XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGENEERING

Development of a graphical user interface for predicting damage zones of geological faults

Cláudio H. B. de Resende[1], Roberto Quevedo[1], Deane Roehl[2], Bruno Carvalho[3]

[1]*Tecgraf Institute, PUC-Rio*
R. Marquês de São Vicente, 225, 22793-260, Gávea, Rio de Janeiro - RJ, Brazil
claudiohorta@tecgraf.puc-rio.br, rquevedo@tecgraf.puc-rio.br
[2]*Department of Civil and Environmental Engineering and Tecgraf Institute, PUC-Rio*
Rua Marquês de São Vicente, 225, 22451-900, Gávea, Rio de Janeiro - RJ, Brazil
droehl@puc-rio.br
[3]*Petrobras Research Center*
Avenida Horácio Macedo, 950, 21941-915, Rio de Janeiro - RJ, Brazil
brcarvalho@petrobras.com.br

Abstract. In the oil and gas industry, the characterization of fault damage zones is important to develop suitable production strategies. Owing to the high levels of deformation experienced in the past, fault damage zones typically exhibit regions with physical properties that are markedly different from those of the host rocks. Those regions can contain fracture networks triggering preferential flow paths or contain deformation bands creating barriers for fluid flow. For practical applications, the width of fault damage zones has been linked to the fault throw through power law relationships. However, the significant dispersion of outcrop observations has shown that other parameters such as lithology and mechanical properties of the host rock under deformation can be responsible for the scattering. Thus, the assessment of damage zones around geological faults has been performed through numerical analysis incorporating rock plasticity. Nevertheless, such simulations are computationally expensive. To address this limitation, the use of data-driven predictive modeling emerges as a promising alternative. In this context, the present study proposes the development of a user-friendly graphical interface application aimed at predicting damage zones around geological faults. The application utilizes fault geometry and calculated displacements at observation points, derived from a geological model, along with key rock geomechanical parameters such as Young's modulus, Poisson's ratio, friction angle, cohesion, critical pressure, and dilatancy angle. The process is efficiently performed using machine learning models, enabling the user to check the sensitivity of the damage zone width regarding the input parameters in real-time. The proposed tool offers a computationally efficient and interactive approach for the prediction and analysis of fault-associated damage zones, contributing to more informed decision-making in reservoir modeling and management within geologically complex settings.

Keywords: Graphical user interface; Geological faults; Geomechanics; Machine learning;

1 Introduction

Fault zones are complex geological structures that commonly develop in the subsurface as a result of various rock deformation mechanisms. According to Fossen [1] these zones typically consist of a fault core, where deformation is highly localized, and an adjacent damage zone characterized by distributed fracturing and strain. The geological features within both the core and the damage zone can function either as flow barriers or preferential fluid pathways. Consequently, the investigation and characterization of fault zones are of significant relevance to the oil and gas industry. However, the identification of such geological formations is often accompanied by substantial uncertainties, mainly due to challenges such as low seismic resolution.

In light of these limitations, recent studies have employed numerical modeling methodologies to simulate the formation of damage zones and characterize their structural properties [2]. Numerical modeling has become an increasingly essential tool for addressing complex problems in geotechnical engineering, particularly in the context of nonlinear analyses [3, 4]. One of the principal challenges in computational geomechanics is the analysis of geometric nonlinearity, arising from large displacements and strains, as well as material nonlinearity, given the inherent difficulty in predicting the mechanical response under such conditions [5].

A widely adopted approach in practical applications involves the use of an equivalent continuous medium, in which geomaterials are represented as a continuous domain governed by appropriate constitutive models. This framework does not explicitly model imperfections such as discontinuities, shear bands, or karst features. The Finite Element Method (FEM) has been extensively utilized under this paradigm. Conventional FEM analyses, based on classical Lagrangian formulations, typically assume infinitesimal strains and negligible changes to the model's initial geometry. As a result, these analyses rely on simplified strain-displacement relationships. However, such assumptions can lead to inaccuracies when modeling scenarios involving large deformations. To overcome these limitations, alternative formulations that accommodate large-strain behaviors have been proposed and increasingly adopted, as evidenced by Augarde et al. [6].

