Hybrid computational model to predict Total Organic Carbon

Clovis Antonio da Silva
Computational Modeling Program
Polytechnic Institute, Rio de Janeiro State University
Nova Friburgo, Brazil
casilva@iprj.uerj.br

Juliana da Costa Cabral
Computational Modeling Program
Polytechnic Institute, Rio de Janeiro State University
Nova Friburgo, Brazil
juliana.costa@iprj.uerj.br

Grazione de Souza Boy
Department of Computational Modeling
Polytechnic Institute, Rio de Janeiro State University
Nova Friburgo, Brazil
gsouza@iprj.uerj.br

Camila Martins Saporetti
Department of Computational Modeling
Polytechnic Institute, Rio de Janeiro State University
Nova Friburgo, Brazil
camila.saporetti@iprj.uerj.br

***Abstract*—One important metric for assessing the amount of organic matter in source rocks is the Total Organic Carbon (TOC) content of rock samples. On the other hand, calculating TOC requires the collecting and analysis of samples from numerous well intervals in the source rock formations, which is a very resource-intensive process. Researchers have looked for creative ways to simplify TOC estimation in order to overcome this difficulty. Among these, machine learning techniques have demonstrated potential as an alternative to traditional well log analysis and stratigraphic study. This work focuses on automating TOC estimate utilizing advanced machine learning techniques enhanced by a hybrid methodology to improve the models' accuracy and adaptability. Four metaheuristic algorithms—Arithmetic Optimization Algorithm, Coronavirus Herd Immunity Optimizer, Differential Evolution, and Particle Swarm Optimization—were used to optimize the machine learning models. Four machine learning techniques—Decision Tree, Extreme Learning Machine, Gradient Boosting, and K-Nearest Neighbors—incorporated these metaheuristics. Core samples from the Shahejie Formation, Dongying Depression, Bohai Bay, China were used to evaluate the hybrid technique. The findings show that a hybrid approach combined with machine learning models produces extremely accurate and flexible models for TOC prediction. Regardless of the metaheuristic that was applied to direct the model selection process, the optimized Extreme Learning Machine produced the best performance metrics. These results demonstrate how hybrid models can improve exploratory geological research by providing a more accurate and efficient way to estimate TOC in source rocks.**

***Index Terms*—total organic carbon, machine learning, metaheuristics, hybrid models.**

I. Introduction

Finding oil and gas deposits depends on figuring out how much and what kind of organic matter is present in the rocks that are created [1]. Exploiting these hydrocarbon resources requires this task. Total organic carbon (TOC), one of the performance requirements for potential reservoirs, is a crucial measure of the amount and caliber of organic matter present in the source rock. The most precise technique for measuring organic matter directly and determining the TOC of a sedimentary rock is geochemical analysis. It can be manually computed by analyzing the original rock samples [2].

However, because it relies on samples taken from many well intervals in the source rocks, this widely used method is impractical [3]. But sometimes there aren't enough samples to gather or the core samples aren't available, thus the analysis is done using drilling fragments. Geochemical techniques might not fully capture the geological formation in this situation [4]. To overcome these obstacles, a number of research have developed a mathematical connection between well geophysical characteristics and TOC [5].

Research on alternative data-driven approaches is becoming more and more crucial as oil and gas exploration increases and more effective methods for determining TOC using petrophysical or petrographic data are developed [6]–[8]. Numerous research have demonstrated the efficacy of using hybrid machine learning techniques to forecast TOC [9]–[13]. Few research, nevertheless, have employed these techniques to forecast TOC in different hybridization metaheuristics. Recently, a number of metaheuristics with various methodologies have been put out; nevertheless, they have not yet been assessed for the tasks at hand. By contrasting several approaches with traditional metaheuristic measurements, this research advances the literature by giving a computational framework for automating parametric selection. This method builds on the work done recently on TOC estimate with intelligent technology by combining three machine learning models and seven metaheuristics. Understanding the deposition circumstances of older sediments requires precise TOC modeling. It also contributes significantly to the reconstruction of past occurrences. This is essential for determining whether rocks could be an oil source.

Few research, nevertheless, have employed these techniques to forecast TOC in different hybridization metaheuristics. Recently, a number of metaheuristics with various method-

ologies have been put out; nevertheless, they have not yet been assessed for the tasks at hand. By contrasting several approaches with traditional metaheuristic measurements, this research advances the literature by giving a computational framework for automating parametric selection. This method builds on the work done recently on TOC estimate with intelligent technology by combining three machine learning models and seven metaheuristics. Understanding the deposition circumstances of ancient sediment pools requires precise TOC modeling. It also contributes significantly to the reconstruction of past occurrences. This is essential for determining whether rocks could be an oil source.

