SPE-202930-MS

Application of Machine Learning-Assisted Fault Interpretation on Large Carbonate Field with Subtle Throws

Mujeeb Oke, ADNOC; Samuel Tilley and Surender Manral, Schlumberger; Mohamed Tarek Gacem, Arwa Mawlod, and Khadijah Al Daghar, ADNOC; Houcine Ben Jeddou, Hussein Mustapha, and Cen Li, Schlumberger

This paper was prepared for presentation at the Abu Dhabi International Petroleum Exhibition & Conference to be held in Abu Dhabi, UAE, 9 – 12 November 2020. Due to COVID-19 the physical event was changed to a virtual event. The official proceedings were published online on 9 November 2020.

Abstract

We illustrate novel application of Machine Learning (ML) assisted fault interpretation in the Middle East, interpreting complex fault structures with subtle throws in a large carbonate field onshore Abu Dhabi. Introduced as part of an integrated multi-disciplinary digital excellence initiative at ADNOC, ML-assisted fault interpretation seeks to overcome historic operational bottlenecks caused by traditional seismic interpretation methods which are slow, labour intensive, repetitive, and subjective. Core objectives for deploying ML-assisted fault interpretation were to reduce evaluation time, improve interpretation accuracy, and ensure integration across an intelligent evaluation ecosystem comprised of various disciplines. Envisaged gains from deploying ML-assisted fault interpretation methodology included effective and efficient utilization of multiple seismic datasets to drive rapid multi-scenario analysis, leading to better subsurface understanding within much shorter time frames.

Input data used of the project was standard amplitude volume with minimal user-end conditioning. PSTM time and PSDM depth seismic volumes were used in separate runs to confirm that applied ML technology is domain agnostic. The ML-Assisted workflow included: Generating a fault prediction cube based on user-supplied fault interpretation labels made on 6 training lines (<0.8% of the available lines); Creation of fault planarity and azimuth cubes; Parameterization of automated extraction function; Extraction of segmented 3D fault pointsets; Creation of fault framework and fault sticks that can be integrated into traditional methods in seismic and geological modelling domains.

Despite limited fault displacement apparent on the seismic volumes, ML fault predictions were of high quality, closely adhering to the seismic response as guided by user-provided training samples. Advantages envisaged from use of ML-assisted interpretation technology in the project were fully realized as the technology enabled rapid extraction of complicated fault structures within a fraction of the time and effort previously taken using traditional means. Efficiency and precision gains from using ML-assisted fault interpretation presents benefits that single seismic volumes can be evaluated thoroughly, and multiple seismic datasets (e.g. various azimuthal volumes) can be evaluated consistently for multi-scenario analysis to reduce subsurface risk and inform better decisions at all phases of the E&P Asset lifecycle.

Introduction

Interpretation of faults on seismic has long been regarded as a critical aspect of accurately defining reservoirs due to importance of faults in a variety of areas, such as their role in top-seal integrity, field compartmentalization, their potential to act as conduits for fluid flow, and the potential well drilling risks they present.

Traditional seismic fault interpretation workflow is well established. It is built on many years of methodology refinements and technological advances that have enabled more efficient and accurate interpretation of faults on seismic data. Despite technology advances however, traditional workflows for picking faults on seismic remain daunting user-intensive tasks. The current 'standard' approach of conducting line-by-line, fault-by-fault interpretation on seismic sections is inherently time consuming, repetitive, and tedious. Opportunities thus exist to exploit emerging digital technologies to augment human efforts in seismic interpretation workflows to reduce time and effort spent on lower level tasks such as repeated mouse button clicks to "draw" faults on seismic sections and then map them to test consistency, as is done in current "standard" workflows.

In recent years, application of Artificial Intelligence (AI) in combination with automation, high performance computing and cloud has been at the fore of emerging technologies shaping our world (Maniar et al., 2018). Machine Learning (ML) as a branch of AI is particularly well suited for human guided fault interpretation on seismic data because it allows optimal fusing of interpreters' judgement in training machines to identify and automatically interpret faults very quickly, more precisely, and potentially more accurately.

Intelligent Integrated Subsurface Modeling

The project upon which this paper is based was conducted as one of six projects contributing to the Intelligent Integrated Subsurface Modeling (IISM) program at ADNOC in partnership with Schlumberger (Mawlad et al., 2019). The IISM system joint research and innovation program was commissioned to deliver orders of magnitude faster performance and ultimately facilitate optimized reservoir management.

The IISM-Seismic project specifically sought to maximize the value that can be extracted from seismic data through use of AI to significantly reduce time spent on manual and repetitive task of fault interpretation (Mawlad et al., 2019). Input data for the IISM-Seismic project was an Onshore Abu Dhabi seismic dataset provided by ADNOC. The data featured large-scale, near vertical faults with subtle throws. As discussed in ensuing sections of this paper, envisaged benefits were realized from the subject project, as ML technology produced robust fault interpretations from both time and depth domain seismic data.

Workflow & methodology

High level ML architecture overview

The ML-assisted fault interpretation approach outlined in this manuscript is based on an adoption of deep convolutional neural network (CNN) using a tuned 2D U-Net architecture. Whilst U-Net was originally developed for biomedical use, this project took advantage from its advanced image recognition performance to derive an ML-assisted fault interpretation solution to identify and extract faults from seismic data.

The deep CNN was trained to identify faults on seismic by inputting 'training labels' (or sparse user-generated fault interpretation objects); primary outputs from the full workflow applied in the project are fault prediction cubes and extracted fault objects that can be used in standard 3D reservoir models or in regular seismic interpretation workflows. Figure 1 depicts high-level overview of the fault prediction workflow used, including a feedback loop to enable re-training of the deep CNN if quality control (QC) reveals opportunities to enhance prediction results.

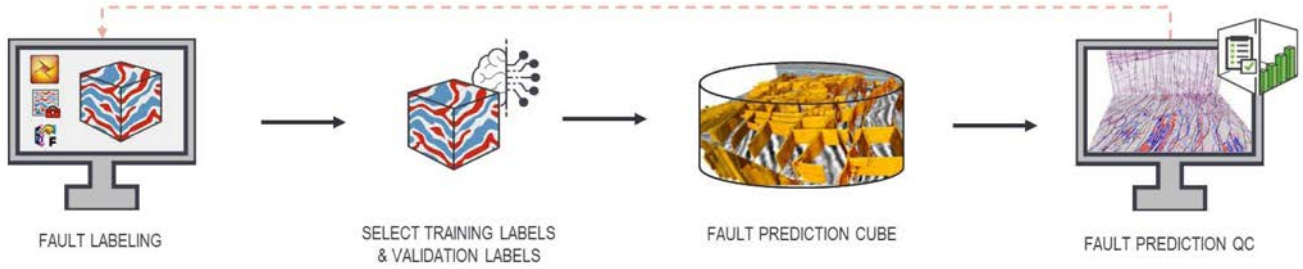

Figure 1—High-level overview of user-trained fault prediction approach

At a more granular level, the applied solution can be said to generate 3 fault prediction outputs per ML run; these are in the inline direction, crossline direction, and a final merged result. The final merge itself is the product of two steps: first, the inline and crossline predictions are merged using a mathematical max function to generate the initial merged cube; next, initial merged cube is used as input for another deep CNN ML run which is based on a pre-trained model that processes the initial merged cube in the Z direction to enhance continuity and consistency of fault prediction results.

For the purpose of this manuscript, the U-Net deep CNN architecture applied in the project will be referred to as the 'ML-System'.

Label generation and fault prediction

Apart from the seismic amplitude volume, the other primary input for the ML-System are user-supplied fault labels. **Fault Labels** are user-supplied sparsely interpreted unassigned fault sticks that are fed into the ML-System to label what faults look like on the given seismic volume according to the interpreter's perspective. The ML-System then trains itself on the input fault labels, and subsequently makes fault predictions for every pixel across the entire seismic volume.

Fault Training Labels were created on six lines in this study, representing <1% of the available seismic crosslines. Selection of 'training lines' (seismic sections on which training labels were picked) was governed by data quality, SNR, frequency content, and structural variability seen on the seismic volume. In order to sufficiently train the ML-System, the variability of structure in the volume must be captured by the training labels to adequately tag structures of interest (i.e. the more variable the structure, the more training labels are required).

'**Validation Lines**' are additional label lines which are used to assess performance of the ML-System's prediction during QC step. Since validation labels are in the same format as training labels, they can be used as additional data in re-training the ML-System to further improve prediction performance.