In recent years, machine learning has gained significant traction in the oil and gas industry, with notable applications in seismic inversion, reservoir characterization, and production optimization [7]. Convolutional Neural Networks (CNNs) have become a powerful tool in seismic reservoir modeling, particularly for identifying fault geometries and estimating the width of damage zones [8, 9]. Several CNN models have proven robust for detecting geomechanical properties [10, 11], characterizing damage zones [12–14], and segmenting geological faults [15–18], among other applications [19–21].

The use of neural networks in seismic data processing dates back to the late 20th century, beginning with seismic deconvolution [22] and early inversion work [23]. More recent efforts, such as those by [8] and [9], used architectures like U-Net and Light-UNet to map damage zones in Norwegian fields. These models employed fault frequency plots to estimate damage zone extents, though results were limited by seismic resolution and data interval spacing [24, 25].

Advancements continue with models like U-Net++, applied by Torabi et al. [7] to predict 3D mechanical properties using labeled seismic and well data. Jiang et al. [26] combined a convolutional autoencoder with Bayesian inversion to estimate reservoir permeability from sparse well data, achieving low error rates and fast convergence. Similarly, a study by Adegbite et al. [27] explored the relationship between porosity, permeability, and pore throat radius using multiple regression analysis, neural networks, and an Adaptive Neuro-Fuzzy Inference System (ANFIS).

This study presents the development of a user-friendly graphical user interface (GUI) designed to evaluate, in real time, the sensitivity of damage zone width to various geomechanical parameters. These parameters include Young's modulus (E), Poisson's ratio (ν), friction angle (ϕ), cohesion (c), critical pressure (c_p), dilatancy angle (ψ), and fault throw (d). The application allows the prediction of damage zone width using five distinct models: three based on power-law regressions, one on linear regression, and one on polynomial regression. To construct these predictive models, a synthetic dataset was generated and numerically evaluated using Abaqus software, which relates the aforementioned parameters to the damage zone width (W). The remainder of this paper is organized as follows: Section 2 presents the methodology; Section 3 provides a detailed description of the graphical user interface; and Section 4 concludes the paper with final remarks and directions for future research.

2 Methodology

This section presents the methodology followed to the development of a graphical user interface designed to predict the width of damage zones in geological faults. The damage zone width is assumed to vary as a function of the fault throw and the geomechanical properties of the surrounding rock. To quantify this width, a finite element model is constructed in Abaqus (Figure 1), where a parabolic distribution of fault throw is applied along the fault plane. The extent of the damage zone is then inferred from the region of plastic deformation within the rock mass.

Despite the accuracy of this numerical approach, the analysis is computationally expensive. Given that one of the main objectives of this study is to investigate the sensitivity of the damage zone width to variations in geomechanical parameters, solving a nonlinear finite element problem for each parameter combination becomes impractical. As a computationally efficient alternative, this work proposes the training and deployment of predictive models capable of approximating the damage zone width with significantly reduced computational cost.

To train a statistical or machine learning model, a synthetic dataset was generated using Latin Hypercube Sampling (LHS), and the corresponding damage zone widths were evaluated for each combination of input parameters. The descriptive statistics of the dataset are presented in Table 1. The minimum and maximum threshold values for each feature were defined based on the typical range observed in carbonate formations from the Brazilian pre-salt reservoirs.

The input variables considered include Young's modulus (E), Poisson's ratio (ν), friction angle (ϕ), cohesion (c), critical pressure (c_p), maximum throw (d_{max}), dilatancy angle (ψ), and damage zone width (W) as the target variable. The direct correlations between the predictor variables and the target are illustrated in Figure 2. This figure highlights a strong and approximately linear correlation, within the defined thresholds, between fault throw and damage zone width. In contrast, the remaining variables exhibit no evident direct relationship with fault throw,

suggesting a less pronounced influence on the target variable.

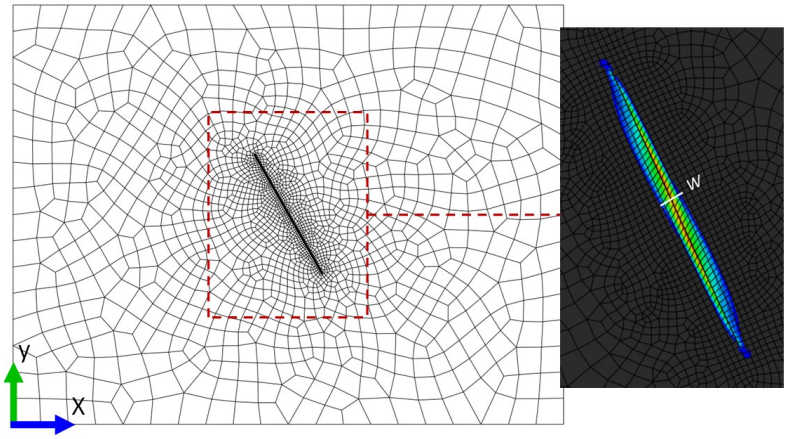


Figure 1. Numerical model.