The following is a list of the paper's most significant contributions:

- to suggest a hybrid strategy that finds the best models for TOC modeling by fusing machine learning methods with both contemporary and traditional metaheuristics;
- to provide the most accurate computational method for petrophysical data-based TOC value prediction.
- improve the state of the art on the use of hybrid models in the Bohai Bay Basin's Dongying Depression.

The dataset is introduced and the hybrid computational methodology is thoroughly explained in Sections II and III. Informations about the computational framework are presents in Section IV. The results are shown in Section V, which also discusses the suggested method after emphasizing the metaheuristics' performance evaluation. Finally, the conclusion is presented in Section VI.

II. Dataset

Bohai Bay Basin is located on the eastern coast of China. The Dongying Depression area lies in the southeast of Jiyang Depression in Bohai Bay Basin. A total of 125 core samples composed of well logging data and geochemical index were analyzed, taken from EsL 3 and EsU 4 member of Liye 1 well with the coring depths between 3500 m and 3800 m. Gamma ray (GR), resistivity (RT), transit interval time (AC), density (DEN), and neutron (CNL) are the feature variables in the dataset [14]. Figures 1 and 2 shows the correlation matrices for the training and test set, respectively. It can be observed that there is a good positive correlation between CNL and AC, a negative correlation between DEN and CNL, DEN and AC.

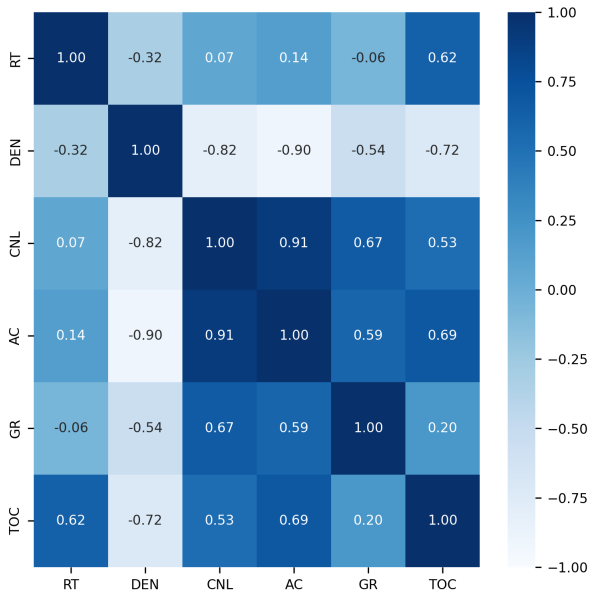

Fig. 1. Training set correlation matrix.

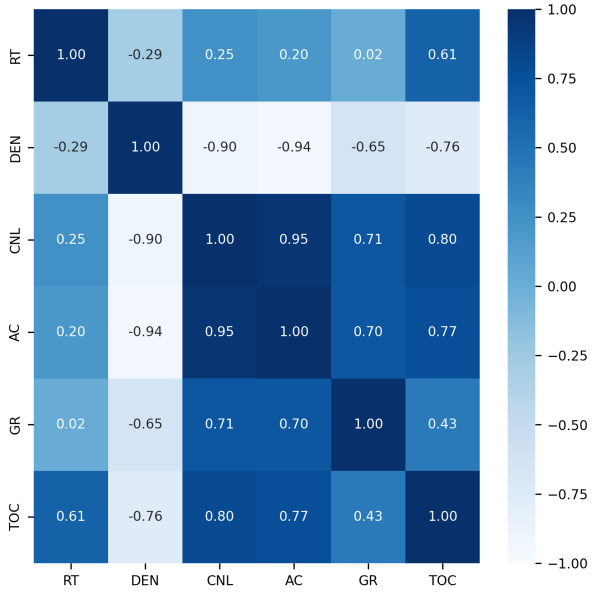

Fig. 2. Test set correlation matrix.

III. Methods

This section presents the metaheuristics and the machine learning methods applied. The choice of methods was based on testing different approaches to optimizers and regression methods that were proposed at different times, allowing us to identify the best option to solve the proposed problem.

