Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 25, Num. 04 (2018) 43-56

RESEARCH ARTICLE

Extreme Learning Machine combined with a Differential Evolution algorithm for lithology identification

Máquinas de Aprendizado Extremo combinadas com um algoritmo de Evolução Diferencial para a identificação litológica

Camila Martins Saporetti[1]*, Grasiele Regina Duarte[1], Tales Lima Fonseca[1], Leonardo Goliatt da Fonseca[2],Egberto Pereira[3]

Abstract: Lithology identification, obtained through the analysis of several geophysical properties, has an important role in the process of characterization of oil reservoirs. The identification can be accomplished by direct and indirect methods, but these methods are not always feasible because of the cost or imprecision of the results generated. Consequently, there is a need to automate the procedure of reservoir characterization and, in this context, computational intelligence techniques appear as an alternative to lithology identification. However, to acquire proper performance, usually some parameters should be adjusted and this can become a hard task depending on the complexity of the underlying problem. This paper aims to apply an Extreme Learning Machine (ELM) adjusted with a Differential Evolution (DE) to classify data from the South Provence Basin, using a previously published paper as a baseline reference. The paper contributions include the use of an evolutionary algorithm as a tool for search on the hyperparameters of the ELM. In addition, an activation function recently proposed in the literature is implemented and tested. The computational approach developed here has the potential to assist in petrographic data classification and helps to improve the process of reservoir characterization and the production development planning

Keywords: Extreme Learning Machines — Differential Evolution — Lithology

Resumo: A identificação litológica, obtida através da análise de várias propriedades geofísicas, tem um papel importante no processo de caracterização de reservatórios de petróleo. A identificação pode ser realizada por métodos diretos e indiretos, mas esses métodos nem sempre são viáveis devido ao custo ou imprecisão dos resultados gerados. Consequentemente, existe a necessidade de automatizar o procedimento de caracterização do reservatório e, neste contexto, as técnicas de inteligência computacional aparecem como uma alternativa à identificação litológica. No entanto, para obter um desempenho adequado, geralmente alguns parâmetros devem ser ajustados e isso pode se tornar uma tarefa difícil, dependendo da complexidade do problema subjacente. Este trabalho tem como objetivo aplicar uma Máquina de Aprendizagem Extrema (ELM) ajustada com uma Evolução Diferencial (DE) para classificar os dados da Bacia do Sul da Provença, usando um artigo publicado anteriormente como referência. As contribuições do artigo incluem o uso de um algoritmo evolucionário como ferramenta de busca nos hiper parâmetros do ELM. Além disso, uma função de ativação recentemente proposta na literatura é implementada e testada. A abordagem computacional desenvolvida aqui tem o potencial de auxiliar na classificação de dados petrográficos e ajuda a melhorar o processo de caracterização de reservatórios e o planejamento do desenvolvimento da produção

Palavras-Chave: Máquina de Aprendizado Extremo — Evolução Diferencial — Litologia

[1] *Post-Graduate in Computational Modeling, Federal University of Juiz de Fora, Brazil*
[2] *Department of Applied and Computational Mechanics, Federal University of Juiz de Fora, Brazil*
[3] *Department of Stratigraphy and Paleontology, University of the State of Rio de Janeiro, Brazil*
***Corresponding author**: camilasaporetti@ice.ufjf.br

DOI: https://doi.org/10.22456/2175-2745.80702 • **Received:** 01/03/2018 • **Accepted:** 06/09/2018

1. Introduction

The knowledge of the lithology of an oil well can be obtained through of the analysis of various geophysical features. This procedure is critical in the reservoir description process and enables to generate lithological patterns that are described by the petrophysical features and may then be applied in flow simulators for the purpose of evaluate the behavior of a reservoir. Changes in lithology usually is the main reason for differences in rock properties.

There are two kinds of conventional methods for the identification of lithology: direct and indirect. Lithology determination by direct observation of underground cores is an expensive process and is not always reliable and valid because different geologists may provide different interpretations. Indirect methods use well logs for quantifying the physical characteristics of geological formations providing most of the data available to a geologist. A well log is a record of the formations and any events that are encountered in the drilling process (Figure 2 shows an example of well logs of the oil well studied here). As well as their importance in conclusion decisions, they are also crucial instruments for mapping and identifying lithologies. Nevertheless, indirect methods do not obtain similar performance as direct methods. Manual interpretation of lithologies from well logs is a labor-intensive process that involves the expense of a considerable amount of time by an experienced well log analyst, even using the aid of graphical techniques like cross-plotting [1]. The problem becomes especially more difficult as the number of simultaneous logs to be analyzed increases. Therefore, it is required to automate the procedure of reservoir characterization and, at this point, computer technologies has shown suitable to lithology identification [2, 3, 4, 5]. These computer technologies assist the geologists to avoid the unnecessary data analysis work and improve the lithology identification accuracy [6]. As a result, geologists can build better quantitative evaluation models of different rock properties, which can also improve overall evaluation.

Machine learning approaches can potentially make the process of reservoir and rock formation identification more efficiently by providing the means to formalize the expert knowledge through know-how engineering [7]. Some research efforts found in the literature are described as follows. An unsupervised Self Organizing Map (SOM) of neural networks for the determination of oil well lithology and fluid contents was proposed by [8] and employed fuzzy inference rules derived from known characteristics of well logs were used in the interpretation of the clusters generated by the SOM neural networks. In [9], it was introduced kernel Fisher discriminant analysis (KFD), an improved Linear Discriminant Analysis (LDA) with kernel trick, to overcome the shortcoming of LDA for lithology identification. This procedure includes two processes: raising dimensions to get nonlinear information and reducing dimensions to get classification features. Cross plots and Principal Component Analysis were used to lithology characterization and mineralogy description from geochemical logging tool data [10]. In [6], five machine learning methods were employed to classify the formation lithology identification using well log data samples. Horrocks *et al* [2] explores different machine learning algorithms and architectures for classifying lithologies using wireline data for coal exploration. Other approaches include multivariate statistical analysis [11], neural networks with probabilistic neurons [12] or radial basis function kernel [13], random forests [14, 15], combination of classification and regression methods [16] and collaborative learning agents [7].

ELM networks may need a higher number of hidden neurons due to the random determination of the input weights and hidden biases. In [17] a hybrid learning algorithm which uses the differential evolutionary algorithm was proposed to select the input weights and Moore-Penrose (MP) generalized inverse to analytically determine the output weights. This approach proved to be able to achieve a good generalization performance with much more compact networks. An adaptive evolutionary ELM learning paradigm was developed by [18], for tool wear estimation in high-speed milling process. A Differential Evolution algorithm (DE) was used to select parameters optimized for the ELM. DE-ELM was used in [19] to classify hyperspectral images. Four sets of hyperspectral reference data were used and confirmed the attractive properties of the DE-ELM method in terms of classification accuracy and computation time. The results indicated that the proposed adaptive evolutionary ELM-based estimation model can effectively estimate the tool wear in high-speed milling process. In [20] a genetic neural network model was applied to predict lithology characteristic. The model exhibited good representation and strong prediction ability, and is suitable for recognition of lithology, lithofacies and sedimentary facies. The lithology identification from well log based on DE-SVM was proposed by [21]. The proposed method was considered feasible and produced satisfactory results.