Figure 2 illustrates the use of training and validation lines in generating initial fault predictions and in re-training the ML-System for improved fault predictions.

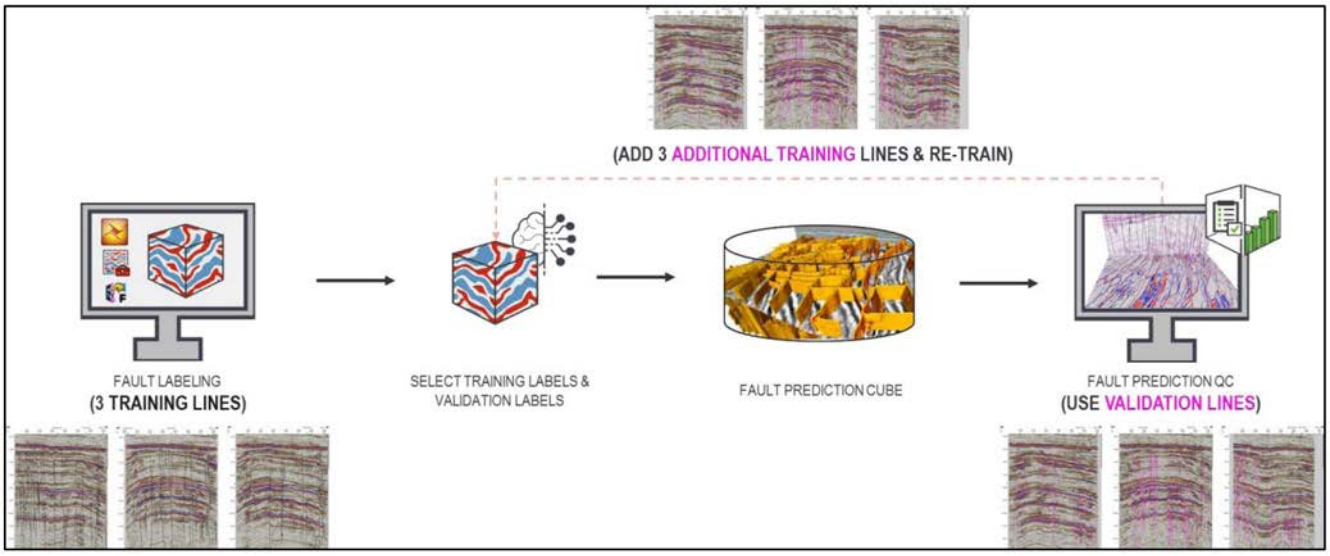

Figure 2—Approach in utilizing training and validation data

The fault labels were used in combination with seismic amplitude volume by the ML-System to create a fault **prediction cube** which can be visualized in 3D and inspected on seismic planes. This fault prediction cube contains fault prediction attribute that is constrained by a confidence threshold (e.g. the yellow planes in figure 3 represent areas where the ML-System was *at least* 50% confident that a fault was present).

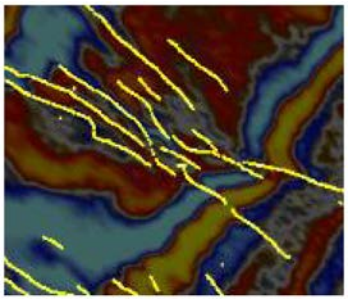

Figure 3—Zoomed example of fault prediction time slice, comprised of original raw seismic data, and fault prediction locations generated by the ML-System highlighted by the yellow

Fault extraction

The prediction cube was used as input for the next phase of the process: fault extraction. In this process, segmented faults were derived from the prediction volume in two primary stages: 'Evaluate' and 'Extract'.

During the **'Evaluate' stage**, geometric analysis is conducted using Principal Component Analysis (PCA), in which Eigen values and vectors are analyzed for the generation of azimuth and planarity cubes. These attributes are created for the purposes of understanding the geometric coherency of faults and to facilitate the segmenting/splitting of faults in low planarity areas, respectively. The planarity cube ranges in value from 0 to 1, where 0 represents highly non-planar, whilst 1 represents a completely flat plane. Figure 4 shows examples of each of these attribute cubes.

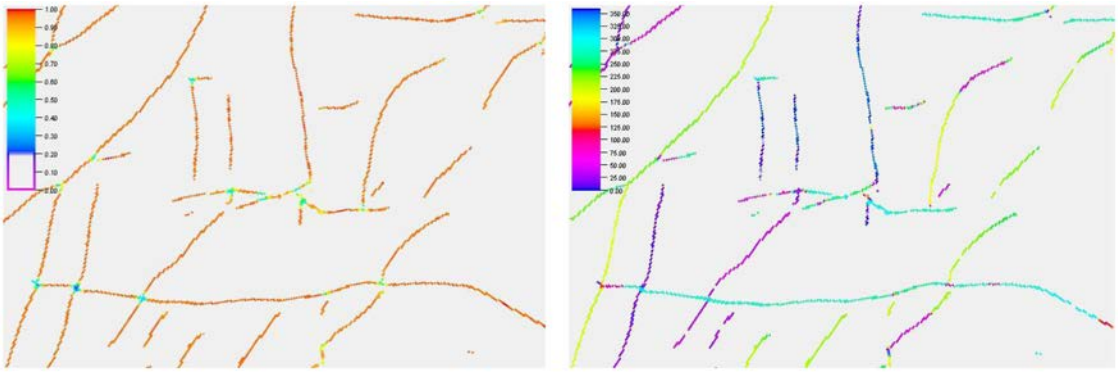

Figure 4—Time slice example of planarity (left) and azimuth (right) cubes created from a fault prediction cube

The **'Extract' stage** involves fault patch extraction based on the azimuth sector and overlap, followed by optimization and merging of extracted fault patches. This was conducted using sectored merging, 3D Gap Check, and graph-based optimization. Additional parameterization enabled fault planarity thresholds to be considered when splitting faults at intersections or when merging faults. Extraction stage result was a set of discretized fault pointsets, which were validated, quality checked, and used for model building.

Model building

Extracted pointsets were utilized as inputs for generating fault framework objects that were incorporated in a Structural Framework model using tools currently available within the Petrel E&P platform. The process of model generation was somewhat simplified due to fault segmentation that already occurred as part of the fault extraction process. Horizons used in the structural framework model were created using standard interpretation workflows.

Automated fault stick generation

To provide closed-loop control over fault object outputs derived from the ML-System, a Python plug-in was developed to extract fault-sticks directly from fault framework models. Output fault-sticks were generated at consistent intervals and can be edited with existing traditional seismic interpretation tools, thereby meeting objective of closed-loop integration with traditional workflows.

The overall high-level cycle from ML-assisted fault interpretation through to closed-loop integration with traditional workflows is depicted in figure 5.

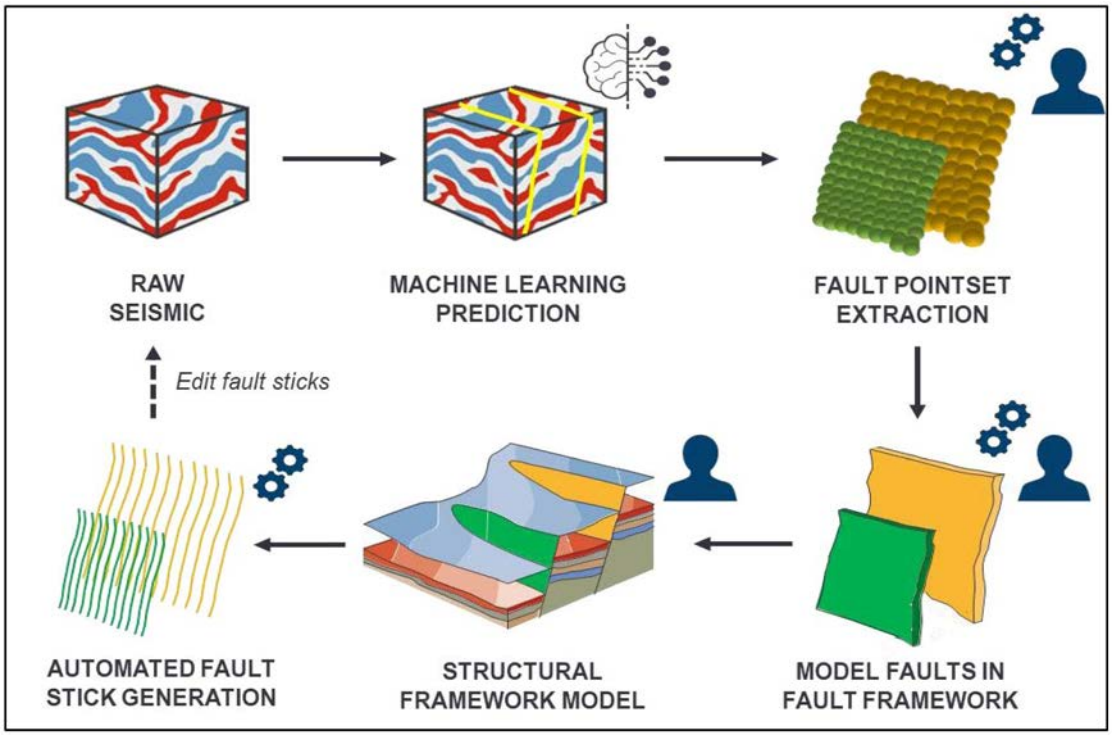

Figure 5—High-level overview of ML-assisted fault interpretation workflow
Key: Brain = machine learning, cogs = automation, human = end-user input