A. Metaheuristics

Metaheuristics (MHAs) are search methods applied to optimization problems, which seek to efficiently explore the solution space of a problem by applying different mechanisms with the purpose of avoiding confinement in local minima or maxima. In this work, the Arithmetic Optimization Algorithm (AOA), Coronavirus Herd Immunity Optimizer (CHIO), Differential Evolution (DE), and Particle Swarm Optimization (PSO) methods were used to hyperparameters tuning of machine learning techniques.

PSO is an MHA inspired by the social behavior of schools of fish or flocks of birds. It was proposed by James Kennedy and Russell Eberhart in 1995 [15]. In this method, each particle (candidate solution to the problem) has a localization in the search space and moves towards the optimal point according to two main factors: pbest (best position of an individual particle)

and gbest (best position found by any particle). The parameters inertia weight, cognitive component and social component control the velocity of the particles at each iteration.

DE method was proposed by Rainer Storn and Kenneth Price in 1997 [16]. It generates new candidate solutions using vector operations between individuals x_i of the population. At each new generation (iteration), a mutant vector v_i is generated for each x_i from vector differences. Then, a vector u_i is created by combining x_i and v_i using a crossover rate chosen in the interval [0,1]. The fitness of u_i is compared with that of x_i and the best one becomes the x_i of the next generation. When the maximum number of generations is reached, the best vector found is the optimal solution.

The AOA algorithm uses the addition (A) and subtraction (S) operators in the exploitation phase and the exploration is carried out by the division (D) and multiplication (M) operators, uses the $\mu = 0.5$ and $\alpha = 5$ parameters to control the exploration and exploitation phases, respectively, and also uses random numbers R1, R2, R3, with values in the range $[0, 1]$. The Math Optimizer Accelerated (MOA) function select the search phase (exploitation or explanation), and the Math Optimizer Probability (MOP) function is used in the A, S, M and D operators to update the positions of the candidate solutions [17].

CHIO is an optimization algorithm inspired by the coronavirus (COVID-19) pandemic and the concepts of social distancing and herd immunity [18]. At each iteration, individuals are classified as susceptible, infected, or immune. The main goal is to increase overall herd immunity, converging when a sufficient proportion of the population becomes immune.

B. Machine Learning Techniques

K-Nearest Neighbors (KNN) was first developed by Fix [19] and later expanded by Cover [20]. KNN is a lazy learning machine learning technique, meaning it does not produce a model from the training data, but rather memorizes the entire training set and makes predictions for new data (x_{new}) based on this information. First, the metric (usually the Euclidean distance) and the k value of neighbors are chosen. Then, the distance between x_{new} and the points x_i in the training set are calculated. The k nearest neighbors are the k points with the smallest distances. Finally, the output for the new data is calculated by averaging the values of the k neighbors.

In Decision Trees Regression [21], the leaf nodes are continuous numerical values, and each internal node of the tree is associated with a predictive variable in the training set. Each path from the root of the tree to a leaf is a rule, and the output of an input x_{new} is defined by following a specific path, according to the input values. One of the great advantages of decision trees is the ease of interpreting how they decide what the output value for a new input is (white-box model). To control overfitting, the hyperparameters maximum tree depth, the minimum number of samples for each leaf node, and the minimum number of samples for splitting an internal node were adjusted.

The Gradient Boosting (GB) method for regression was introduced in [22]. This method combines weak learners (decision trees with low depth) sequentially to obtain a strong learner. Each new regression tree constructed tries to minimize the error (residual) of the previous tree, and the weight of each tree's contribution to the final prediction is defined by the learning rate. The value of the dependent variable is the average of the values of the trees and the number of trees constructed is adjusted to avoid overfitting.

Extreme Learning Machines (ELM) [23] are neural networks with a single hidden layer of the feedforward type. In these neural networks, the weights that connect the input layer to the hidden layer are randomly assigned, with their values remaining fixed. And the weights that connect the hidden layer to the output layer are obtained from the training data. The model is adjusted by the number of neurons in the hidden layer and the chosen activation function.

C. The $\Delta LogR$ Method

The $\Delta LogR$ method was developed by [24] in 1990. This technique is used to estimate TOC in source rocks using well logs. This is done by identifying intervals of potentially hydrocarbon-producing rocks in a well, and quantifying TOC continuously in these intervals. The mathematical expression for calculating $\Delta LogR$ is

$$\Delta LogR = log\left(\frac{R}{R_{baseline}}\right) + 0.02 * (\Delta t - \Delta t_{baseline}) \quad (1)$$

The empirical formula that relates $\Delta LogR$ to TOC is

$$TOC = (\Delta LogR) * 10^{(2.297 - 0.1688 * LOM)} \quad (2)$$

D. Performance Metrics

Table I displays the performance metrics along with their corresponding descriptions.

