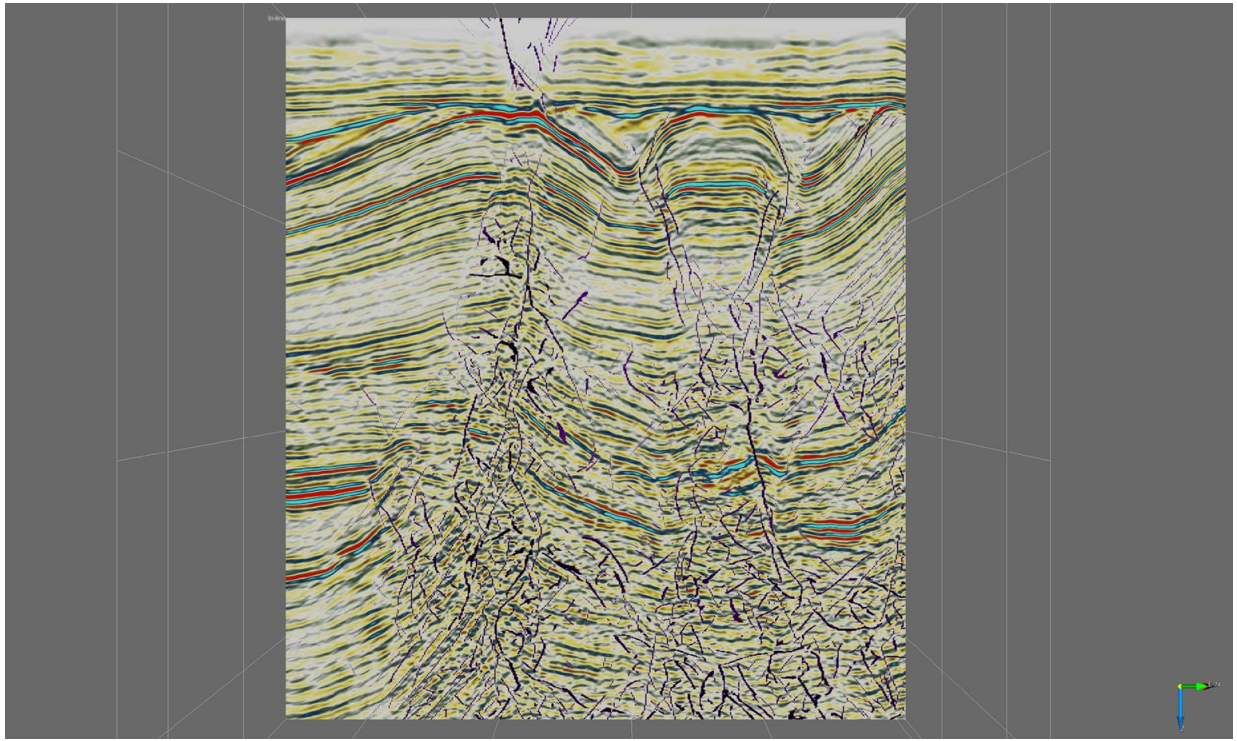
Fault Likelihood predictions on Delft

a) line 2275
b) TFL (defaults)
c) pre-trained 3D Unet applied to normal seismic (This 128x128x128 Unet is the pre-trained Fault Predictor that is shipped with the software. It is trained on synthetic data)
d) same pre-trained 3D Unet applied to EPS
e) pre-trained 2D Unet applied to normal seismic (This Unet is trained on regular and flipped 2D sections (128x128) extracted in the inline directions from the same synthetic dataset)
f) same pre-trained 2D Unet applied to EPS