Presentation of Data

Seismic Data (ML-System input). Primary input for the project was onshore seismic data approximately 36x19km in size. Both PSTM time and PSDM depth cubes were used in order to assess ML-System applicability to both domains. Figure 6 shows z-slices from the depth and time volumes; observe that PSDM volume is smoother than the PSTM volume; it will be shown later how this affected ML-System prediction results. Figure 7 shows vertical section through the seismic volume; indications of near vertical NW-SE trending faults are seen on section.

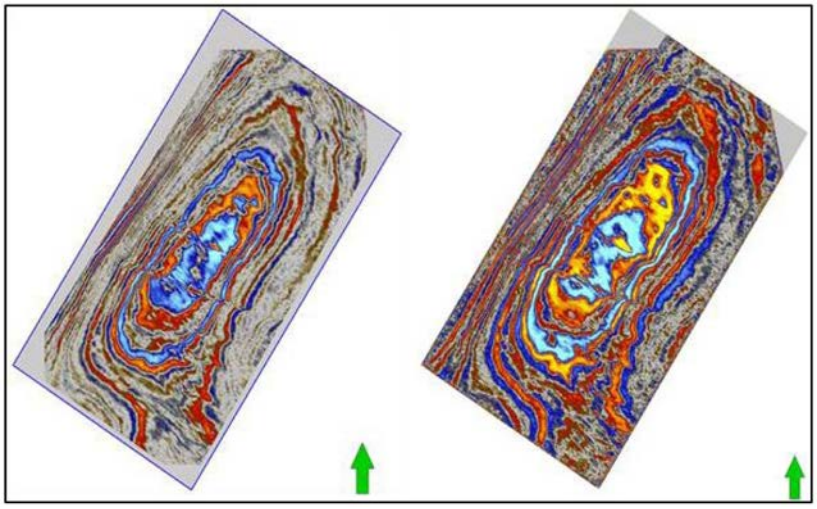

Figure 6—Time slices from input seismic volumes. (Left: PSDM Depth cube. Right: PSTM time cube)

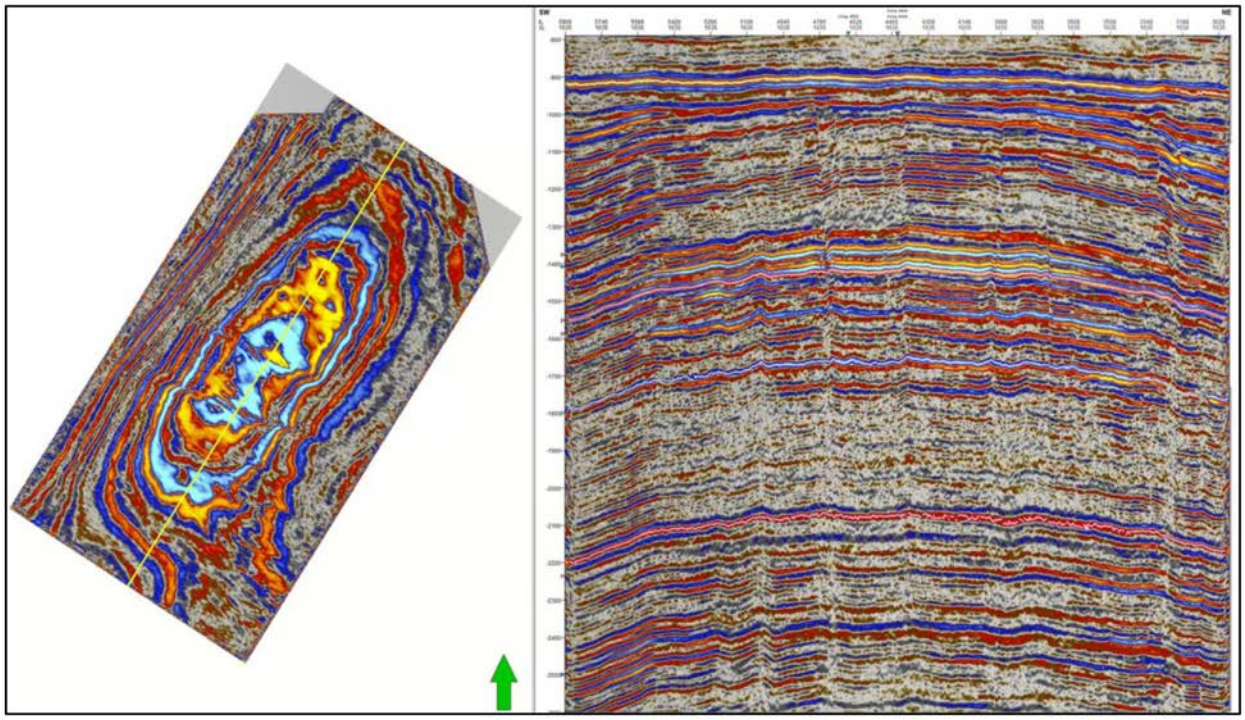

Figure 7—Time slice (left) and seismic crossline (right) showing predominant NW-SE trending faults

Training Data (ML-System input). ML fault prediction training data was by way of user-interpreted unassigned fault sticks that were picked on few seismic sections ("label lines"). To assess impact of quantity of training data on prediction outcome, ML-System was run in two scenarios: one scenario using only 3 training lines, and another using 6 training lines. All training labels were picked on crossline, as the crossline direction is perpendicular to predominant NW-SE trend of faults in the field. To sufficiently train the ML-System, label lines were selected to adequately capture variability of fault data across the seismic volume.

Validation Data (ML-System input). Validation of ML-System training convergence was carried out in two scenarios. In the first scenario, user interpreted faults on three training label lines were inputted into the ML-System for prediction training, whilst a different set of three label lines were used for validation. In this case, the user interpreted faults on the validation lines were used by the ML training algorithm for self-calibration to reach optimum training. Optimum training was deemed to be reached at the point where fault prediction results at validation label lines best matched user interpreted faults on the lines. In the second validation scenario, user interpreted faults on six training label lines were inputted into the ML-System for prediction training, and validation was done by comparing prediction outputs with user interpreted faults on a subset of three lines from the input label lines. Figure 8 illustrates the positions of label line used for training and validation, while Figures 9 shows example fault labels picked on one of the label lines using depth and time domain seismic. Notice that the PSDM depth volume is seen to be smoother and, as such, showed less indication of fault structures.