TABLE I
PERFORMANCE METRICS. R IS CORRELATION COEFFICIENT. R^2 IS DETERMINATION COEFFICIENT. MSE, RMSE AND MAE, ARE THE MEANS SQUARED ERROR, ROOT MEAN SQUARED ERROR AND MEAN ABSOLUTE ERRORS, RESPECTIVELY. yt_i AND yp_i INDICATE THE TRUE AND ESTIMATED VALUES, RESPECTIVELY. $\bar{yt}$ AND $\bar{yp}$ IS THE AVERAGE OF TRUE AND ESTIMATED TOC.

Metric	Formula
R	$\frac{\sum_{i=1}^{N}(yt_i - \bar{yt_i})(yp_i - \bar{yp_i})}{\sqrt{\sum_{i=1}^{N}((yt_i - \bar{yt})^2)\sum_{i=1}^{N}((yp_i - \bar{yp})^2)}}$
R^2	$\frac{\sum_{i=1}^{N}(yt_i - yp_i)^2}{\sum_{i=1}^{N}(yt_i - \bar{yt})^2}$
RMSE	$\frac{1}{N}\sqrt{\sum_{i=1}^{N} yt_i - yp_i)^2}$
MAE	$\frac{1}{N}\sum_{t=1}^{N}\|(yt_i - yp_i)\|$
MAPE	$100 \times \frac{1}{N}\sum_{i=1}^{N}\frac{\|yt_i - yp_i\|}{\|yt_i\|}$

IV. Computational Framework

The encoding of potential solutions for any machine learning technique is shown in Table II. Four internal parameters $(\theta_1, \theta_2, \theta_3, \theta_4)$ are included in the candidate solution for the DT model. The maximum depth is the first, followed by minimum samples split, minimum samples leaf, and criterion. The number of neurons in the hidden layer is the first of the three internal parameters $(\theta_1, \theta_2, \theta_3)$ for the ELM model. The regularization parameter is the second, and the activation function is the third. The GB encodes the number of estimators (θ_1), learning rate (θ_2), maximum depth (θ_3) and criterion (θ_4). Finally, the three parameters also influence the KNN model encoding the number of neighbors (θ_1), weigths (θ_2), and the power for the Minkowski metric (θ_3).

TABLE II
Encoding of the internal parameters for each ML model.

Estimator	Encoding	Description	Settings/Range
DT	θ_1	Max depth	[1,20]
	θ_2	min. samples split	[1e-10, 1]
	θ_3	min. samples leaf	[1e-10,0.99]
	θ_4	Criterion	squared_error, poisson, friedman_mse, absolute_error
	θ_3	Number of neighbors	[2,20]
ELM	θ_1	No. neurons	[1, 200]
	θ_2	Activation function	tanh, sigm, relu, lin
	θ_3	alpha	[1e-9,10]
GB	θ_1	No. estimators	[1,200]
	θ_2	Learning rate	[0.0001, 1]
	θ_3	Max depth	[1,20]
	θ_4	Criterion	friedman_mse, squared_error
KNN	θ_1	Number of neighbors	[2,20]
	θ_2	Weights	distance, uniform
	θ_3	Power parameter for the Minkowski metric	[1,3]

After the candidate solutions are encoded, the metaheuristics generate a population of them that are iteratively evolved over a predefined number of epochs. Table III displays the parametric setting of the metaheuristics that were utilized to determine the ideal parameters. The objective function that was evaluated in every instance was the RMSE. After the search, the best model is used to forecast new TOC values from new samples.

The dataset was divided into 80% for the training set and 20% for the test set. K-fold (k=5) was used as a cross-validation technique in the training set. It was chosen 30 independent runs to be able to analyze the statistical difference between the methods. It could be to use the major number, but the consuming time would be longer.

V. Results and Discussion

The computational framework was implemented in Python 3.8 employing the packages mealpy [25]–[27], pandas [28], scikit-learn [29], scipy [30].

TABLE III
Metaheuristics configuration.