In the literature, ELM integrated with DE have been used in several applications. There are some studies that used evolutionary algorithms combined with machine learning to identify lithologies. However, there are a lack of studies that used the Swish activation function in ELM implementations. Swish activation function has recently proposed by [22]. This paper explores an Extreme Learning Machine (ELM) [23] associated with a Differential Evolution (DE) [24] to classify data from the South Provence Basin, taken from [25], using the paper by [26] as a baseline reference. The output of the classifier is created from input data composed by the combination of mineralogy and textural information and divided into seven classes. The contribution of this paper includes the use of an evolutionary algorithm as a tool for search on the hyperparameters of the ELM. In addition, a recently proposed activation function called Swish [22] is implemented and its performance is compared with other well established activation functions in the literature. We have performed computational experiments and we have consistently achieved better results than those achieved by [26]. The remainder of this paper is organized as below. Section 2 describes the experimental data used in this study, the research methodology and evaluation methods. Section 3 presents modeling processes, discusses prediction results, and compares model performance. Concluding remarks and research contributions are given in the final section.

2. Material and Methods

2.1 Experimental Dataset

The well of interest in this paper is located within the Southern Basin of Provence, near Cassis and La Ciotat, called La Ciotat-1 (Figure 1). The database employed here, shown in Table 10, is a subset of samples of the database found in [25]. This subset was selected for purposes of comparison with [26]. The dataset was generated through core plugs extracted from land well La Ciotat-1 drilled down to 150 m from the surface. Figure 2 shows an example the well logs. The set of well logs include gamma-ray, P- and S-wave sonic, density, and resistivity data. This database is composed by elastic, mineralogical, and petrographic properties and was used in several studies ([26, 27, 28, 29]). Integrates ultrasonic measurements of P- and S-wave velocities at various effective pressures, density and porosity measurements, quantitative mineralogic analyses using X-ray diffraction (XRD), detailed petrographic studies of thin sections, and critical porosity and elastic properties of microporous mixed carbonate-siliciclastic rocks. Table 1 shows the petrographic classes and their description according to [25].

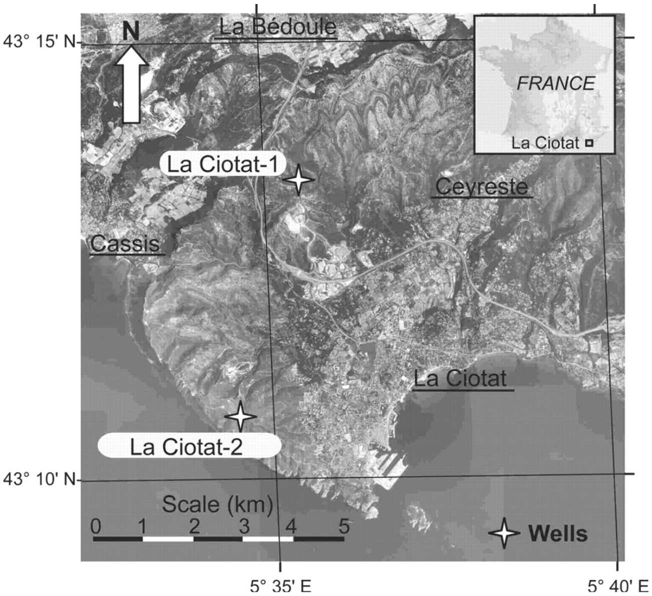


Figure 1. Location of La Ciotat-1 and La Ciotat-2 wells (extracted from [25]).

Table 1. Petrographic classes and their description according to [25].

Class	Description
C1	Limestone with grainstone texture (quartz $< 5\%$)
C2	Limestone with wackestone-packstone texture (quartz $< 5\%$)
C3	Quartz-rich limestone with sparitic/microsparitic intergranular space: grainstone texture or wackestone-packstone texture with recrystallized matrix (quartz $5\% - 50\%$)
C4	Quartz-rich limestone with micritic intergranular space: wackestone-packstone texture (quartz $5\% - 50\%$)
C5	Slightly argillaceous quartz-rich limestone with wackestone-packstone texture (quartz $5\% - 50\%$ and clay $2\% - 5\%$)
C6	Clean cemented sandstone (quartz $> 50\%$)
C7	Sandstone with carbonate micritic matrix (quartz $> 50\%$)

2.2 Extreme Learning Machines

The Extreme Learning Machine (ELM) [23] is a feed forward artificial neural networks, which has a single hidden layer. ELM strikes a balance between speed and generalization performance, and attracts more and more attention from various respects. Compared with the Artificial Neural Network (ANN), the Support Vector Machine (SVM) and other traditional prediction models, the ELM model retains the advantages of fast learning, good ability to generalize and convenience in terms of modeling [30]. In ELMs there are three levels of randomness [31]: (1) fully connected, hidden node parameters are randomly generated, (2) the connection can be randomly generated, not all input nodes are connected to a particular hidden node, and (3) a hidden node itself can be a subnetwork formed by several nodes resulting in learning local features. The output function of ELM used in this paper is given by

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{L} \beta_i G(\alpha, \mathbf{w}_i, b_i, \mathbf{c}, \mathbf{x}) = \sum_{i=1}^{L} \beta_i G(\alpha \mathrm{MLP}(\mathbf{w}_i, b_i, \mathbf{x}) + (1-\alpha)\mathrm{RBF}(\mathbf{x}, \mathbf{c}))$$

where $\hat{y}$ is the ELM prediction associated to the input vector $\mathbf{x}$, $\mathbf{w}_i$ is the weight vector of the i-th hidden node, b_i are the biases of the neurons in the hidden layer, β_i are output weights, $\mathbf{c}$ is the vector of centers. MLP and RBF are the input activation functions, respectively, while α is a user-defined that multiplies $\mathrm{MLP}(\cdot)$ and $\mathrm{RBF}(\cdot)$ terms. $G(\cdot)$ is the nonlinear output activation function and L is the the number of neurons in the hidden layer. The output activation functions