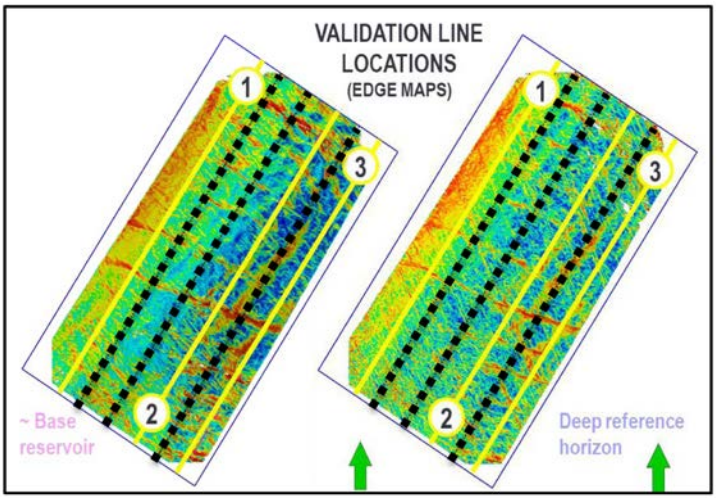

Figure 8—Location of Label Lines on which Fault Labels were picked for input into ML-System input. (*Dashed black lines mark position of seismic crossline sections that were exclusively used for training, and not used for validation. Solid yellow lines mark positions of seismic crosslines that were used for validation in first ML-System scenario, and then as additional training lines in second ML-System scenario*)

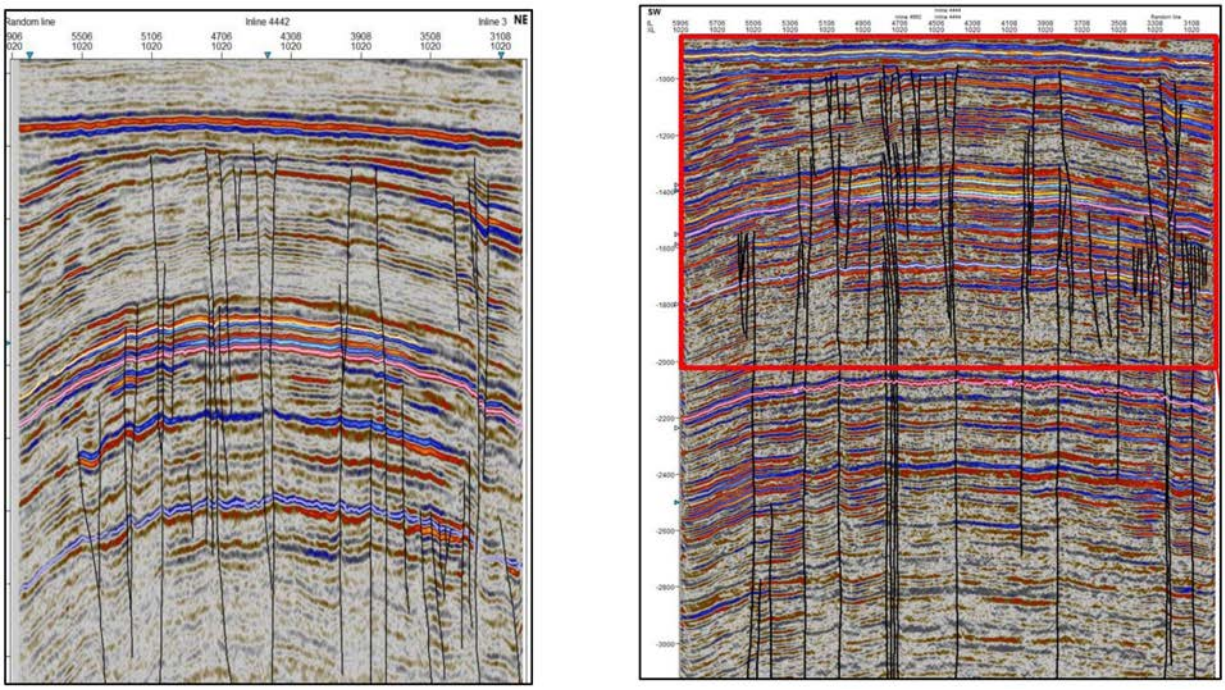

Figure 9—Fault labels picked on PSDM depth seismic data (left) and PSTM time domain data (right). Both lines were picked at location of label line 2 in figure 8.

'Ground-Truth' Interpretation (ML-System output validation). To facilitate 'ground-truth' validation, faults derived from ML-System were compared with a set of 91 manually picked faults from the same seismic data volumes. The manual faults were independently picked prior to commencement of the IISM-Seismic project. Comparison of the manual faults with faults derived from the ML-System is discussed under Results section of this paper. Total number of available manually picked faults was 131; of these, the 91 faults selected for "ground-truth" validation are those that correspond to discernable fault features on available seismic volumes. The remaining 22 manually picked faults were deemed to be below seismic resolution and were mainly picked based on non-seismic data (e.g. well data).

Presentation of Results

As earlier described, the ML-System was run in two scenarios: one scenario using training fault labels from three lines, paired with validation fault labels from a separate 3 lines; the other scenario used fault labels from six training lines, while validation was done on a subset comprised of three lines. Thus, two sets of predicted faults were generated from the different ML-System runs. The output ML-System derived faults were analyzed against the following criteria:

- *ML Prediction Performance* – How performant was the ML prediction in the 3 training-line scenario versus the 6 training-line scenario?
- *Domain Dependence* – Was there noticeable difference in the nature of the ML prediction between time and depth volumes?
- *Prediction Convergence at Training Lines* – How well did the ML prediction correlate with labels on training lines?
- *Prediction far from Training Lines* – How well did the ML prediction identify structures furthest away from label lines with training data?

ML Prediction Performance:. ML prediction results were compared for runs using 3 lines (0.38% of the available crosslines) vs 6 lines (0.76% of the available crosslines). Figure 10 highlights the differences in prediction results when three training label lines were used versus when six training label lines were used. Lateral and vertical continuity of predicted faults was clearly enhanced when more training label lines were introduced.

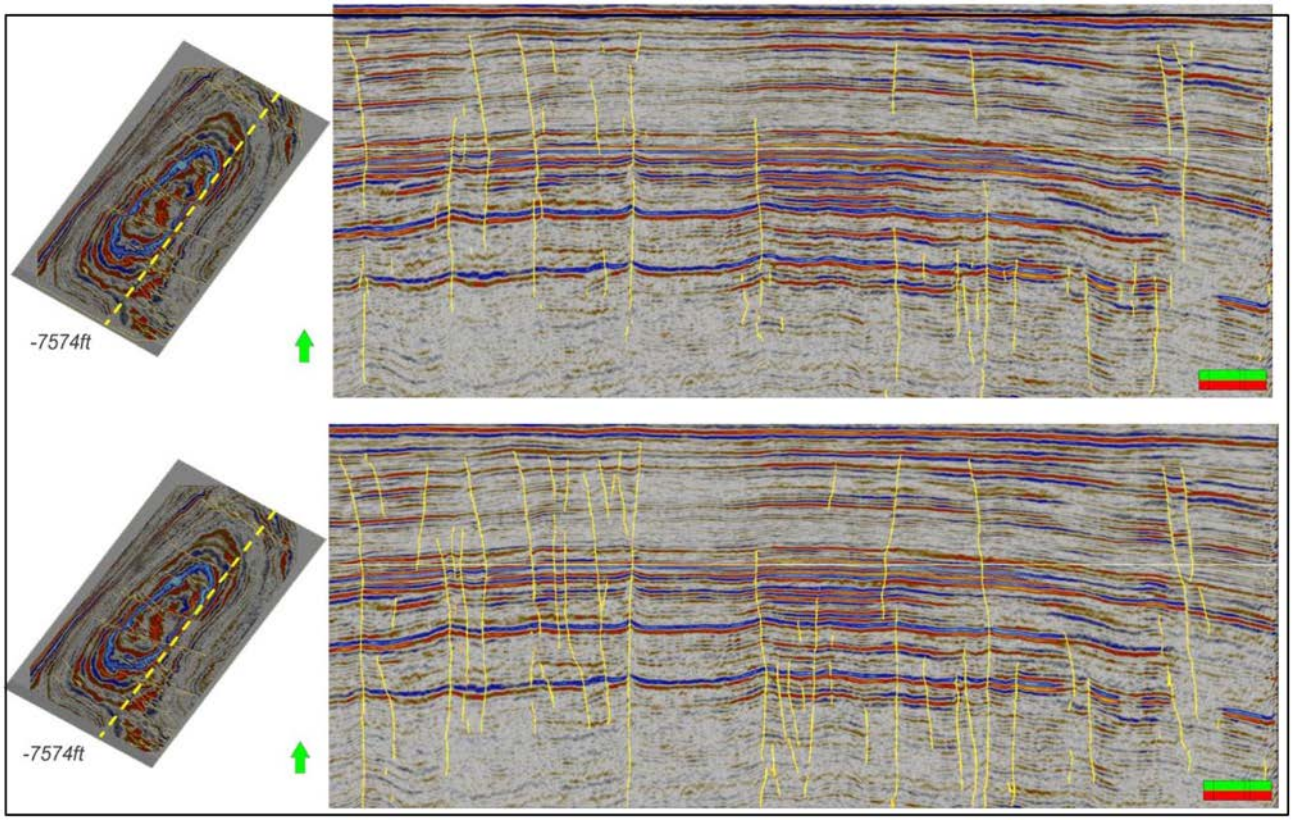

Figure 10—Comparing fault prediction results from ML-System run on PSDM depth volume using 3 label line inputs (top) versus 6 label line inputs (bottom). Images show fault prediction cube rendered with the seismic amplitude data. Predicted faults are shown as yellow lineaments. Advantage of additional training data in the 6 training lines scenario is evident, as fault predictions are more continuous and more conformant with the seismic image.