Table 1. Descriptive statistics.

	E (MPa)	ν (-)	ϕ (deg)	c (MPa)	c_p (MPa)	d_{max} (m)	Ψ (deg)	W (m)
Count	200	200	200	200	200	200	200	200
Mean	26960.49	0.29	28.16	16.13	135.91	160.88	8.16	147.91
Std	13053.60	0.03	4.00	4.43	44.87	79.99	4.00	74.39
Min	5073.23	0.24	21.07	8.13	60.94	23.19	1.07	18.09
25%	15772.99	0.27	24.75	12.33	97.03	91.53	4.75	81.54
50%	26472.75	0.29	28.09	16.15	133.91	157.09	8.09	142.30
75%	38078.31	0.32	31.57	19.97	173.90	230.97	11.57	210.37
Max	50136.78	0.35	35.11	23.55	215.46	299.31	15.11	292.37

The main objective of the application is to delineate a geometric boundary of the damage zone, represented by a set of points, based on the geometric characteristics of the fault, specifically the positions of the points in UTM coordinates and their respective depths, along with the throw values associated with each point. Figure 3 provides an overview of the entire process. Initially, using the throw values (d) and a predictive model (which may or may not rely solely on the throw), the width of the damage zone (W) corresponding to each fault point is estimated. This width is defined as the perpendicular distance from the fault line; thus, it is assumed that half of this distance extends on either side of the fault. As a result, two secondary points are derived for each fault point P_i, located to the left and right, denoted as P_1 and P_2, respectively. Using basic trigonometric relations, it is then possible to compute the coordinates of these secondary points, which collectively define the boundary of the damage zone.

To define W, the application initially relies on five predictive models: three Power Laws, one linear regressor, and one polynomial regressor. The Power Laws are expressed in the form $W = ad^b$, depending solely on the fault throw, and are derived through direct fitting to carbonate samples collected in the field. These models are categorized into analysis scenarios: a minimum case ($a = 1.0$), a medium case ($a = 3.397$), and a maximum case ($a = 5.795$), according to the nature of the analysis being conducted regarding the damage zone. The exponent $b = 0.5877$ is fixed across all three scenarios.

The polynomial regression model is developed through training (70%) and testing (30%) using the full dataset, which incorporates all relevant parameters. This model is obtained through a multi-objective optimization process that simultaneously considers two conflicting objectives: the root mean squared error (RMSE) and the number of polynomial terms.

The linear regression model ($W = ad + b$) is constructed in a similar manner. In this case, two polynomial regression models are trained to predict the values of a and b based on the remaining parameters of the model ($E, \nu, \phi, c, c_p, \psi$), allowing users to assess the influence of each variable on the coefficients of the linear relation.

To address the multi-objective optimization challenge, four distinct evolutionary strategies grounded in differential evolution (DE) [28] are utilized. The methods adopted include: the third evolutionary phase of the generalized differential evolution algorithm (GDE-3) [29]; the adaptive multi-objective differential evolution technique with success-history (SHAMODE) [30]; a hybrid version of SHAMODE integrated with whale optimization, referred to as SHAMODE-WO [30]; and a meta-heuristic multi-objective approach that applies iterative parameter distribution estimation, known as MM-IPDE [31].

The selection of the most suitable candidate vectors for progression to the next generation is guided by the