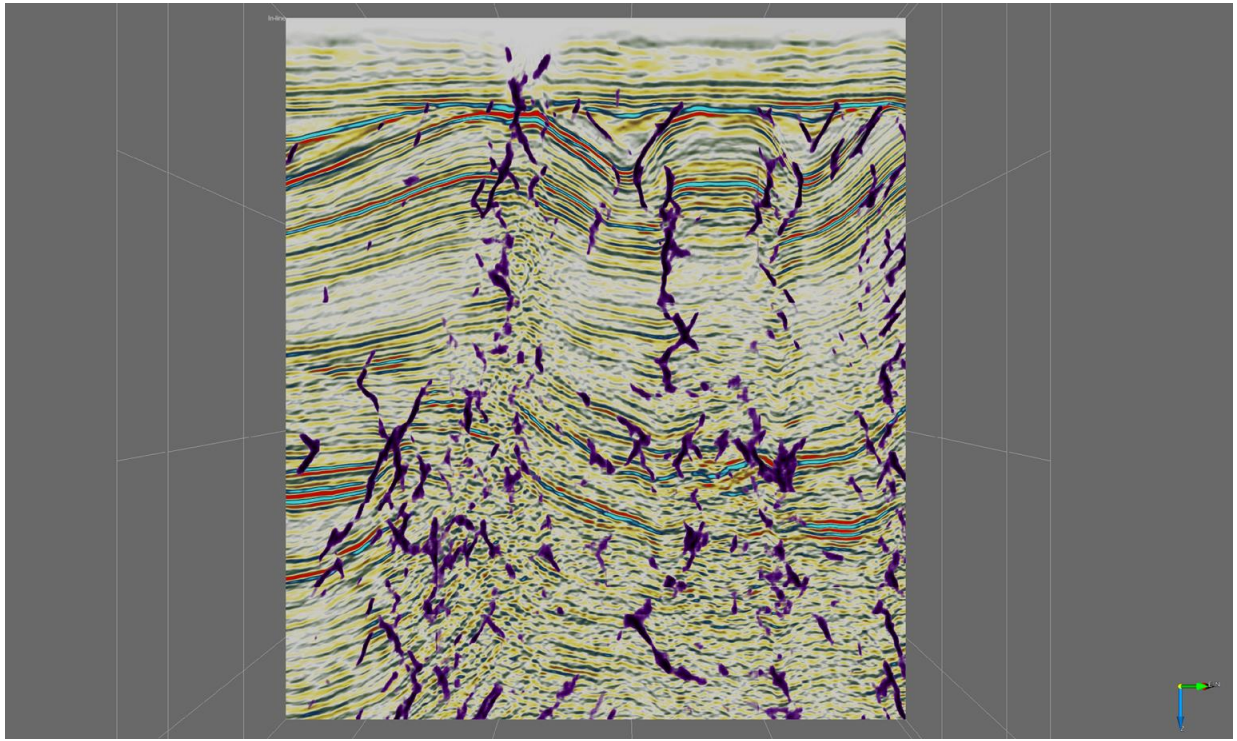
a) Seismic line 2275



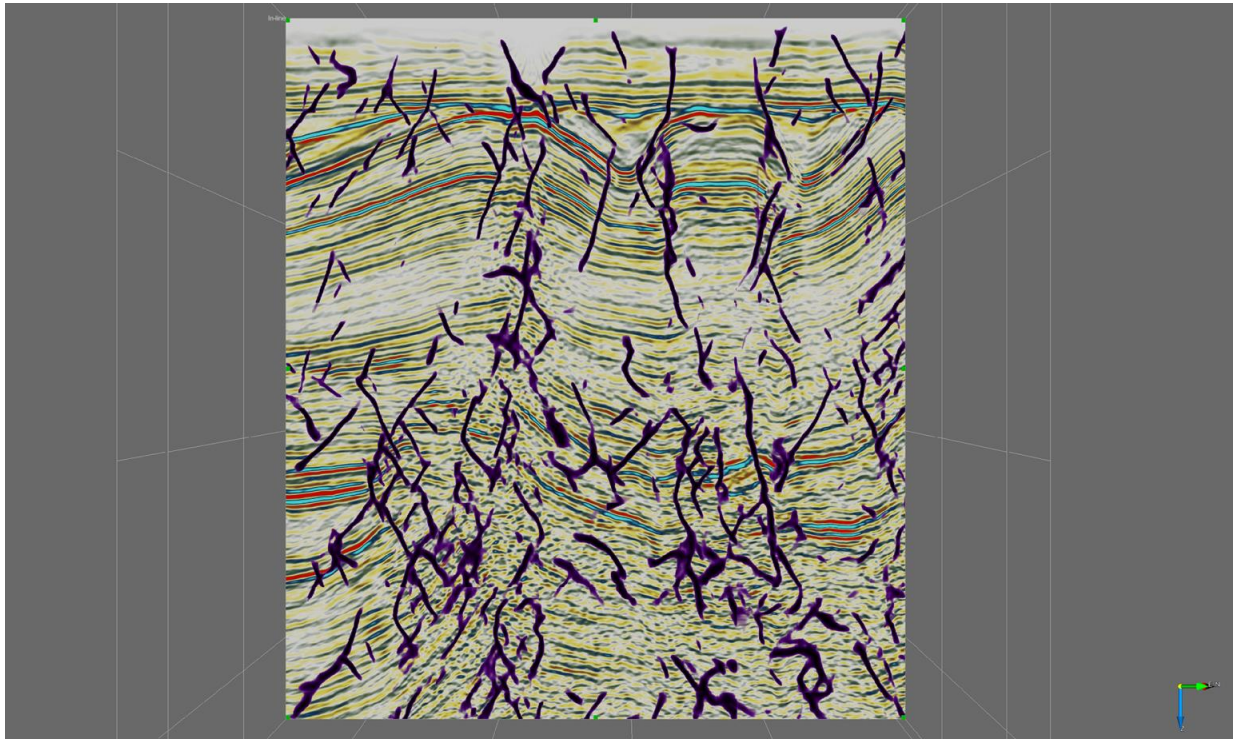