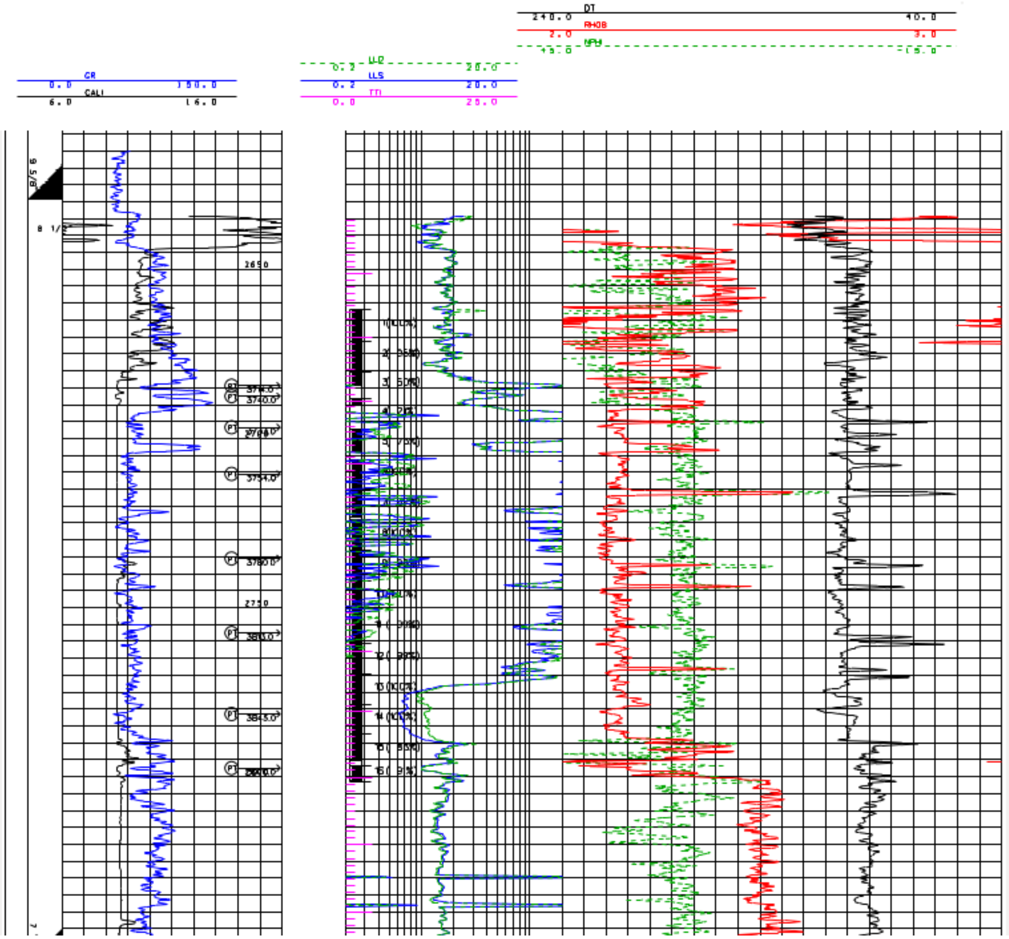


Figure 2. Well logs example (provide from Petroleum National Agency (ANP)).

$G(\alpha, \mathbf{w}_i, b_i, \mathbf{c}, \mathbf{x})$ with the hidden nodes weights $(\mathbf{w}, b)$ are presented in Table 2.

The parameters $(\mathbf{w}, b)$ are randomly generated (normally distributed with zero mean and standard deviation equals to one), and weights β_i of the output layer are determined analytically. MLP and RBF are the multilayer perceptron and Radial Basis Function activation functions, respectively, written as

$$\mathrm{MLP}(\mathbf{w}_i, b_i, \mathbf{x}) = \sum_{k=1}^{D} w_{ik} x_k + b_i \quad \text{and} \tag{1}$$

$$\mathrm{RBF}(\mathbf{x}, \mathbf{c}) = \sum_{j=1}^{D} \frac{x_j - c_{ij}}{r_i} \tag{2}$$

where D is the number of input features, the centers c_{ij} are taken uniformly from the bounding hyperrectangle of the input variables and $r = \max(||\mathbf{x} - \mathbf{c}||)/\sqrt{2D}$.

The output weight vector $[\beta_1, ..., \beta_L]$ can be determined by minimizing the approximation error [32]

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^L} ||\mathbf{H}\boldsymbol{\beta} - \mathbf{y}||$$

where $\mathbf{y}$ is the output data vector, $\mathbf{H}$ is the hidden layer output matrix

$$\mathbf{H} = \begin{bmatrix} G_1(\alpha, \mathbf{w}_1, b_1, \mathbf{c}, \mathbf{x}_1) & \cdots & G_L(\alpha, \mathbf{w}_L, b_L, \mathbf{c}, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ G_1(\alpha, \mathbf{w}_1, b_1, \mathbf{c}, \mathbf{x}_N) & \cdots & G_L(\alpha, \mathbf{w}_L, b_L, \mathbf{c}, \mathbf{x}_N) \end{bmatrix} \text{ and}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

is the output data vector with N the number of data points.

Table 2. Output activation functions used in ELM.

#	Name	Activation Function G
0	Tribas	$G(x) = 1 - \|x\|$ if $-1 \geq x \geq 1$ otherwise 0
1	Identity	$G(x) = x$
2	ReLU	$G(x) = \max(0, x_i; i = 1, \cdots, D)$
3	Swish	$G(x) = \dfrac{x}{1 + exp(-x)}$
4	Inverse Tribas	$G(x) = \|x\|$ if $-1 \geq x \geq 1$ otherwise 0
5	HardLim	$G(x) = 1$ if $x \geq 0$ otherwise 0
6	SoftLim	$G(x) = x$ if $0 \geq x \geq 1$ else 0 if $x < 0$ otherwise 1
7	Gaussian	$G(x) = exp(-x^2)$
8	Multiquadric	$G(x) = \sqrt{x^2 + b^2}$
9	Inverse Multiquadric	$G(x) = \dfrac{1}{(x^2 + b^2)^{1/2}}$

The optimal solution is given by

$$\boldsymbol{\beta} = (\mathbf{H}^{\mathbf{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathbf{T}}\mathbf{y} = \mathbf{H}^{\dagger}\mathbf{y}$$

where $\mathbf{H}^{\dagger}$ is the pseudoinverse of $\mathbf{H}$. Figure 3 shows an example of a 4-8-1 ELM with four inputs, one hidden layer (8 neurons in each) and one output (petrographic class).

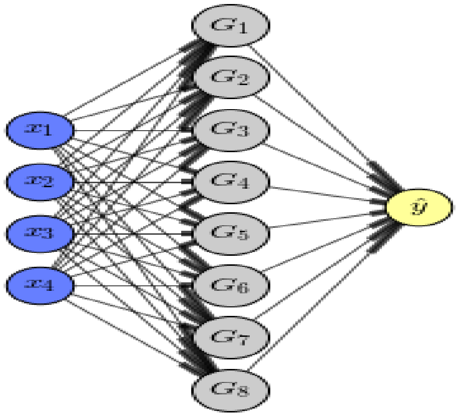


Figure 3. Connectivities for a 4-8-1 Extreme Learning Machine: four inputs, one hidden layer (8 neurons in each) and one output.

2.3 Evolutionary setting of ELM parameters using Differential Evolution

In general, classification models have important parameters which cannot be directly estimated from the data [33]. Such parameters are called hyperparameters, whose values are set before the learning process starts. The performance of a model can be significantly affected by the choice of the hyperparameter, but choosing the best set of values can become a complex task [34]. Often, these hyperparameters are defined empirically by testing different settings by hand or with an exhaustive search (Grid Search) [35]. Optimization techniques, such as nature-inspired algorithms [36] and automatic configuration procedures [37, 38] were also alternatives to search towards good parameter sets. It is important to note that Grid Search can be extremely computationally expensive depending on the size of the hyperparameter space and may take an impractical time to find the set of parameters that leads to the best performance of the model. An alternative is the use of population-based evolutionary optimization algorithms to find a set of hyperparameters values that produce an optimal or sub-optimal model which minimizes a predefined loss function on given test data.

Setting the parameters of a classifier is usually a difficult task. Often, these parameters are defined empirically, by testing different settings by hand. An alternative is the use of population-based evolutionary algorithms. Here we employ an Differential Evolution (DE) [24] to find the best set of ELM parameters, where each individual is a representation of an Extreme Learning Machine.

Differential evolution is known as one of the most efficient evolutionary algorithms (EAs). The basic strategy of DE can be described as follows [17]. Given a set of parameter vectors $\{\boldsymbol{\theta} i, J | i = 1, 2, \ldots, NP\}$ as a population at each generation J, we do iteratively:

1. Mutation: For each target vector $\boldsymbol{\theta}_{i,J+1}, i = 1, 2, \ldots, NP$, a mutant vector is generated according to

   $$\boldsymbol{\nu}_{i,J+1} = \boldsymbol{\theta}_{r_1,J} + F(\boldsymbol{\theta}_{r_2,J} - \boldsymbol{\theta}_{r_3,J})$$

   with random and mutually different indices $r_1, r_2, r_3 \in 1, 2, \ldots, NP$ and $F \in [0, 2]$. The constant factor F is used to control the amplification of the differential variation $(\boldsymbol{\theta}_{r_2,G} - \boldsymbol{\theta}_{r_3,G})$.