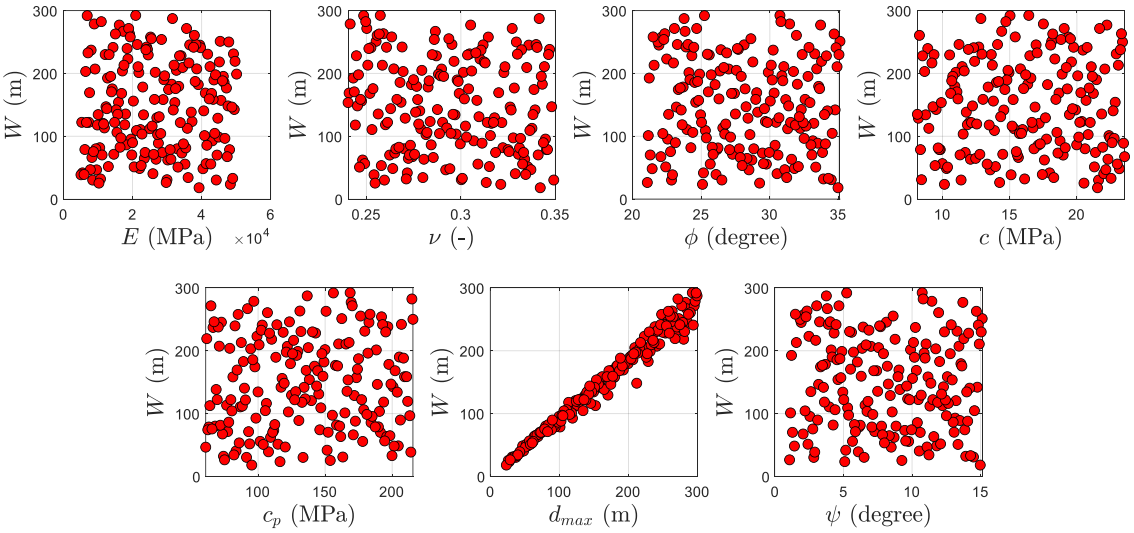


Figure 2. Database feature correlations with target.

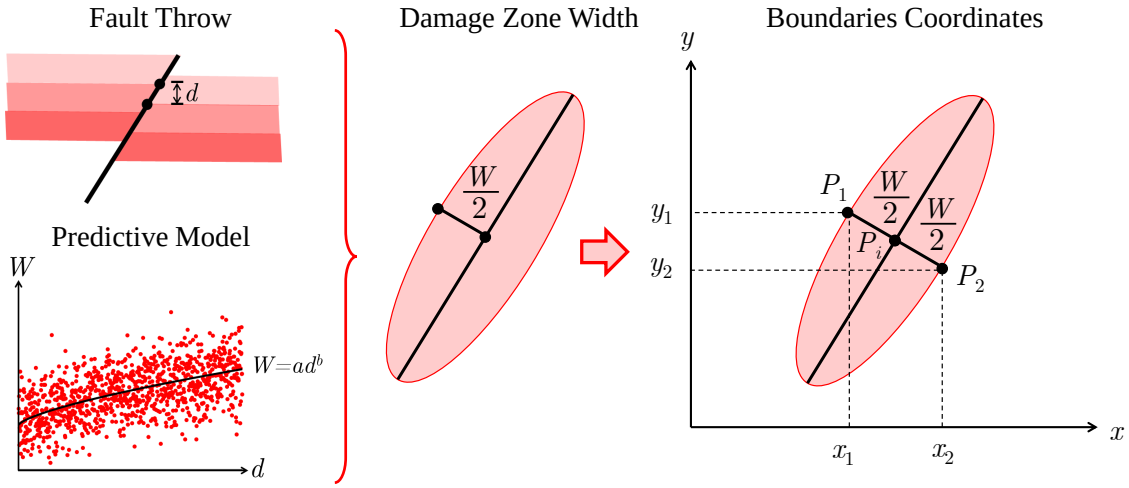


Figure 3. Damage zone boundaries coordinates determination scheme.

dominance and crowding distance principles from the Non-dominated Sorting Genetic Algorithm II (NSGA-II), as introduced by Deb et al. [32]. Table 2 presents the performance metrics of the two regression models trained on the dataset.

Table 2. Performance metrics for linear and polynomial models.

Model	R^2		$RMSE$		$MAPE$	
	Train	Test	Train	Test	Train	Test
Linear	0.99	0.99	5.21	5.07	2.56%	2.53%
Polynomial	0.95	0.95	14.99	14.44	6.21%	6.47%

3 Graphical User Interface

The application features a simple graphical user interface primarily designed to display the geometric shape of the damage zone for a given set of six features and one fault throw variable (totaling seven variables). The fault throw, which varies along the fault, is provided by the input file derived from a geological model containing raw field-measured data. In addition, the application must be efficient enough to update the damage zone geometry in real time and refresh the visualizer as the sliders associated with each feature are adjusted (triggering the "value

changing" signal). The graphical interface includes four panels: the visualization axis, the parameter management panel, the panel that provides information about the predictive functions, the damage envelope visualization and the maximum observed damage zone width, and the solver panel. The entire application was developed in MATLAB, and its graphical interface was designed using App Designer. Figure 4 illustrates the application's graphical interface and identifies the panels mentioned above.