Domain Dependence:. Results from using time and depth seismic data were compared to assess if domain dependent difference exists in ML-System performance. For this assessment, equivalent training labels were generated for the time and depth volumes used in the project, and then ML-System outputs from both scenarios were compared. It was observed that the ML-System consistently identified equivalent structures in both runs, as such it was concluded that data domain has no impact on system performance. Figure 11 depicts a side by side comparison of prediction results at corresponding Z-slices. The time cube arguably contains subtleties that were smoothed out during processing of the depth cube; thence, ML-System results from the time seismic cube input were slightly more granular and prediction resolution slightly higher. Overall, however, ML predictions appeared consistent between the two volumes, therefore supporting conclusion that this approach is domain-agnostic and can be successfully applied to datasets containing seismically resolvable faults regardless of domain.

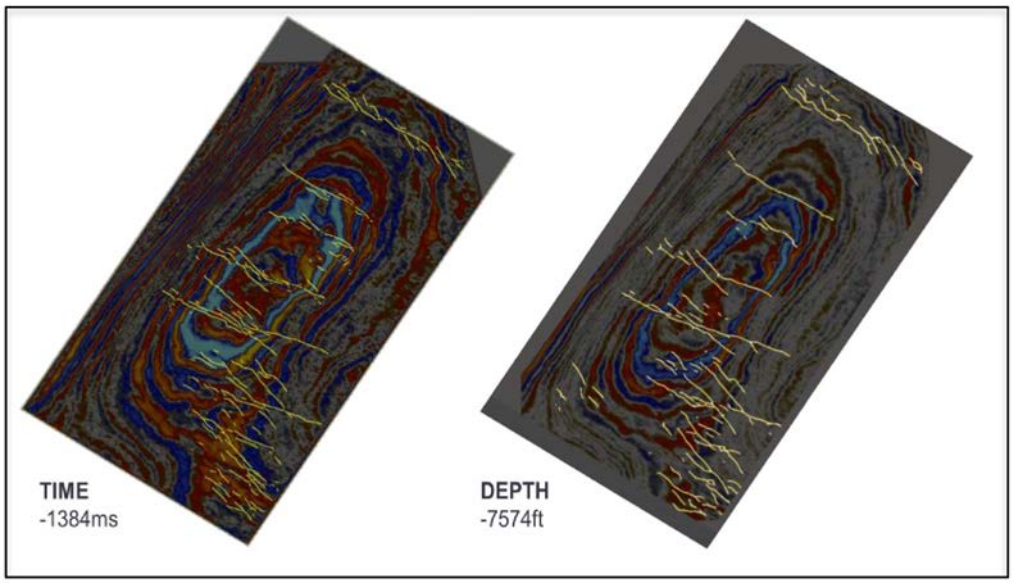

Figure 11—Comparison between time (left) and depth (right) results shown in the time slice orientation at equivalent Z locations, based on training from *6 lines*

Prediction Convergence with Training Lines:. To evaluate a prediction, it is first useful to assess its performance in areas guided by training data. This was done by observing how the ML-System predicted faults on the same lines that contained training labels. It was found that the ML-System was 'well converged' with the training data, meaning it was sufficiently trained to generate robust predictions in areas with training data. Generally, predictions on sections that contain training labels were seen to exhibit excellent continuity and mapped well to input fault labels. This gives initial confidence in the results, because if prediction were not properly converged on label lines, there would be little expectation for results to improve in areas farther away from training data.

Figures 12 provides examples of the ML-System fault prediction compared with the fault labels used to train the ML-System. The figure exhibits generally excellent continuity of predictions and shows that predictions map extremely well to input fault labels. This is particularly evident towards the center of the section, where the narrow and highly verticalized fault corridor is accurately captured by the prediction.

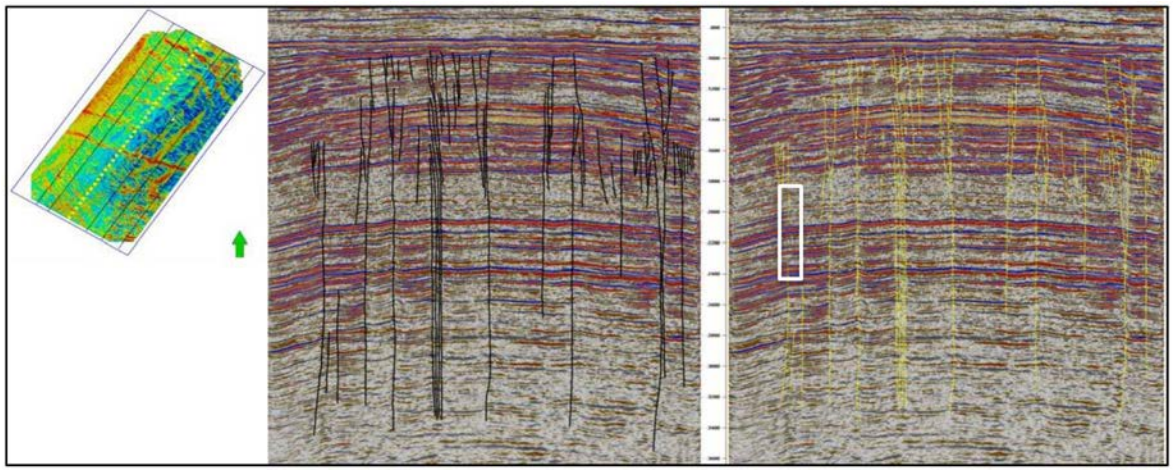

Figure 12—Example crossline illustrating convergence of the ML prediction (shown in yellow, right) with training labels (shown in black, center). Line location is shown by the dashed yellow line on the map. White oblong highlights the only area where the prediction is not fully continuous.

Slight variations were observed in few places between ML-System fault predictions and input fault labels. An example of this is highlighted in figure 12, where prediction for one fault was not fully continuous within an interval over which the training label was continuous. This is thought to have occurred because seismic events in the affected interval are strong and continuous, thereby reducing the ML-System's confidence for existence of faults therein. This is further indication that the ML-System is sufficiently data-driven and discriminatory to limit likelihood of false positives (i.e. the incorrect prediction of a fault where no fault is visible). This was deemed reassuring. The said small gaps in predicted faults were of little concern to final outcome, as such gaps did not prevent fault patches from being connected as single throughgoing faults during subsequent fault extraction and modelling phases.

Prediction far from Training Lines:. Arguably, the most important aspect of the ML prediction is in sections farthest from any training data. If a prediction is deemed to be of 'good quality' and appears consistent with interpretation expectations in areas that are not directly guided by training labels, this evidences that the ML-System has been sufficiently trained to make predictions with high level of confidence. Figure 13 includes two examples of sections halfway between the training lines, and therefore farthest from guidance of user-generated training labels. The sections show that the larger-scale faults were clearly predicted. Areas that show small 'gaps' in the prediction planes are highlighted. The gaps are attributed to continuous high amplitude reflectors that do not exhibit significant displacement features. As stated earlier, such small gaps in prediction attribute did not create issues in later stages of the workflow, as continuous fault objects were modelled and extracted. Figure 14 shows raw prediction cube in 3D view, highlighting prediction density and continuity. It also highlights some minor prediction noise, which can be eliminated using stricter confidence thresholds.