2. Crossover: In this step, the D-dimensional trial vector: $\boldsymbol{\mu}_{i,J+1} = (\boldsymbol{\mu}_{1i,J+1}, \boldsymbol{\mu}_{2i,J+1}, \ldots, \boldsymbol{\mu}_{Di,J+1})$ is formed so that

$$\boldsymbol{\mu}_{ji,J+1} = \begin{cases} \nu_{ji,J+1} & \text{if } randb(j) \leq CR \quad \text{or} \quad j = rnbr(i), \\ \theta_{ji,J} & \text{if } randb(j) > CR \quad \text{and} \quad j \neq rnbr(i). \end{cases} \tag{3}$$

   In Eq. (3), $randb(j)$ is the jth evaluation of a uniform random number generator with outcome in $[0, 1]$. CR is the crossover constant in $[0, 1]$ which is determined by user; $rnbr(i)$ is a random chosen integer index $\in [1, D]$ which ensures that $\boldsymbol{\mu}_{i,J+1}$ gets at least one parameter from $\boldsymbol{\nu}_{i,J+1}$.

3. Selection: If vector $\boldsymbol{\mu}_{i,J+1}$ is better than $\boldsymbol{\theta}_{i,J}$, then $\boldsymbol{\theta}_{i,J+1}$ is set to $\boldsymbol{\mu}_{i,J+1}$. Otherwise, the old value $\boldsymbol{\theta}_{i,J}$ is retained as $\boldsymbol{\theta}_{i,J+1}$.

Each candidate solution $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ encodes an ELM classifier. An individual represents the number of neurons in the hidden layers (θ_1), the activation function (θ_2) according to Table 2, and the parameter α (θ_3) as shown in Table 3. Considering the DE approach, the goal is to find the decision variables, corresponding to the ELM parameters and a subset of features, so that the network generates computed outputs that match the outputs of the training data.

2.4 Cross-validation

Cross-validation is a sampling statistical technique to evaluate the ability of generalization of a model from a dataset. Among the cross-validation techniques, k-fold [39] is one of the most used. k-fold uses a part of the data available to fit the model, and another different part to test it. The dataset is randomly divided into $k > 1$ subsets; from the k subsets, $k - 1$ are used for training and the remaining set is used for testing. This process is repeated k times, using a different test set in each iteration. Different from the Hold-out validation, where the data are divide only one time in train set and test set, the k-fold validation reduces the variance in the performance estimate for different data samples and, because of that, the performance estimate is less sensitive to the partitioning of the data. Figure 4 shows an example of 5-fold cross validation scheme.

Table 3. Encoding of DE for ELM hyperparameters setting. The column DV indicates the Decision Variable in the candidate solution.

DV	Description	Range
θ_1	Number of neurons in the hidden layer (see Fig. 3)	1–300
θ_2	Coding representing the activation function according to Table 2	0: Tribas, 1: Identity, 2: ReLU, 3: Swish, 4: Inverse Tribas, 5: HardLim, 6: SoftLim, 7: Gaussian, 8: Multiquadric, 9: Inverse Multiquadric
θ_3	Coding representing the α parameter	[0,1]

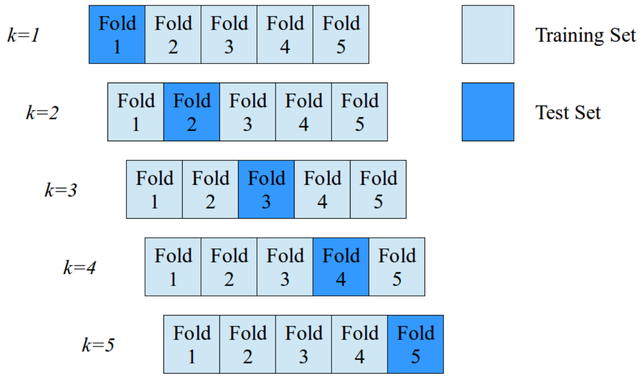


Figure 4. k-fold cross-validation method diagram ($k = 5$).

2.5 Performance Metrics

In order to evaluate the performance of the methods we used the following metrics: Accuracy, Recall, F1, and Kappa. The Accuracy, defined in Eq. (4), measures the percentage of correct classification by comparing the predicted classes with classified by the manual method, by direct counting.

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} I(f(x_i) = y_i) \tag{4}$$

where, $f(x_i)$ is the predicted class of a test samples and y_i is the true class of this sample. Consider that $I(true) = 1$ and $I(false) = 0$.

The Recall, given by

$$\text{Recall}(c_k) = \frac{TP_k}{TP_k + FN_k} \tag{5}$$

measures the percentage of actual positive samples that were classified as positive. In Eq. (5) where TP_k and FN_k are the number of true positives and the number of false negatives for class c_k, respectively.

The F1 score, also known as F-Measure, is written as

$$\text{F1}(c_k) = \frac{2TP_k}{2TP_k + FP_k + FN_k} \tag{6}$$

where TP_k is the number of positive samples that were correctly classified, FP_k is the number of negative samples classified as positive and FN_k is the number of positive samples classified as negative. F1 score reaches its best value at 1 and worst score at 0.

The Kappa Test is a measure of interobserver agreement and measures the degree agreement beyond what would be expected solely by chance. To describe whether there is an agreement between two or more evaluators, or between two classification methods, we used the Kappa measure which is based on consistent number of responses, i.e., the number of cases in which the result is even among evaluators. This agreement measure assumes a maximum value of 1; values close and even below 0 indicates no agreement. The KAPPA coefficient is calculated according to Eq. (7).

$$\text{Kappa} = \frac{P_o - P_E}{1 - P_E} \tag{7}$$

where

$$P_o = \frac{\text{no. agreement}}{\text{no. agreement} + \text{no. disagreement}} \tag{8}$$

and

$$P_E = \sum_{i=1}^{N} (p_{i1} \times p_{i2}) \tag{9}$$

where N is the number of categories, i is the index of categories, p_{i1} is the occurrence of proportion category i for evaluator 1, p_{i2} is the occurrence of proportion category i for evaluator 2. Table 4 shows the interpretation of the Kappa Statistics according to [40] to assess whether a agreement is reasonable.

Table 4. Kappa Statistics Strength Agreement.

Kappa Statistic	Strength Agreement
< 0.0	Poor
$0.00 - 0.20$	Slight
$0.21 - 0.40$	Fair
$0.41 - 0.60$	Moderate
$0.61 - 0.80$	Substantial
$0.81 - 1.00$	Almost Perfect

3. Computational Experiments

The computational experiments described here were conducted based in scikit-learn framework [35] and implementations adapted from [41], [42] and [43]. All codes and data