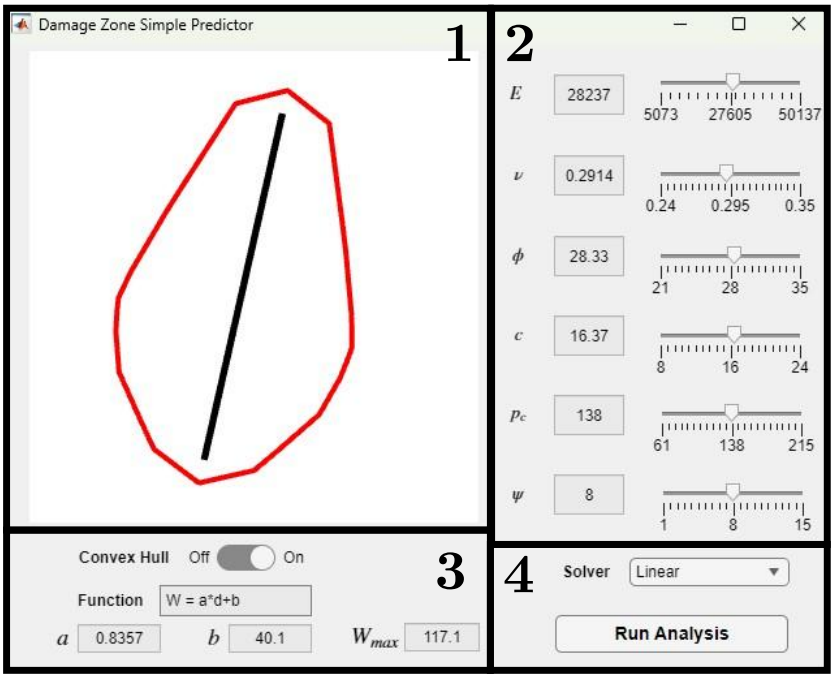


Figure 4. Graphical User Interface (GUI).

The name and function of the six panels are:

(**1**) **Visualization Axis**: The primary function of the Visualization Axis is to render the fault as a black line and the boundary of the damage zone as a red polyline. Users can choose to display either the convex or non-convex envelope of the damage zone. This flexibility is important because fault throw values obtained from raw field data often lead to irregular or noisy geometries that may not be well-behaved. As such, the interface allows users to visualize either the actual damage zone width, as derived from the raw fault throw data, or a smoothed approximation based on a convex geometric envelope;

(**2**) **Parameters Panel**: This panel contains all the variables that influence the geometry of the damage zone, with the exception of fault throw, since this variable is intrinsic to the input data file and should not be modified by the user. Here, users have the flexibility to either input known values for any of these properties or adjust them interactively using sliders, observing in real time how these changes affect the damage zone geometry in the visualizer;

(**3**) **Function Information Panel**: This panel displays information regarding the predictive functions and their parameters, such as coefficients a and b for both linear and power-law models. In the case of power-law models, which depend solely on the fault throw, the parameters a and b are fixed for each of the three predefined scenarios (minimum, medium, and maximum). For the linear model, however, these parameters depend on the other editable variables provided in the Parameters Panel and are updated dynamically as the sliders are adjusted. This panel also indicates which predictive function is currently being used and allows the user, via a radio button, to toggle between convex and non-convex representations of the damage zone boundary.

(**4**) **Solver Panel**: In this section of the interface, users can select the predictive function to be applied using a combo box. The available options include three power-law regressors, one linear model, and one polynomial model. Additionally, users can manually execute the Run Analysis command by pressing a button, which is particularly useful when parameters are manually defined. However, if sliders are used to adjust the parameters, the callback linked to the Run Analysis function is automatically triggered, rendering the button unnecessary.

This initial version of the interface is still a prototype under development and does not yet include functionality for loading external input files, relying instead on a single default file. File loading capabilities will be

implemented in future versions of the application.