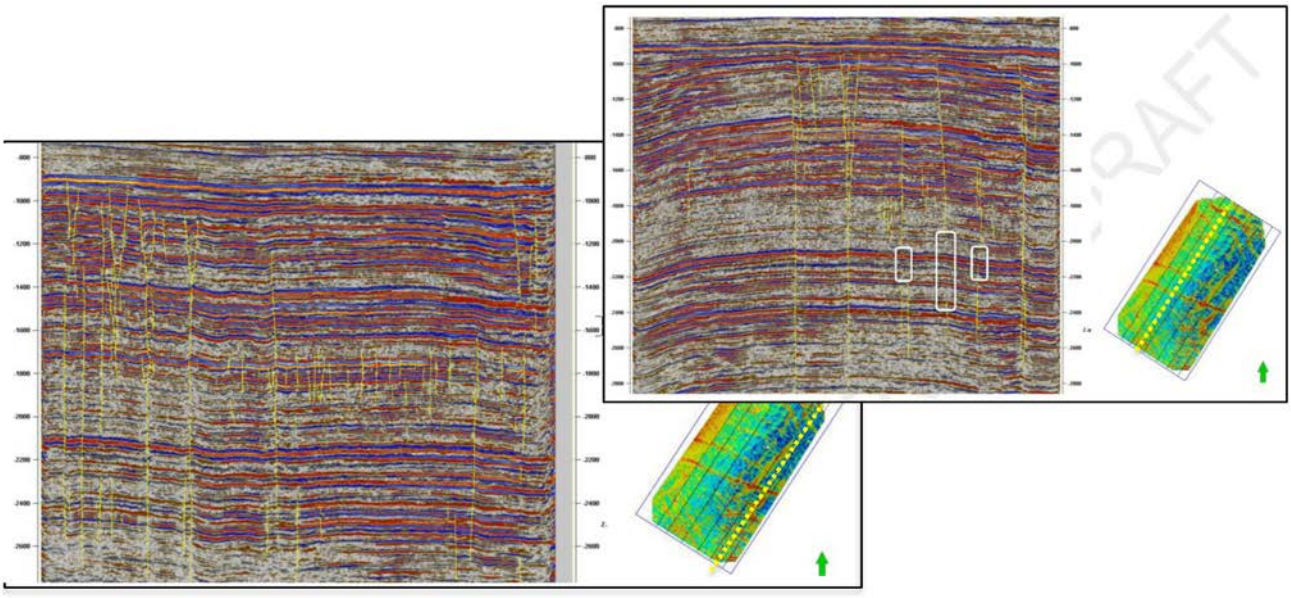

Figure 13—Fault predictions shown on sections farthest from training labels. Prediction 'gaps' highlighted with white boxes.

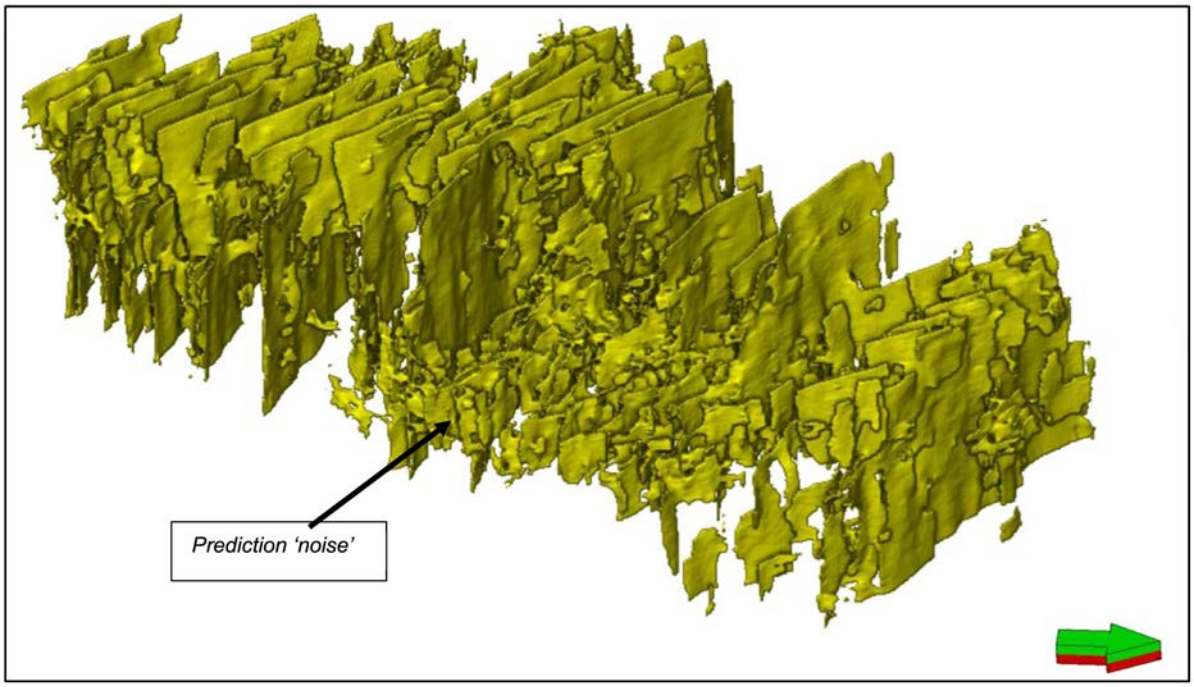

Figure 14—Fault predictions cube with filtered null prediction values, illustrating continuous nature of fault prediction attribute. Minor prediction 'noise' is also annotated; these can be eliminated using stricter confidence thresholds to discriminate null prediction values.

Fault Pointset Extraction. Following fault prediction, the next key phase of the ML-assisted fault interpretation workflow is to extract fault pointsets using the methodology earlier explained under workflow and methodology section of this manuscript. Figure 15 shows the planarity and azimuth cubes generated for purpose of fault pointset extraction. Planirity and azimuth cubes were used to parameterize fault pointset extraction from the prediction cube; essentially, extracted fault pointsets were segmented based on azimuth and planarity discriminants. For this project, faults pointsets were extracted in particular orientations and were constrained by minimum size limit. Parameters can however be set to allow all faults to be extracted, regardless of said constraints.

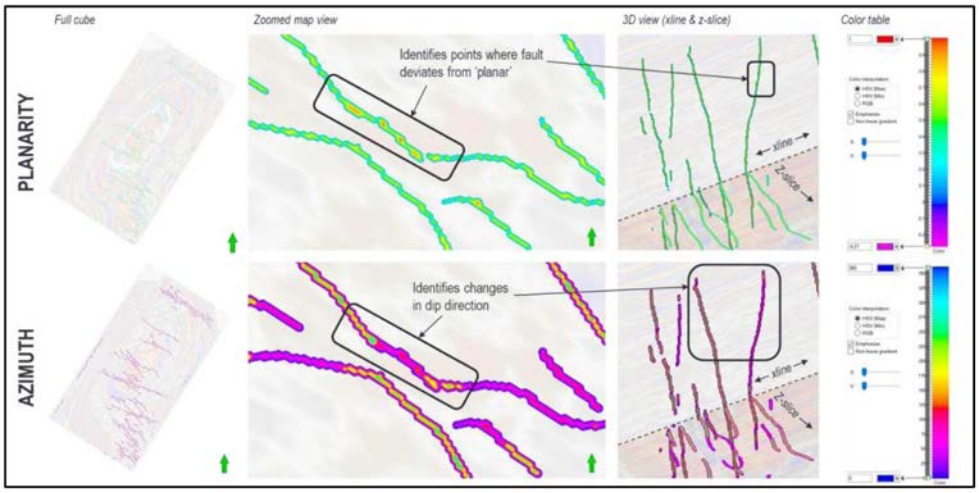

Figure 15—Planarity and azimuth cubes generated for the fault extraction process.

Two approaches were used for parameterizing fault pointset extraction in the project; both were found to be equally effective. In the first approach, the extraction process was parameterized using four individual 'narrow' fault orientations, while the second approach used a single broader discriminant to include all fault orientations of interest. As said already, results were similar in both cases. Figure 16 highlights the various groups used in the first parametrization scenario and the cumulative range used for the second scenario.