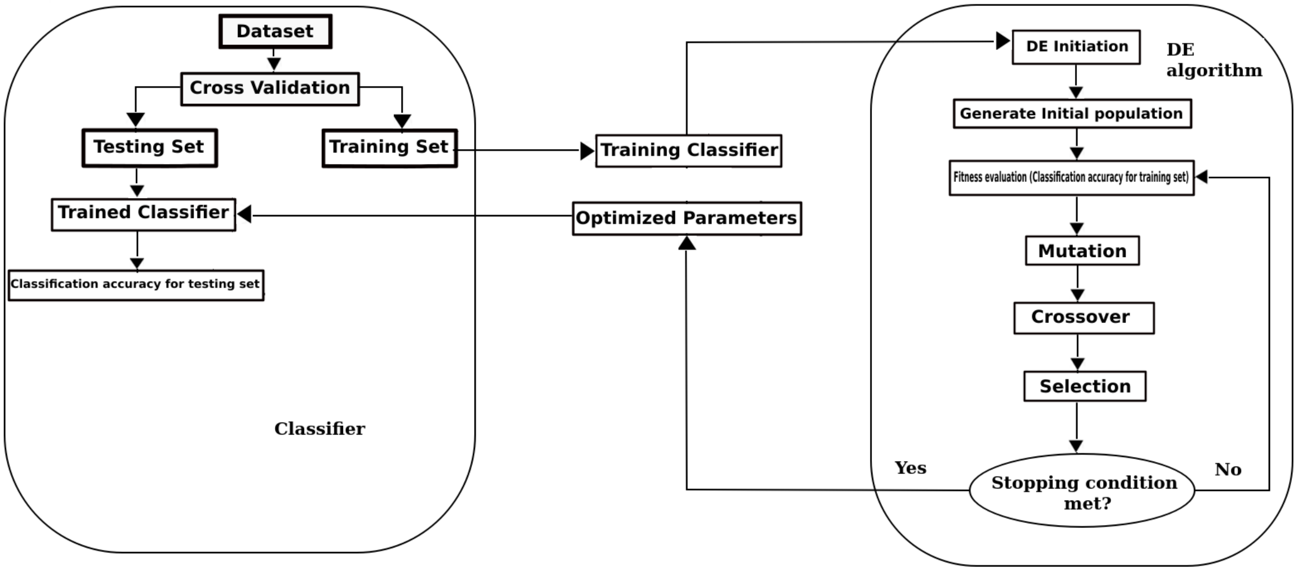


Figure 5. Scheme showing the procedure used for one iteration.

are made available by the authors upon request. Computer specifications used to execute ELM+DE are given as follows: CPU AMD Opteron Processor 6272 (64 cores of 2.1GHz and cache memory of 2MB), RAM of 250GB and operating system Linux Ubuntu 14.04.4 LTS. In order to obtain consistent and reliable results, 100 independent runs were performed using 5-fold cross-validation with shuffled data generated by different random seeds. Figure 5 shows the scheme representing the procedure used for one iteration. On average, each iteration takes approximately 10 minutes (in the ELM case).

The parameter settings used in the evolutionary process for ELM model selection are displayed in Table 5. CR was set to 0.7 and F was randomly chosen in the interval [0.5, 1]. A technique called Dither, proposed by [44], randomly selects the parameter F from the interval [0.5, 1.0] for each generation or for each difference vector that significantly improves convergence behavior, especially in noisy objective functions. A total of 30 individuals evolved under 50 generations for each run. The lower and upper bounds for the number of neurons in the hidden layer were set to 1 and 300, respectively. For each candidate solution, nine activation functions were available according to Table 2 and $\alpha \in [0,1]$. For the number of neurons in the hidden layer and the activation function, the nearest integer was used to define the parameters to be used in the classifier. The objective function (to be maximized) is the Accuracy given by Eq. (4).

Table 8 shows mean and standard deviation of the Accuracy, F1 and Recall for each class. It can be observed that as classes C4 and C7 produced the best averaged values for Accuracy, F1 and Recall. The results agreed with those obtained by [25]: Classes C1 and C2 were associated with the highest error rate (Limestone–Grainstone pair with respect to their lithology and texture according to Table 1) and also poor performance in class C5 justified by input data set limitations.

Table 5. DE parameter settings used in the optimization of ELM hyperparameters.

Parameter	Name	Value/Range
CR	Amplification factor	0.7
F	Mutation rate	[0.5, 1] (randomly chosen)
NP	Population size	30
J_{max}	Number of generations	50
θ_L	Lower bounds	$(\theta_1,\theta_2,\theta_3) = (1,0,0)$
θ_U	Upper bounds	$(\theta_1,\theta_2,\theta_3) = (300,9,1)$
Fitness function	Accuracy	Eq. (4)

Table 9 exhibits the percentage of the samples. It can be observed that the data used are unbalanced, which justifies the accuracy values produced by the method. A discussion on unbalanced lithologic datasets can be found in [45].

Figure 6 presents the confusion matrix of the seven petrographic classes, where rows represent a classification, since the columns represent a reference and a main diagonal representing the correctness of the classification. It is observed which petrographic classes are misclassified to other classes. Overall, classes C4, C7 and C6 have the highest prediction accuracy. In the class C1, 37% of the samples were classified as C2. Considering the class C2, 11% were predicted as C1 and 8% as C3. For C3 samples 16% as C2, 18%, 2% and 3% as C4, C6 and C7 respectively. For class C6 29% of samples were predicted to be C7 while for class C7 13% were considered as C6. This result may occur for C1 and C5 in cases where the lithology interpretations present possible errors. For C5, there is only one sample, which was mostly classified in C4 class due to their similarities in elastic, mineralogical and petrographic properties as can be seen in Table 1. Although ELM learned the training sample from C5, it did not record how to generalize new situations, which represents an overfitting problem.

In the barplot presented in Figure 7(a) one can observe

Table 6. Mean and standard deviation of the Accuracy, F1, Kappa, Recall and R^2 for 5-fold-cross-validation.

Reference	Classifier	Accuracy	F1	Kappa	Recall	R^2
This paper	ELM	0.696±0.044	0.696±0.044	0.630±0.054	0.696±0.044	0.878±0.064
	KNN	0.420±0.044	0.420±0.044	0.303±0.053	0.420±0.044	0.610±0.090
	LDA	0.412±0.058	0.412±0.058	0.292±0.070	0.412±0.058	0.404±0.142
Ref. [26]	–	–	–	–	–	0.8562(*)

(*) best result obtained in the training set

Table 7. Best model (according to accuracy) produced by the Differential Evolution (over 100 independent runs).

Parameters	Accuracy	F1	Kappa	Recall	R^2
G = ReLU, α = 0.04170, HL = 21	0.800	0.800	0.756	0.800	0.896

Table 8. Mean and standard deviation of the Accuracy, F1, and Recall, for each class, for 5-fold-cross-validation. A total of 100 independent runs were performed.

Class	# samples	Accuracy	F1	Recall
C1	5	0.565±0.290	0.494±0.240	0.473±0.256
C2	8	0.668±0.118	0.702±0.093	0.756±0.121
C3	6	0.638±0.173	0.571±0.134	0.532±0.139
C4	9	0.772±0.091	0.823±0.081	0.892±0.110
C5	1	0± 0	0±0	0±0
C6	4	0.725±0.192	0.682±0.174	0.676±0.219
C7	7	0.772±0.131	0.767±0.096	0.786±0.136

Table 9. Classes and Percentage of samples

Class	Percentage of samples (%)
1	12.50
2	20.00
3	15.00
4	22.50
5	2.50
6	10.00
7	17.50

the higher frequencies are for the ReLU and Swish activation functions. Figure 7(b) shows the optimal number of neurons according to each activation function. As shown in the figure, SoftLim and HardLim functions require a larger number of neurons (on average) in the hidden layer: 236 and 204, respectively. The larger number of neurons present by SoftLim and HardLim when compared with other activaction functions is explained due to the simplicity of its form. A high number of neurons in the network increases its complexity but, as shown in the figure, this complexity appears only 6 out 100 independent runs. For the function ReLU the indicated number of neurons is on average 26. Figure 7(c) shows the variation of α according to the activation functions (G): ReLU and Swich (selected 77 out 100 runs) produced the wide range of parameter α.