4 Conclusions and future works

This work presented the development of an interactive and user-friendly graphical interface designed for real-time analysis of the sensitivity of damage zone width to geomechanical parameters. The application integrates five predictive models, three based on power laws, one on linear regression, and one on polynomial regression, trained using a synthetic dataset generated through numerical simulations.

The interface allows users to dynamically adjust the input parameters and instantly visualize the resulting changes in the geometry of the damage zone. Additionally, the tool provides real-time updates on the variation of the coefficients a and b of the linear model as a function of the geomechanical parameters, and it supports the visualization of the damage zone boundary using either convex or non-convex envelopes.

Developed in MATLAB using the App Designer environment, the application is currently at a prototype stage, with several short-term enhancements already planned. These include support for importing external input files, implementation of multiple simultaneous visualizers, and the integration of new predictive models such as artificial neural networks. In the long term, future developments aim to incorporate a 3D visualizer for fault points derived from geological models, enable the selection of preferential planes for 2D visualization, and define a three-dimensional damage volume that responds in real time to variations in the problem's geomechanical features.

Acknowledgements. This research was carried out in association with the ongoing R&D project registered as ANP n° 23507-7, "GeoBanD 2 – Geomodelagem numérica de zonas de falhas" (PUC-Rio/CENPES/ANP), sponsored by Petrobras. The authors also gratefully acknowledge the support from Carlos Chagas Filho Foundation for Supporting Research in the State of Rio de Janeiro (FAPERJ) Grant E-26/211.766/2021, Brazilian National Council for Scientific and Technological Development (CNPq) Grants 141617/2020-9, 420074/2021, and 407388/2022-2, and Tecgraf/PUC-Rio Institute.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] H. Fossen. *Structural Geology*. Cambridge University Press, 2010.
[2] K. Oliveira, R. Quevedo, D. Roehl, and B. Carvalho. Numerical modeling of the linking damage zone between geologic faults. In *XLIII Ibero-Latin-American Congress on Computational Methods in Engineering*, pp. 7, 2022.
[3] M. Kardani, M. Nazem, D. Sheng, and J. Carter. Large deformation analysis of geomechanics problems by a combined rh-adaptive finite element method. *Computers and Geotechnics*, vol. 49, pp. 90–99, 2013.
[4] M. Nazem, J. Carter, and D. Airey. Arbitrary lagrangian–eulerian method for dynamic analysis of geotechnical problems. *Computers and Geotechnics*, vol. 36, pp. 549–557, 2009.
[5] D. Wang, B. Bienen, M. Nazem, Y. Tian, J. Zheng, T. Pucker, and M. Randolph. Large deformation finite element analyses in geotechnical engineering. *Computers and Geotechnics*, vol. 65, pp. 104–114, 2015.
[6] C. Augarde, S. Lee, and D. Loukidis. Numerical modelling of large deformation problems in geotechnical engineering: A state-of-the-art review. *Soils and Foundations*, vol. 61, pp. 1718–1735, 2021.
[7] A. Torabi, J. Rudnicki, B. Alaei, and G. Buscarnera. Envisioning faults beyond the framework of fracture mechanics. *Earth-Science Reviews*, vol. 238, pp. 104358, 2023.
[8] S. Ahmed. *Structural analysis of Horda Platform and Stord Basin in the Norwegian North Sea using Machine Learning methods*. PhD thesis, University of Oslo, 2020.
[9] W. Bonke. *Identification and Characterization of Faults Using Deep Learning*. PhD thesis, University of Oslo, 2022.
[10] A. Ramesh, N. Satyavani, and M. Attar. Improved feature extraction in seismic data: multi-attribute study from principal component analysis. *Geo-Marine Letters*, vol. 41, 2021.
[11] L. Xu and E. C. Green. Inferring geological structural features from geophysical and geological mapping data using machine learning algorithms. *Geophysical Prospecting*, vol. 71, n. Advanced Techniques, Methods and Applications for an Integrated Approach to the Geophysical Prospecting, pp. 1728–1742, 2023.
[12] H. Garcia, P. Szafian, C. Han, R. Williams, and J. Lowell. Convolutional neural networks for fault interpretation – case study examples around the world. In *Advances in Subsurface Data Analytics: Traditional and Physics-Based Machine Learning*, pp. 141–164, 2022.