Metaheuristic	Parameter	Name	Value
AOA	NP	Population size	50
	E_{max}	No. epochs	50
CHIO	NP	Population size	50
	E_{max}	No. epochs	50
DE	NP	Population size	50
	E_{max}	No. epochs	50
	CR	Crossover probability	0.9
	WF	Weighting Factor	0.1
PSO	NP	Population size	50
	E_{max}	No. epochs	50
	w	Inertia weight	0.4
	c_1	Social component	2.05
	c_2	Cognitive component	2.05

In 30 individual runs of the techniques DT, ELM, GB, and KNN optimized by AOA, CHIO, DE, and PSO, the mean and standard deviation for the performance metrics in the training set are displayed in Table IV. All metaheuristics showed that ELM produced the best results, while AOA-ELM produced the best predictions (R^2 = 0.759 (0.018), RMSE = 0.797 (0.029), MAE = 0.596 (0.021)).

Figures 3, 4, 5 and 6 display the Taylor diagram for the each metaheuristics. Each diagram illustrates the standard deviation, correlation coefficient, and centered root mean square deviation (RMSD) of the models in comparison to the reference data. It can be seen the results suggest that most ML models achieve around 0.8 correlation, indicating agreement with the reference data. However, the standard deviation values for all models are slightly below 1.0, suggesting a mild smoothing effect, likely due to the regularization imposed by ML models.

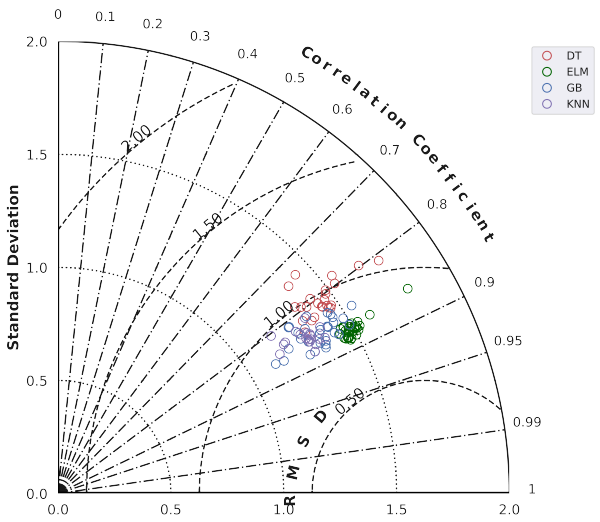

Fig. 3. Taylor diagram by AOA.

The p-values from the ANOVA test are shown in Table V. The alternative hypothesis suggests that the average varies for at least one particular metaheuristic, whereas the null hypothesis for the ML models states that the average of

TABLE IV
TABLE TYPE STYLES

Optimizer	Estimator	R	R^2	RMSE	MAE	MAPE	MSE
AOA	DT	0.803 (0.022)	0.634 (0.042)	0.983 (0.055)	0.735 (0.036)	22.258 (1.165)	0.968 (0.111)
AOA	ELM	**0.874** (0.007)	**0.759** (0.018)	**0.797** (0.029)	**0.596** (0.021)	**18.531** (0.597)	**0.636** (0.048)
AOA	GB	0.852 (0.019)	0.721 (0.032)	0.857 (0.048)	0.652 (0.024)	20.536 (0.638)	0.737 (0.083)
AOA	KNN	0.843 (0.017)	0.704 (0.030)	0.883 (0.045)	0.658 (0.030)	20.245 (0.868)	0.782 (0.080)
CHIO	DT	0.795 (0.042)	0.620 (0.077)	0.997 (0.095)	0.741 (0.047)	22.520 (1.345)	1.004 (0.205)
CHIO	ELM	**0.868** (0.014)	**0.752** (0.025)	**0.809** (0.039)	**0.610** (0.019)	**18.987** (0.496)	**0.657** (0.067)
CHIO	GB	0.840 (0.018)	0.701 (0.031)	0.887 (0.047)	0.671 (0.029)	20.968 (0.980)	0.790 (0.083)
CHIO	KNN	0.843 (0.016)	0.703 (0.029)	0.885 (0.043)	0.658 (0.029)	20.224 (0.765)	0.784 (0.076)
DE	DT	0.806 (0.018)	0.646 (0.031)	0.966 (0.042)	0.727 (0.027)	22.375 (0.889)	0.935 (0.082)
DE	ELM	**0.870**(0.016)	**0.751** (0.048)	**0.809** (0.069)	**0.600** (0.035)	**18.544** (0.726)	**0.659** (0.127)
DE	GB	0.849 (0.020)	0.718 (0.033)	0.861 (0.050)	0.654 (0.028)	20.550 (0.775)	0.744 (0.086)
DE	KNN	0.844 (0.021)	0.704 (0.034)	0.883 (0.051)	0.659 (0.035)	20.348 (1.020)	0.783 (0.090)
PSO	DT	0.810 (0.020)	0.645 (0.043)	0.967 (0.058)	0.733 (0.038)	22.492 (1.310)	0.939 (0.114)
PSO	ELM	**0.868** (0.011)	**0.751** (0.020)	**0.810** (0.032)	**0.607** (0.020)	**18.908** (0.576)	**0.657** (0.052)
PSO	GB	0.845 (0.019)	0.712 (0.032)	0.871 (0.049)	0.656 (0.031)	20.550 (0.797)	0.761 (0.085)
PSO	KNN	0.849 (0.017)	0.716 (0.027)	0.866 (0.042)	0.649 (0.029)	20.085 (0.761)	0.752 (0.073)