Figure 8 shows a comparison of the activation function and its derivative for ReLU and Swish. As depicted in the figure, the functions and their derivatives of Swish and ReLU behave very similarly. This similar behavior between Swish and ReLU also leads to similar results in this work, where they have been chosen almost the same number of times and the number of neurons is similar as well as the range of parameter α.

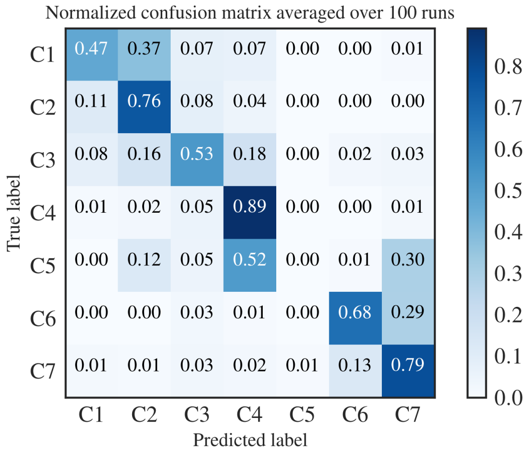
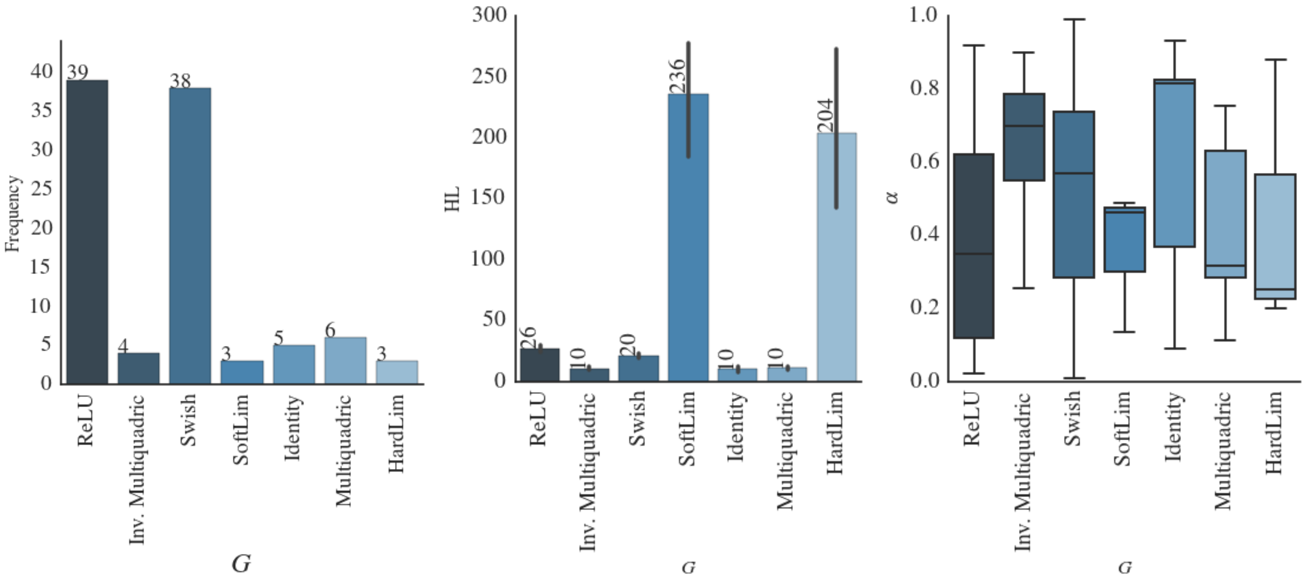


Figure 6. Confusion matrix plots on the test dataset. Normalized entries were averaged over 100 independent runs.

Figure 9 shows boxplots of the Accuracy, F1, Kappa and Recall according to activation functions. For all metrics, ReLU and Swish have similar behavior considering the boxplots. For mertic F1 the function ReLU obtained higher value, followed by the Swish function. In relation a Kappa metric ReLU presented great value and variety. The mean value 0.64 indicates substantially concordance with classification found in [25].

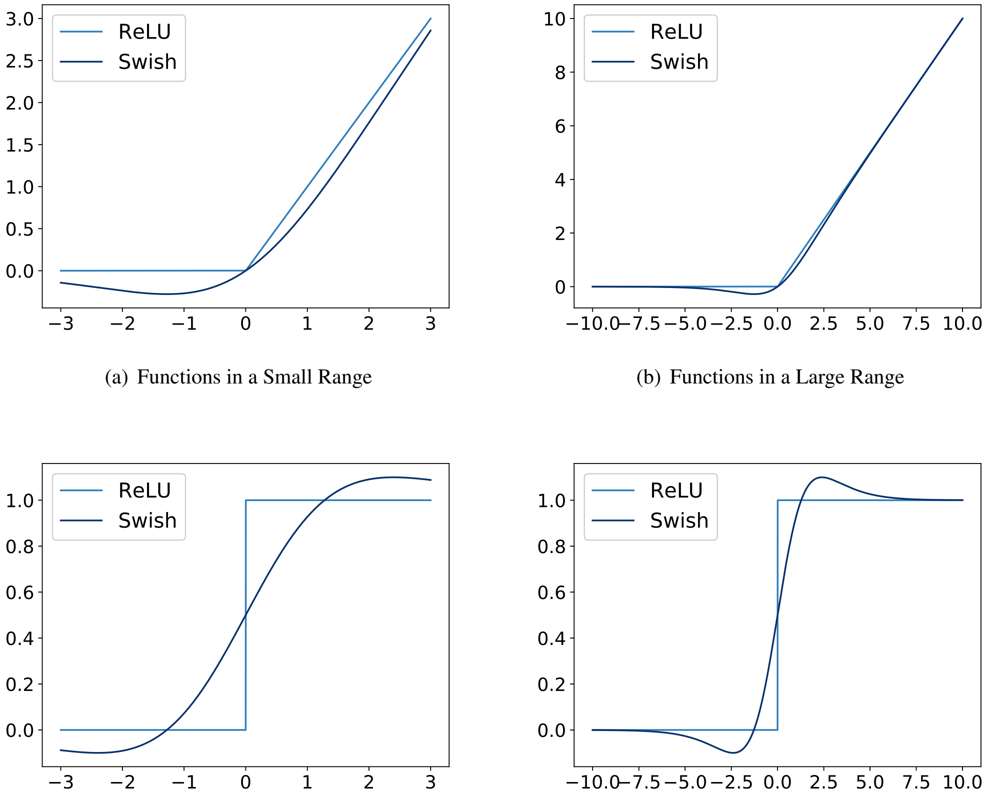
According to [22], the Swish activation function presents better performance when applied to Deep Learning techniques. The simplicity of Swish and its similarity to ReLU means that replacing the ReLU activation function in any network is just a simple one line code change. The properties of one-sided boundedness at zero, smoothness, and non-monotonicity presented by Swish may be the reason of your efficacy, but it is difficult to prove the cause of one activation function outperforms another. Based upon the results obtained in this paper, we can also observe that Swish shows competitive results compared to ReLU when applied on traditional neural models.

The construction of lithology databases often require a subjective and manual process to interpret and classify the



(a) *G* Frequency (b) Histogram HL x *G* (c) Boxplot α x *G*

Figure 7. Frequency of the activation functions (G) and the relation of the HL and α with G.



(a) Functions in a Small Range (b) Functions in a Large Range

(c) First Derivative of the Functions in a Small Range (d) First Derivative of the Functions in a Large Range

Figure 8. Comparison between ReLU and Swish: activaction functions and derivatives.