[13] H. Garcia and L. Plant. Application of artificial intelligence tools for fault imaging in an unconventional reservoir: A case study from the permian basin. In *SPE/AAPG/SEG Unconventional Resources Technology Conference (URTC)*, 2021.
[14] W. Ju, G. Hou, and B. Zhang. Insights into the damage zones in fault-bend folds from geomechanical models and field data. *Tectonophysics*, vol. 610, pp. 182–194, 2014.
[15] H. Di, D. Gao, and G. Alregib. Developing a seismic texture analysis neural network for machine-aided seismic pattern recognition and classification. *Geophysical Journal International*, vol. 218, pp. 1262–1275, 2019.
[16] L. Mosser, S. Purves, and E. Naeini. Deep bayesian neural networks for fault identification and uncertainty quantification. In *82nd EAGE Conference and Exhibition*, pp. 2262–2266, 2021.
[17] L. Mosser, T. Papadopoulos, A. Kuha, J. Herredsvela, and E. Naeini. A comparison of traditional, supervised, and unsupervised machine learning-based denoising methods for post-stack seismic data. In *2nd EAGE Digitalization Conference and Exhibition*, 2022.
[18] X.-L. Wei, C.-X. Zhang, S.-W. Kim, K.-L. Jing, Y.-J. Wang, S. Xu, and Z.-Z. Xie. Seismic fault detection using convolutional neural networks with focal loss. *Computers & Geosciences*, vol. 158, 2022.
[19] S. Ahmed and H. Guneyli. Improved deep learning (inception) cnn for detecting small tectonic plates of broken anatolian, turkey. *All Earth*, vol. 33, pp. 52–65, 2021.
[20] Y. Guo, S. Peng, W. Du, and D. Li. Fault and horizon automatic interpretation by cnn: A case study of coalfield. *Journal of Geophysics and Engineering*, vol. 17, pp. 1016–1025, 2020.
[21] S. Li, J. Chen, and J. Xiang. Applications of deep convolutional neural networks in prospecting prediction based on two-dimensional geological big data. *Neural Computing and Applications*, vol. 32, pp. 2037–2053, 2020.
[22] X. Zhao and J. Mendel. Minimum-variance deconvolution using artificial neural networks. In *SEG Technical Program Expanded Abstracts*, pp. 738–741, 1988.
[23] G. Röth and A. Tarantola. Neural networks and inversion of seismic data. *Journal of Geophysical Research: Solid Earth*, vol. 99, n. B4, pp. 6753–6768, 1994.
[24] Z. Bi, X. Wu, Z. Geng, and H. Li. Deep relative geologic time: A deep learning method for simultaneously interpreting 3-d seismic horizons and faults. *Journal of Geophysical Research: Solid Earth*, vol. 126, 2021.
[25] X. Wu, Y. Shi, S. Fomel, and L. Liang. Convolutional neural networks for fault interpretation in seismic images. In *SEG Technical Program Expanded Abstracts 2018*, pp. 1946–1950, 2018.
[26] Z. Jiang, S. Zhang, C. Turnadge, and T. Xu. Combining autoencoder neural network and bayesian inversion to estimate heterogeneous permeability distributions in enhanced geothermal reservoir: Model development and verification. *Geothermics*, vol. 97, pp. 102262, 2021.
[27] J. O. Adegbite, H. Belhaj, and A. Bera. Investigations on the relationship among the porosity, permeability and pore throat size of transition zone samples in carbonate reservoirs using multiple regression analysis, artificial neural network and adaptive neuro-fuzzy interface system. *Petroleum Research*, vol. 6, n. 4, pp. 321–332, 2021.
[28] R. Storn. Differrential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical report, International Computer Science Institute*, vol. 11, 1995.
[29] S. Kukkonen and J. Lampinen. GDE3: The third evolution step of generalized differential evolution. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pp. 443–450. IEEE, 2005.
[30] N. Panagant, S. Bureerat, and K. Tai. A novel self-adaptive hybrid multi-objective meta-heuristic for reliability design of trusses with simultaneous topology, shape and sizing optimisation design variables. *Structural and Multidisciplinary Optimization*, vol. 60, n. 5, pp. 1937–1955, 2019.
[31] K. Wansasueb, N. Pholdee, N. Panagant, and S. Bureerat. Multiobjective meta-heuristic with iterative parameter distribution estimation for aeroelastic design of an aircraft wing. *Engineering with Computers*, vol. 38, n. 1, pp. 695–713, 2022.
[32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, vol. 6, n. 2, pp. 182–197, 2002.