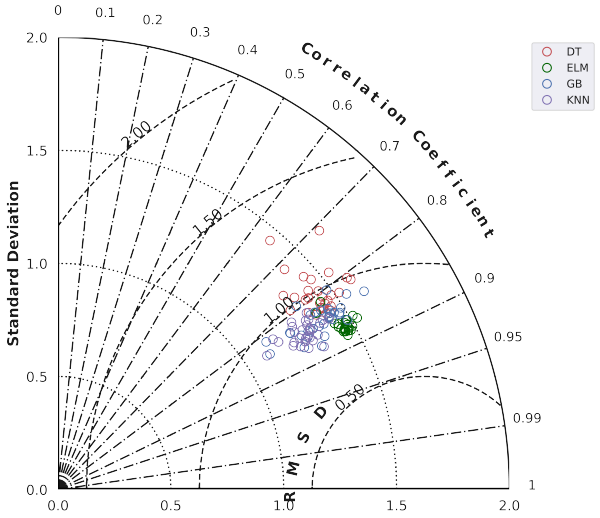

Fig. 4. Taylor diagram by CHIO.

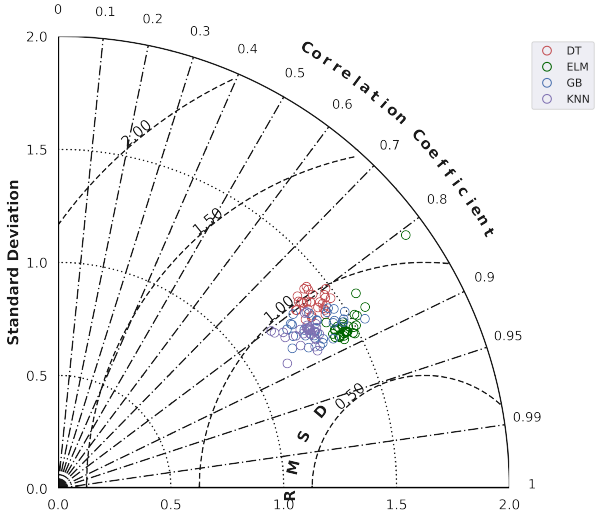

Fig. 5. Taylor diagram by DE.

each performance indicator is the same for all metaheuristics. There is no statistically significant difference in relation to the application of metaheuristics, according to the ML approaches, which produced p-values > 0.05 for all measures. The alternative hypothesis for the metaheuristics suggests that the average varies for at least one particular model, whereas the null hypothesis states that the average of each performance parameter is the same for all ML models. There is a statistically significant variation in the ML method selection for every case.

Table VI shows the best hyperparameters of ELM model for each metaheuristics and ΔlogR method in the test set. DE-ELM the best result (n_neurons = 104, alpha = 0.0214, ufunc = 'sigm') and RMSE = 0.750.

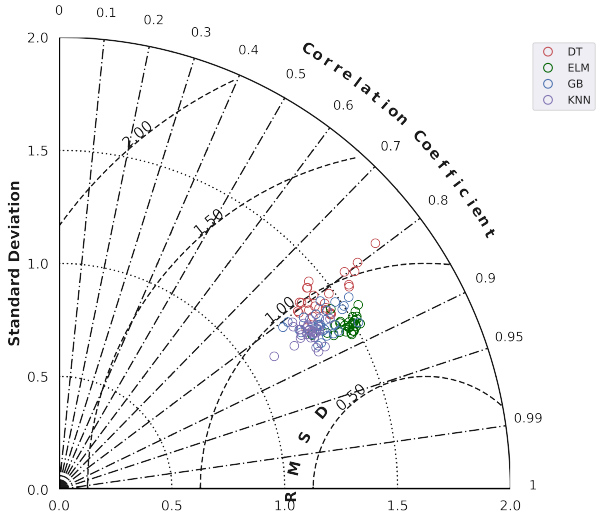

Fig. 6. Taylor diagram by PSO.