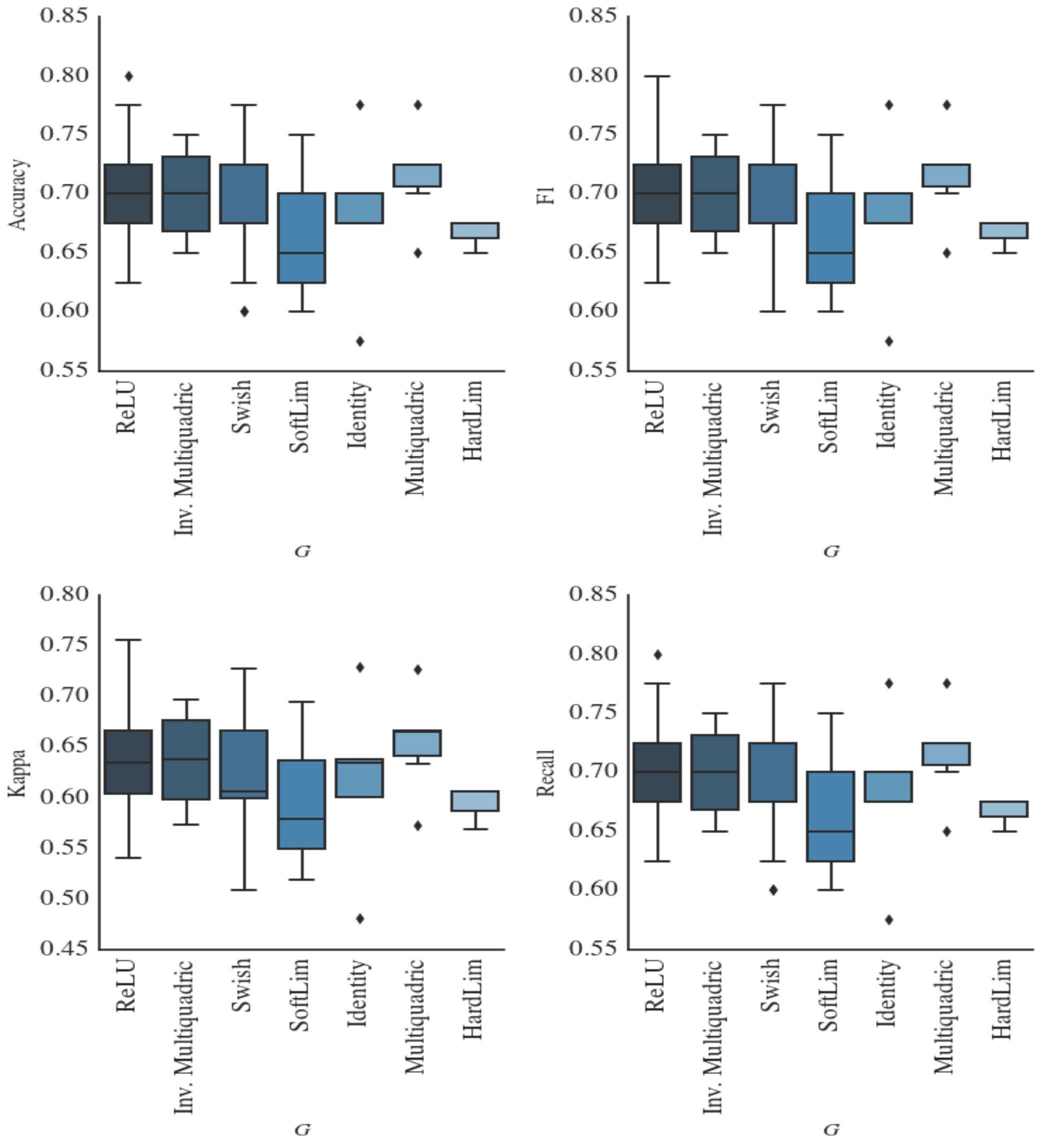


Figure 9. Boxplots showing performance measures accuracy, F1, Kappa and Recall as function of activation functions.

descriptive data. Although detailed, some inconsistencies can be present in lithology descriptions, such as [46]: the information recorded is dependent on the experience, skill, and prior knowledge of the person recording the logs; lithology databases often contain data collected over a period of many years and generated by different drilling equipment and with different aims and objectives. Another potential limitation is the wide variation in composition of some common materials [47]. As a result, in cases where lithology has been incorrectly mapped in the original source data, or is not recorded due to scale limitations, it can lead to poor results in the performance of the classifiers.

The proposed approach produced a model with good classification accuracy (the best model presented in the Table 7 obtained 80%), which can potentially help geologists/petrologists in determining the heterogeneity of a reservoir. In addition, specialists can apply the classification model to analyze a well logging database during geological exploration, which also provides an improvement in the efficiency of data analysis in the oil industry.

We leave as future work to use techniques to generate synthetic samples, since the number of samples in the database is small and may interfere with the performance of computational methods in order to investigate if the methodology will also perform effectively. Additionally, it is necessary to use more databases and also compare the results obtained by the ELM with other classification methods, such as Artificial Neural Networks, Support Vector Machines, Decision Trees, among others.

4. Conclusions

In this paper we analyzed the use of the Differential Evolution in the search for the optimal hyperparameters of the Extreme Learning Machines classifier is applied for lithology prediction from data that has elastic, mineralogical and petrographic properties. A newly proposed activation function called Swish was implemented and its performance was compared with other well established activation functions in the literature. It can be seen that the Swish function shown competitive results in comparison with the ReLU when applied to the problem in question. k-fold ($k = 5$) method is used as data partitioning criterion for testing and training data sets separation. We perform computational experiments and achieve better results than those achieved by [26]. It is concluded that the ELM is capable of assisting predicting lithology in reservoirs. The developed computational tool assists in petrographic data classification, helping the geologist to quickly identify the degree of heterogeneity of the reservoir, there by improving the process of reservoir characterization and the production development planning.

Acknowledgements

This work was supported by the Federal University of Juiz de Fora (UFJF), FAPEMIG (grant 01606/15) and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Author Contributions

- Camila Martins Saporetti: design the computational setup, implement the source code, wrote the paper, collect the data, perform the analysis.
- Grasiele Regina Duarte: perform the critical review of the computational framework, conduct the statistical analysis.
- Tales Lima Fonseca: implement the source code, perform the analysis of activation functions Relu and Swich.
- Leonardo Goliatt da Fonseca: design the computational setup, implement the source code, wrote the paper, perform the analysis.
- Egberto Pereira: perform the analysis, perform the critical review for petrographic results.

References

[1] VASINI, E. M. et al. Interpretation of production tests in geothermal wells with t2well-ewasg. *Geothermics*, v. 73, n. 1, p. 158–167, 2018.

[2] HORROCKS, T.; HOLDEN, E.-J.; WEDGE, D. Evaluation of automated lithology classification architectures using highly-sampled wireline logs for coal exploration. *Comput. Geosci-uk.*, v. 83, n. 1, p. 209 – 218, 2015.

[3] YANG, H. et al. Performance of the synergetic wavelet transform and modified k-means clustering in lithology classification using nuclear log. *J. Petrol. Sci. Eng.*, v. 144, p. 1 – 9, 2016.

[4] BORSARU, M. et al. Automated lithology prediction from pgnaa and other geophysical logs. *Appl. Radiat. Isotopes*, v. 64, n. 2, p. 272 – 282, 2006.

[5] POUR, A. B. et al. Lithological and alteration mineral mapping in poorly exposed lithologies using landsat-8 and aster satellite data: North-eastern graham land, antarctic peninsula. *Ore Geol. Rev.*, v. 1, n. 1, p. –, 2017.