b) Thinned Fault Likelihood



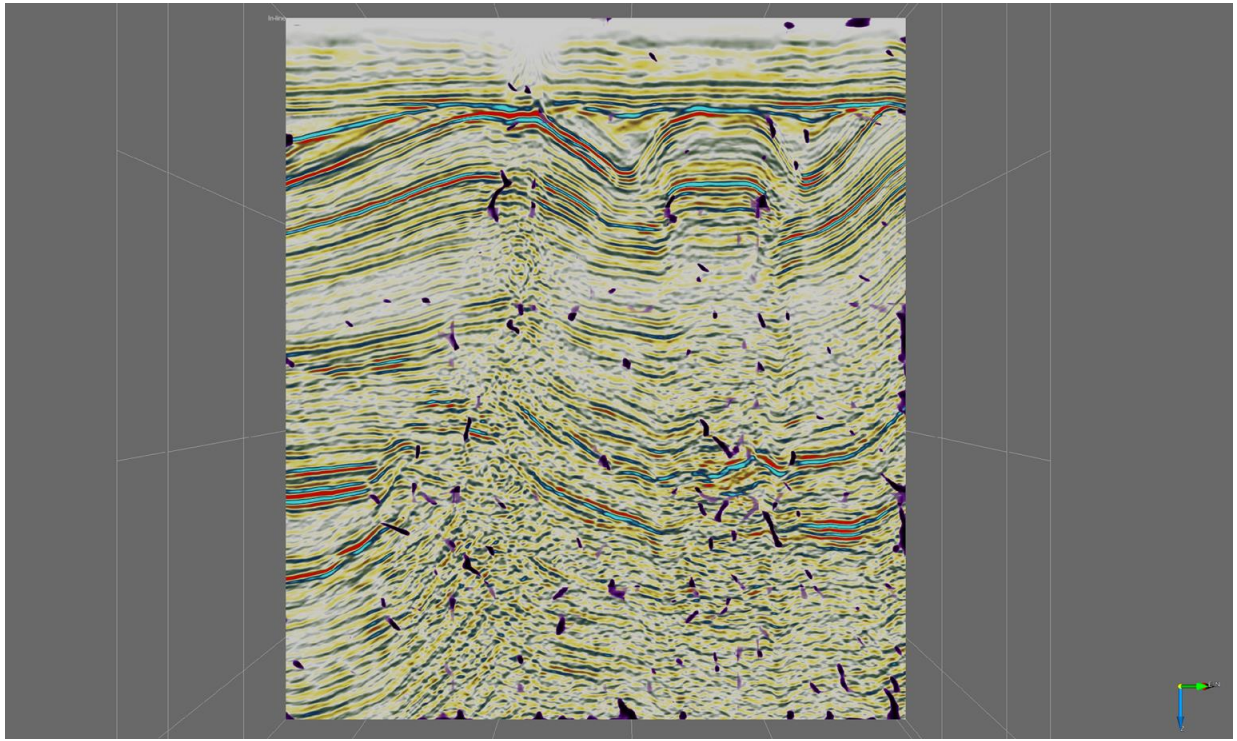
c) 3D Unet Fault Likelihood from seismic