TABLE V
P-VALUES ANOVA

Estimator	R	R^2	RMSE	MAE	MAPE	MSE
DT	0.69	0.6	0.602	0.826	0.687	0.603
ELM	0.145	0.227	0.227	0.007	0.001	0.227
GB	0.09	0.1	0.1	0.095	0.355	0.1
KNN	0.576	0.469	0.469	0.678	0.719	0.469
Optimizer	R	R^2	RMSE	MAE	MAPE	MSE
AOA	0	0	0	0	0	0
CHIO	0	0	0	0	0	0
DE	0	0	0	0	0	0
PSO	0	0	0	0	0	0

VI. Conclusion

Using data of Shahejie Formation in Dongying Depression, China, this study examined how metaheuristics were applied in conjunction with four machine learning models to solve a TOC prediction problem. By enhancing a performance metric, four metaheuristics, some of which were newly proposed, were used to optimize the hyperparameter selection procedure. The goal was to minimize RMSE by optimizing the model selection, which would enable us to look for local minima throughout the parameter space. After optimization, the algorithms' performance was evaluated. The following performance metrics were used to quantify prediction errors:

TABLE VI
BEST MODELS

Optimizer	Hyperparameters	RMSE
AOA	n_neurons = 198, alpha = 0.0001, ufunc = 'lin'	0.755
CHIO	n_neurons = 19, alpha = 0.0041, ufunc = 'lin'	0.764
DE	n_neurons = 104, alpha = 0.0214, ufunc = 'sigm'	0.750
PSO	n_neurons = 31, alpha = 0.0002, ufunc = 'lin'	0.761
ΔlogR	LOM=10	13.490

R, R^2, RMSE, MAE, MAPE, and MSE. To further confirm the importance of the computational results, statistical tests were performed.

The main conclusions are listed below:

- The Extreme Learning Machine (ELM) model was the machine learning model that produced the best average results for all metaheuristics.
- Combining the ELM with the Arithmetic Optimization Algorithm (AOA) model produced the best performance metrics in training set.
- In test set, ELM with the Differential Evolution (DE) presented the best result.
- The ANOVA test for all cases there is a statistically significant difference in the choice of ML method.

Even though the metaheuristics produced the dataset's most accurate TOC predictions, further study may be able to apply them to a wider range of datasets from sedimentary basins with various diagenetic features. Furthermore, future research can expand to explainable Artificial Intelligence and feature selection methods, which may offer a more thorough comprehension of attempting to create relationships between factors in TOC prediction and pinpointing the most important step in the entire procedure.

Acknowledgment

This work was carried out with the support of the Coordination for the Improvement of Higher Education Personnel – Brazil (CAPES) – Financing Code 001. The authors thank the Carlos Chagas Filho Foundation for Research Support of the State of Rio de Janeiro (FAPERJ) for the financial support through 10.432/2024-APQ1.

References

[1] F. L. Staplin, "Sedimentary organic matter, organic metamorphism, and oil and gas occurrence," *Bulletin of Canadian Petroleum Geology*, vol. 17, no. 1, pp. 47–66, 1969.

[2] M. R. Shalaby, N. Jumat, D. Lai, and O. Malik, "Integrated toc prediction and source rock characterization using machine learning, well logs and geochemical analysis: case study from the jurassic source rocks in shams field, nw desert, egypt," *Journal of Petroleum Science and Engineering*, vol. 176, pp. 369–380, 2019.

[3] S. Larter and A. Aplin, "Reservoir geochemistry: methods, applications and opportunities," *Geological Society, London, Special Publications*, vol. 86, no. 1, pp. 5–32, 1995.

[4] S. A. Tedesco, *Surface geochemistry in petroleum exploration*. Springer Science & Business Media, 2012.

[5] B. E. Khesin, V. Alexeyev, and L. Eppelbaum, *Interpretation of geophysical fields in complicated environments*. Springer Science & Business Media, 2013, vol. 14.

[6] D. Zheng, S. Wu, and M. Hou, "Fully connected deep network: An improved method to predict toc of shale reservoirs from well logs," *Marine and Petroleum Geology*, vol. 132, p. 105205, 2021.