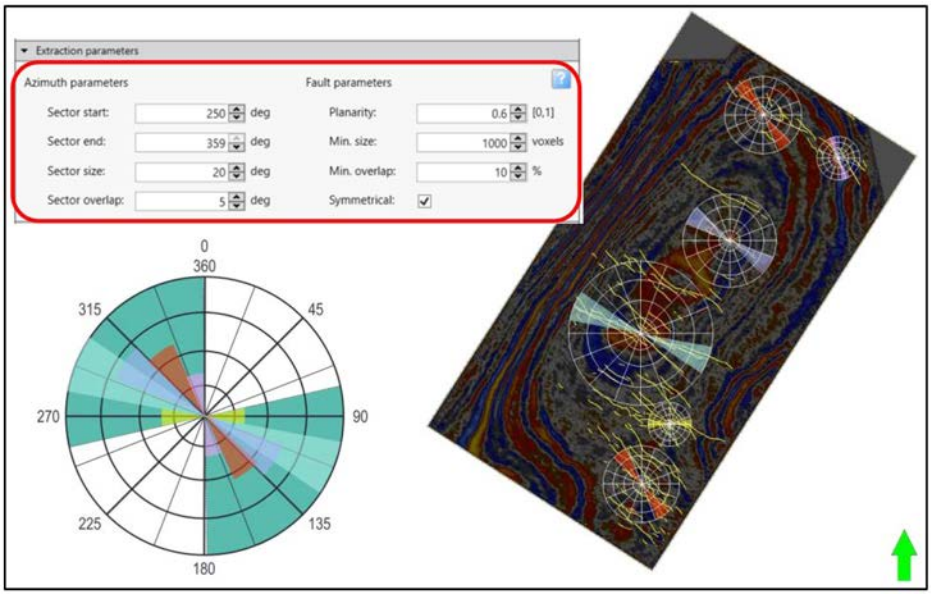

Figure 16—Parameterization of fault pointsets extraction process using azimuth and planarity discriminants.Rose diagram shows different orientations used for extraction discrimination

Figure 17 shows result of the fault pointset extraction process in this project. A total of 262 faults were extracted, comprised of 59 'major' faults (as defined by a minimum fault surface area threshold and >1km in length) and 203 smaller-scale faults. As expected, extracted fault pointsets proved to strongly correlate with the prediction results and pointsets mapped exactly to fault prediction planes.

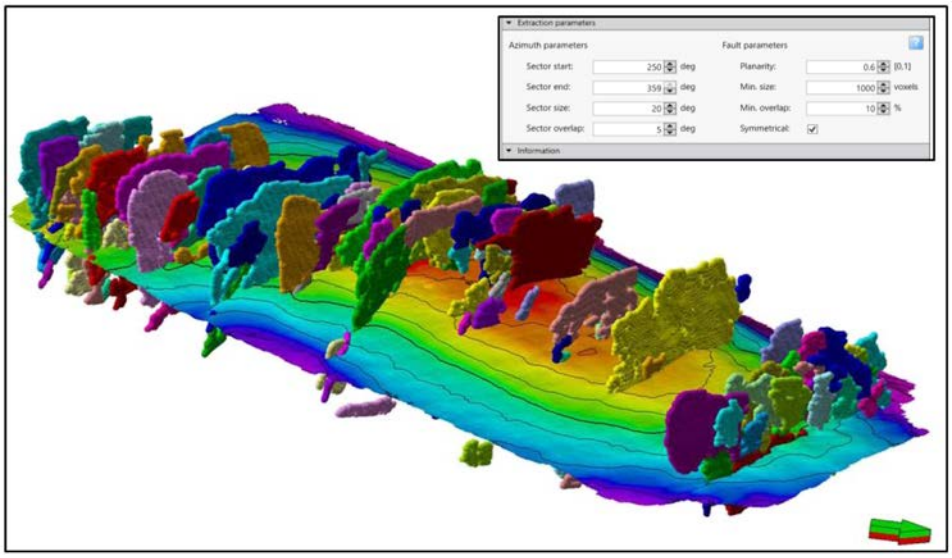

Figure 17—Discretized fault pointsets extracted from fault prediction cube; segmentation was parameterized by azimuthal and planarity values

Fault Framework Modeling and Automated Fault Stick Generation:. Extracted fault pointsets were used as inputs for fault framework modelling. The model was cropped to focus on the region surrounding the primary reservoir interval. Faults required little manipulation and were typically modeled at a resolution of 200m, with extrapolation varying between 50m and 250m (with only 12 faults requiring adjustment from a default value of 50m). This ensured robust truncations between faults, and fault relationships (major/minor) were accurately calculated automatically in all but two cases.

To ensure closed-loop integration of faults produced by ML-System with traditional workflows, a tool was developed to automatically extract fault sticks from the fault framework model. The extracted fault sticks are regular interpretation objects, which can be directly incorporated into traditional seismic interpretation and modelling workflows. Figure 18 illustrates the final fault framework model derived from the fault pointsets that were extracted from ML-System predictions; the model included all 59 major faults from ML-System predictions. Figure 18 also shows fault sticks extracted from the final fault framework model, thus demonstrating that objective of closed-loop integration of ML-System with traditional workflows was attained.

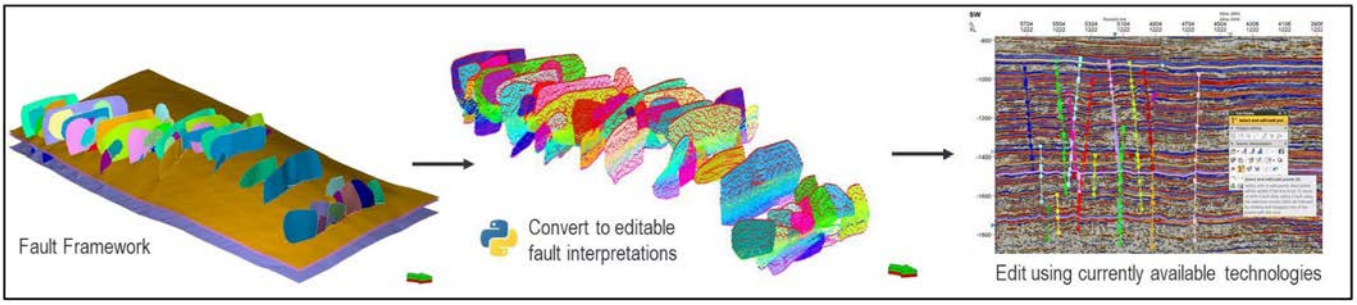

Figure 18—Extraction of fault sticks from fault framework model, allowing for closed-loop integration ML-System outputs with regular interpretation and modelling technologies

Validation and Comparison with Traditional Approach:. Output of ML-System was compared with manually interpreted faults to assess value contribution from deploying the ML-assisted fault interpretation. Assessment was done in terms prediction match with manual fault picks and workflow efficiency gains.

Reference fault dataset from manual interpretation included only the 91 faults earlier described as corresponding to discernable fault features on seismic amplitude volumes used for the project.

Prediction Match:. To compare how much ML-derived faults match manually interpreted faults, it was decided that a ML-derived fault and a manual fault would be deemed as matching where there is >50% overlap between the two. There were 91manually picked faults associated with discernable fault features on seismic; of these, there were 59 that matched ML-derive faults (i.e. on this project, the ML-System predicted 65% of faults that were manually picked prior to commencement of ML project). It should be noted however that the ML-System also yielded valid faults that were not previously picked manually; thus, further lending credence to effectiveness of the applied ML-assisted workflow for fault interpretation. Figure 19 shows example of matching ML-System derived fault and manually interpreted fault. The match between the two faults is particularly good.

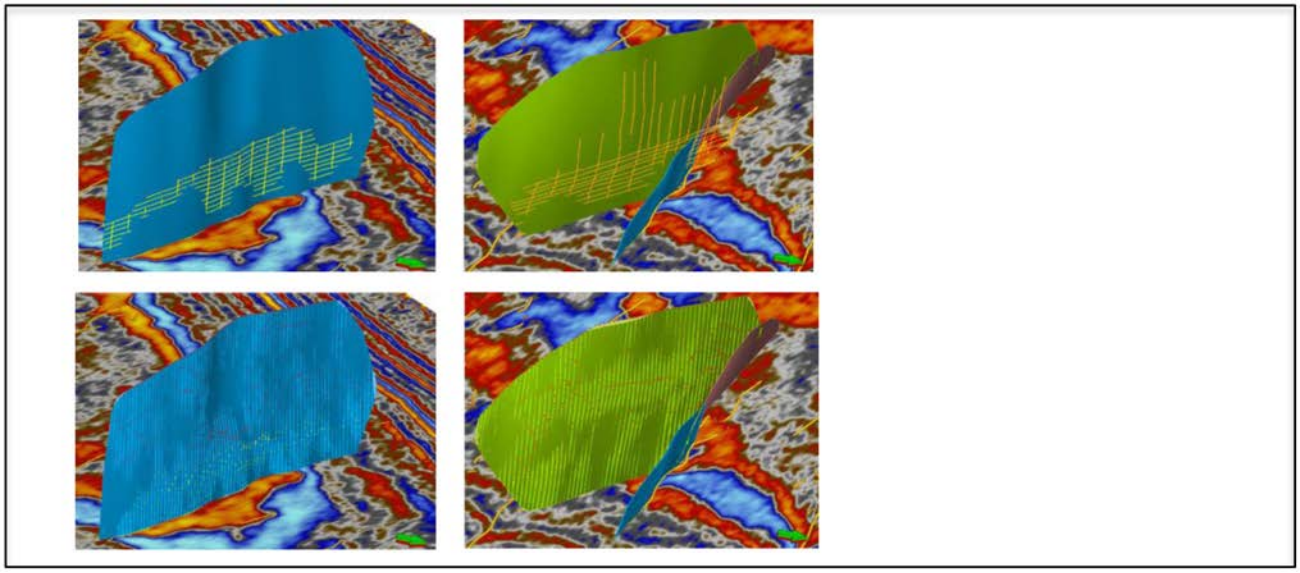

Figure 19—Comparison between fault sticks generated via traditional means and those generated by ML-assisted fault interpretation.