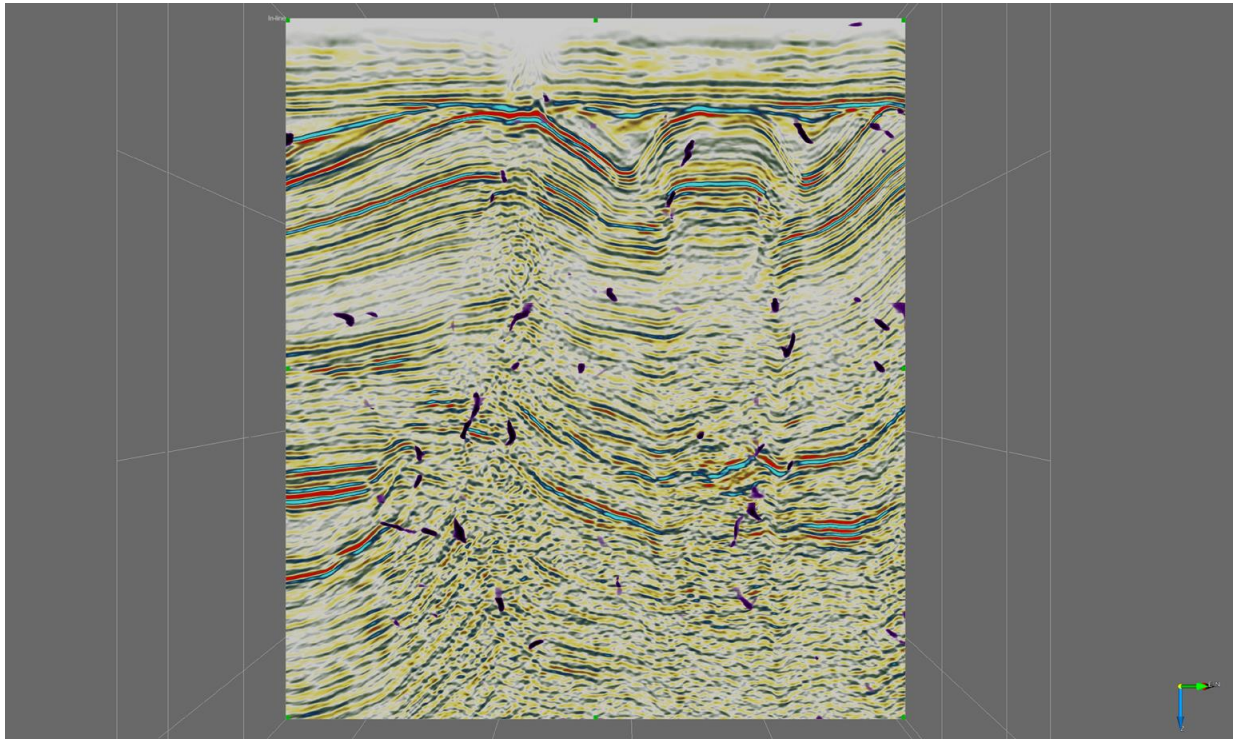
d) 3D Unet Fault Likelihood from EPS



e) 2D Unet Fault Likelihood from seismic



f) 2D Unet Fault Likelihood from EPS



What can you do?