[7] S. A. Chan, A. M. Hassan, M. Usman, J. D. Humphrey, Y. Alzayer, and F. Duque, "Total organic carbon (toc) quantification using artificial neural networks: Improved prediction by leveraging xrf data," *Journal of Petroleum Science and Engineering*, vol. 208, p. 109302, 2022.

[8] S. Asante-Okyere, S. A. Marfo, and Y. Y. Ziggah, "Estimating total organic carbon (toc) of shale rocks from their mineral composition using stacking generalization approach of machine learning," *Upstream Oil and Gas Technology*, vol. 11, p. 100089, 2023.

[9] A. Sultan, "New artificial neural network model for predicting the toc from well logs," in *SPE Middle East Oil and Gas Show and Conference*. SPE, 2019, p. D021S001R002.

[10] S. Azizah, A. Haris *et al.*, "Evaluation of shale hydrocarbon potential in upper talang akar formation based on laboratory geochemical data analysis and total organic carbon (toc) modelling," in *IOP Conference Series: Earth and Environmental Science*, vol. 538, no. 1. IOP Publishing, 2020, p. 012069.

[11] Y. Chen, X. Deng, X. Wang, Q. He, D. Huang, L. Cheng, Y. Zhang, and Q. Li, "Application of a pso-svm algorithm for predicting the toc content of a shale gas reservoir: A case study in well z in the yuxi area," *Geophys Prospect Pet*, vol. 60, no. 4, pp. 652–663, 2021.

[12] J. Wang, Y. Xu, P. Sun, Z. Liu, J. Zhang, Q. Meng, P. Zhang, and B. Tang, "Prediction of organic carbon content in oil shale based on logging: a case study in the songliao basin, northeast china," *Geomechanics and Geophysics for Geo-Energy and Geo-Resources*, vol. 8, no. 2, p. 44, 2022.

[13] L. Zhu, X. Zhou, W. Liu, and Z. Kong, "Total organic carbon content logging prediction based on machine learning: A brief review," *Energy Geoscience*, vol. 4, no. 2, p. 100098, 2023.

[14] H. Wang, W. Wu, T. Chen, X. Dong, and G. Wang, "An improved neural network for toc, s1 and s2 estimation based on conventional well logs," *Journal of Petroleum Science and Engineering*, vol. 176, pp. 664–678, 2019.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. ieee, 1995, pp. 1942–1948.

[16] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, pp. 341–359, 1997.

[17] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer methods in applied mechanics and engineering*, vol. 376, p. 113609, 2021.

[18] M. A. Al-Betar, Z. A. A. Alyasseri, M. A. Awadallah, and I. Abu Doush, "Coronavirus herd immunity optimizer (chio)," *Neural Computing and Applications*, vol. 33, no. 10, pp. 5011–5042, 2021.

[19] E. Fix and J. L. Hodges, "Discriminatory analysis, nonparametric discrimination," 1951.

[20] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[21] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Boca Raton: Chapman and Hall/CRC, 1984.

[22] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[23] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, vol. 2. Ieee, 2004, pp. 985–990.

[24] Q. Passey, S. Creaney, J. Kulla, F. Moretti, and J. Stroud, "A practical model for organic richness from porosity and resistivity logs," *AAPG bulletin*, vol. 74, no. 12, pp. 1777–1794, 1990.

[25] N. Van Thieu and S. Mirjalili, "Mealpy: An open-source library for latest meta-heuristic algorithms in python," *Journal of Systems Architecture*, 2023.

[26] N. Van Thieu, S. D. Barma, T. Van Lam, O. Kisi, and A. Mahesha, "Groundwater level modeling using augmented artificial ecosystem optimization," *Journal of Hydrology*, vol. 617, p. 129034, 2023.

[27] A. N. Ahmed, T. Van Lam, N. D. Hung, N. Van Thieu, O. Kisi, and A. El-Shafie, "A comprehensive comparison of recent developed meta-heuristic algorithms for streamflow time series forecasting problem," *Applied Soft Computing*, vol. 105, p. 107282, 2021.

[28] J. Reback, W. McKinney, J. Van Den Bossche, T. Augspurger, P. Cloud, A. Klein, S. Hawkins, M. Roeschke, J. Tratner, C. She *et al.*, "pandas-dev/pandas: Pandas 1.0. 5," *Zenodo*, 2020.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[30] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, "Scipy 1.0: fundamental algorithms for scientific computing in python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.