- **Top left: Geophysicist's fault sticks (yellow) shown with modelled fault from ML prediction (blue)**
- **Top right: Geophysicist's fault sticks (orange) shown with modelled fault from ML prediction (green)**
- **Bottom left: Fault sticks automatically extracted from modelled fault (blue) shown with modelled fault from ML prediction (blue)**
- **Bottom right: Fault sticks automatically extracted from modelled fault (green) shown with modelled fault from ML prediction (green)**

Comparison between fault sticks generated via traditional means and those generated by ML-assisted fault interpretation.

On the flip side, Figure 20 shows example of an area where manual faults were picked but ML-System did not detect any fault. The figure shows that there was indeed no apparent fault indication on seismic in that interval. The manually interpreted faults in that location relied more on finer interpreter judgement than data signature (for example, user may pick the manual fault based on attitude of adjacent sections). This underscores the undeniable value of interpreter judgement for robust results.

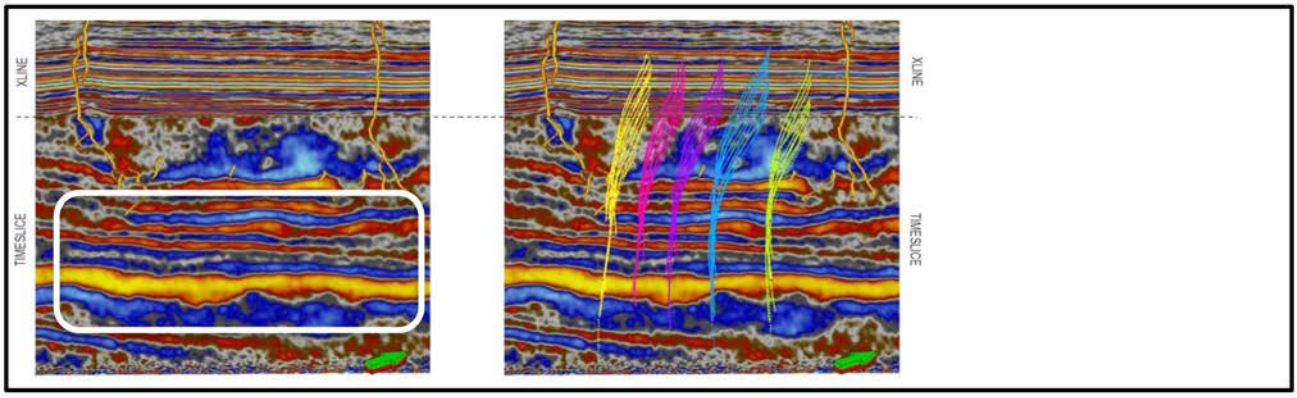

Figure 20—Highlighting an area that predictions were not made by the ML-System because there was no fault indication evident on seismic (boxed area on left-hand-side image). Faults were picked across the same area by the interpreter (right-hand-side image), possibly relying judgement from adjacent intervals or other data sources.

Workflow Efficiency:. Time spent on fault interpretation using ML-assisted workflow was compared with time spent on manual fault interpretation. It was calculated that use of ML-assisted interpretation workflow can save upwards of 80% of evaluation time required to interpret the major fault features. With ML-assistance making interpretation of most major faults quicker and more precise, the interpreter can spend freed time on more challenging areas of the data where the ML-assisted method may struggle to give reliable results because of data quality and/or other factors. Within the scope of faults that can be derived from ML-assisted means, it was observed that results were not just quicker, but were better quality overall (figure 21).

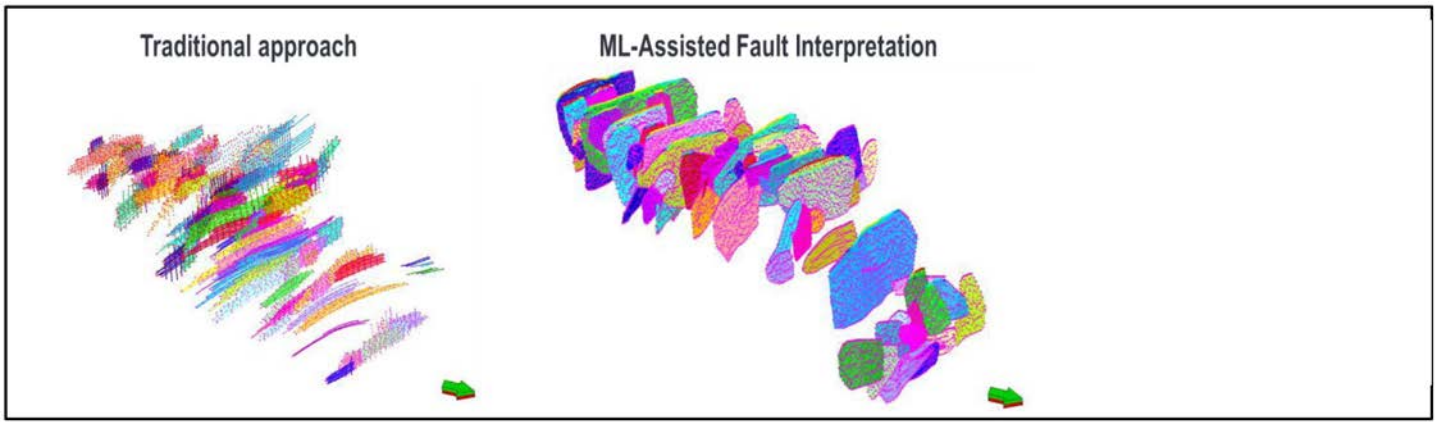

Figure 21—Faults from ML-assisted method (right) took less than 20% of evaluation time to complete versus fault interpretation using manual methods (left). ML-assisted fault interpretation outputs also showed overall better quality than manual faults. However, ML-assisted method did not capture as many faults as manual method, particularly in areas where seismic expression of faulting was moot.

Discussion and conclusions

The approach employed for this project saw the successful application of ML-assisted fault interpretation to a Middle Eastern carbonate field dataset that was characterized by extensive near vertical faults with subtle throws. Overall project objectives of increasing efficiency and precision were realized as evaluation time to complete 65% of major fault interpretation was reached in less than 20% of evaluation time for traditional methods. ML-assisted method was implemented with closed-loop integration with traditional methods, thereby enhancing benefits overall.

Results achieved in this project demonstrate benefit of hybridized workflow where the geoscientist is empowered by ML to accelerate repetitive lower level tasks to free time for deeper multi-scenario analysis.

ML enabled reduction in evaluation time and effort, and increased evaluation accuracy presents benefit that single seismic volumes can be evaluated thoroughly, and multiple seismic datasets (e.g. multi-azimuthal volumes) can be evaluated consistently for multi-scenario analysis to reduce subsurface risk and inform better decisions at every phase of the E&P Asset lifecycle.

Acknowledgement

The authors acknowledge the following people for support and enablement in the course of the IISM-Seismic project and writing this paper: Salem El-Abd Salem, Ali Rawahi, Mohammed Ahmed Al Nuaimi (ADNOC); Stewart Smith, Steve Freeman, Ali Razouki (Schlumberger).

Nomenclature

AI – Artificial Intelligence
CNN – Convolutional Neural Network
ML – Machine Learning
SNR – Signal to Noise Ratio

References

Maniar, H., Ryali, S., Kulkarni, M. S., & Abubakar, A. 2018, November 30. Machine-learning methods in geoscience. Society of Exploration Geophysicists.

Mawlad A., Mohand, R., Agnihotri, P., Pamungkas, S., Omobude, O., Mustapha, H., Freeman, S., Ghorayeb, K. and Razouki, A. 2019. Embracing the Digital and Artificial Intelligence Revolution for Reservoir Management - Intelligent Integrated Subsurface Modelling IISM. Society of Petroleum Engineers.