- Help us with tips on how we can develop a workflow that predicts faults without pre-processing and which can be tuned by feeding new examples
- Or, create your own solution to beat the result in slide 8
- Decide yourself if you wish to share this solution for free or under your own commercial terms, or keep it for yourself (and your company)

What is in it for you?

- Your advice and/or code improvements lead to a better, user-friendlier, and more stable environment benefitting the entire community

- Trained models added to the library of trained models we are building can be used by other workers either:
  - For free meaning you get credits for your research and your work can be used directly in operational settings
  - Under your own commercial terms meaning you can get whatever the market is willing to pay for your work

How to start?

- Download the free “Delft” dataset from TerraNubis. This dataset does not check for license keys, hence can be used for free Machine Learning experiments

- Xingmin Wu’s 3D synthetic seismic examples (128x128x128) with fault masks can be downloaded from: https://github.com/xinwucwp/faultSeg (with a bit of manipulations you can also use these examples for training a 2D model)

- If you go for a solution involving a 2D model, you can create new examples using the workflow described on the next slide

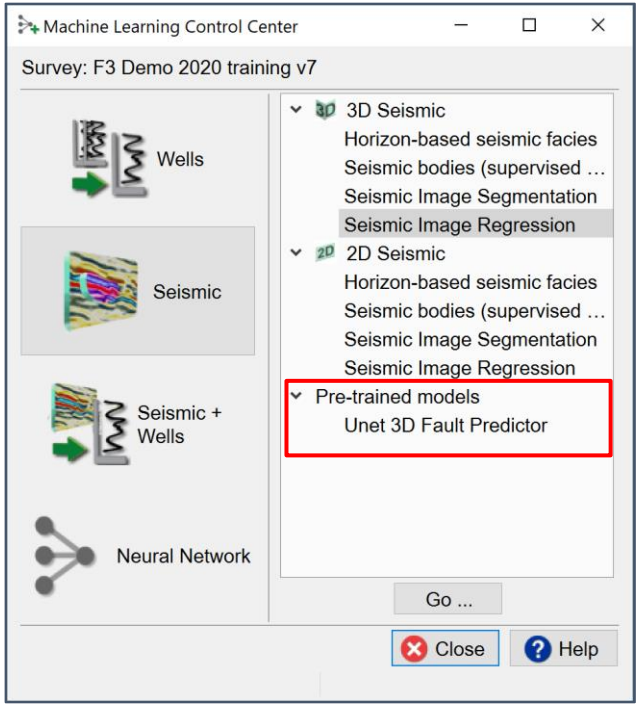
How to create 2D (or pseudo-3D) examples?

- Interpret fault sticks on a limited number of inlines
- Use the Volume Builder Fault Painter module to create masks from interpreted sticks on a line-by-line basis
- Use the Seismic Image Segmentation workflow to create a training set from all interpreted seismic lines and the corresponding fault masks
- Save the selected data so that you can easily retrieve it in case you want to add more interpreted lines in a subsequent iteration
- To create 2D examples specify 0xNxN samples to extract examples along inlines
- For pseudo-3D modify the HDF5 formatted 2D example set yourself (copy N times, shift & change as you see fit; if N=128, the training set can be used for transfer training on the pre-trained “Unet 3D Fault Predictor”, see next slide)

How to recreate the best results so far (slide 8)?

- Launch the Faults & Fractures control center and apply an “Edge Preserving Smoother” to the seismic data (OpendTect Pro Plugins User Documentation Chapter 2.2.4)

- Launch the Machine Learning control center and apply the “Unet 3D Fault Predictor” to the smoothed data



Questions?

Please ask “Paul de Groot (dGB)” on the Discord server of the OpendTect ML Developers (ODML) Community

To join ODML: https://discord.gg/9cVrW2sNza