[6] XIE, Y. et al. Evaluation of machine learning methods for formation lithology identification: A comparison of tuning processes and model performances. *J. Petrol. Sci. Eng.*, v. 139, n. 27, p. 182–193, 2018.

[7] GIFFORD, C. M.; AGAH, A. Collaborative multi-agent rock facies classification from wireline well log data. *Eng. Appl. Artif. Intel.*, v. 23, n. 7, p. 1158 – 1172, 2010.

[8] AKINYOKUN, O. et al. Well log interpretation model for the determination of lithology and fluid contents. *Pac. J. Sci. Technol.*, v. 10, n. 1, p. 507–517, 2009.

[9] DONG, S.; WANG, Z.; ZENG, L. Lithology identification using kernel fisher discriminant analysis with well logs. *J. Petrol. Sci. Eng.*, v. 143, n. 1, p. 95 – 102, 2016.

[10] KONATÉ, A. A. et al. Lithology and mineralogy recognition from geochemical logging tool data using multivariate statistical analysis. *Appl. Radiat. Isotopes*, v. 128, n. 1, p. 55 – 67, 2017.

[11] RAMKUMAR, M.; BERNER, Z.; STüBEN, D. Multivariate statistical discrimination of selected carbonate petrographic classifications: Implications on applicability of classification systems and predictability of petrographic types. *Chem. Erde*, v. 62, n. 2, p. 145–159, 2002.

[12] AL-ANAZI, A.; GATES, I. On the capability of support vector machines to classify lithology from well logs. *Nat. Resour. Res.*, v. 19, n. 2, p. 125–139, 2010.

[13] SEBTOSHEIKH, M. A.; SALEHI, A. Lithology prediction by support vector classifiers using inverted seismic attributes data and petrophysical logs as a new approach and investigation of training data set size effect on its performance in a heterogeneous carbonate reservoir. *J. Petrol. Sci. Eng.*, v. 134, n. 1, p. 143 – 149, 2015.

[14] CRACKNELL, M. J.; READING, A. M. Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information. *Comput. Geosci-uk.*, v. 63, n. 1, p. 22 – 33, 2014.

[15] HARRIS, J.; GRUNSKY, E. Predictive lithological mapping of canada's north using random forest classification applied to geophysical and geochemical data. *Comput. Geosci-uk.*, v. 80, n. 1, p. 9 – 25, 2015.

[16] PLASTINO, A. et al. Combining classification and regression for improving permeability estimations from 1h nmr relaxation data. *J. Appl. Geophys.*, v. 146, n. 1, p. 95 – 102, 2017.

[17] ZHU, Q.-Y. et al. Evolutionary extreme learning machine. *Pattern Recogn.*, v. 38, n. 10, p. 1759 – 1763, 2005.

[18] YANG, W.-A.; ZHOU, Q.; TSUI, K.-L. Differential evolution-based feature selection and parameter optimisation for extreme learning machine in tool wear estimation. *Int. J. Prod. Res.*, v. 54, n. 15, p. 4703–4721, 2016.

[19] BAZI, Y. et al. Differential evolution extreme learning machine for the classification of hyperspectral images. *IEEE Geosci. Remote Sens. Lett.*, v. 11, n. 6, p. 1066–1070, 2014.

[20] SHAO, Y.; CHEN, Q. Application genetic neural network in lithology recognition and prediction: Evidence from china. In: QI, L.; ZHOU, Q. (Ed.). *Intelligent Information Technology Application, 2008. Second International Symposium on*. Shanghai, China: IEEE, 2008. (IITA '08, v. 2).

[21] AN-NAN, J.; LU, J. Studying the lithology identification method from well logs based on de-svm. In: *Control and Decision Conference, 2009.* Studying the lithology identification method from well logs based on de-svm: IEEE, 2009. (CCDC, '09).

[22] ZOPH, P. R. anf B.; LE, Q. V. Searching for activation functions. *CoRR*, abs/1710.05941, n. 1, p. 1–13, 2017.

[23] HUANG, G.-B.; ZHU, Q.-Y.; SIEW, C.-K. Extreme learning machine: a new learning scheme of feedforward neural networks. In: *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*. Budapest, Hungary: IEEE, 2004. v. 2.

[24] STORN, R.; PRICE, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. glob. optim.*, v. 11, n. 4, p. 341–359, 1997.

[25] FOURNIER, F.; BORGOMANO, J. Critical porosity and elastic properties of microporous mixed carbonate-siliciclastic rocks. *Geophysics*, v. 74, n. 2, p. E93–E109, 2009.

[26] SILVA, A. A. et al. Artificial neural networks to support petrographic classification of carbonate-siliciclastic rocks using well logs and textural information. *J. Appl. Geophys.*, v. 117, n. 1, p. 118–125, 2015.

[27] BRIGAUD, B. et al. Acoustic properties of ancient shallow-marine carbonates: Effects of depositional environments and diagenetic processes (middle jurassic, paris basin, france). *J. Sediment. Res.*, v. 80, n. 9, p. 791–807, 2010.

[28] MATONTI, C. et al. Structural and petrophysical characterization of mixed conduit/seal fault zones in carbonates: Example from the castellas fault (se france). *J. Struct. Geol.*, v. 39, n. 1, p. 103 – 121, 2012.

[29] CEIA, M. A. de et al. Relationship between the consolidation parameter, porosity and aspect ratio in microporous carbonate rocks. *J. of Appl. Geoph.*, v. 122, p. 111 – 121, 2015.

[30] GUO, P.; CHENG, W.; WANG, Y. Hybrid evolutionary algorithm with extreme machine learning fitness function evaluation for two-stage capacitated facility location problems. *Expert Syst. Appl.*, v. 71, n. 1, p. 57 – 68, 2017.

[31] HUANG, G.-B. What are extreme learning machines? filling the gap between frank rosenblatt's dream and john von neumann's puzzle. *Cogn. Comput.*, v. 7, n. 3, p. 263–278, 2015.

[32] HUANG, G. et al. Trends in extreme learning machines: A review. *Neural Networks*, v. 61, n. Supplement C, p. 32–48, 2015.

[33] KUHN, M.; JOHNSON, K. *Applied predictive modeling*. 1. ed. Berlin, Germany: Springer, 2013. v. 26.

[34] MOOR, M. C. andBart D. Hyperparameter search in machine learning. *CoRR*, abs/1502.02127, n. 1, p. 1–5, 2015.

[35] PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, v. 12, n. 1, p. 2825–2830, 2011.

[36] BIAN, X.-Q. et al. Integrating support vector regression with genetic algorithm for co_2-oil minimum miscibility pressure (mmp) in pure and impure co_2 streams. *Fuel*, v. 182, n. 1, p. 550 – 557, 2016.

[37] BALAPRAKASH PRASANNAAND BIRATTARI, M. S. T. Improvement strategies for the f-race algorithm: Sampling design and iterative refinement. In: BARTZ-BEIELSTEIN THOMASAND BLESA AGUILERA, M. J. (Ed.). *Hybrid Metaheuristics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. (HM, '07).

[38] LÓPEZ-IBÁÑEZ, M. et al. The `irace` package: Iterated racing for automaticalgorithm configuration. *Oper. Res. Perspect.*, v. 3, n. 1, p. 43–58, 2016.

[39] HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*. 2. ed. Verlag, New York: Springer, 2009. v. 1. (Springer Series in Statistics, v. 1).

[40] GR, L.; GG, K. The measurement of observer agreement for categorical data. *Biometrics*, v. 33, n. 1, p. 159–174, 1977.

[41] FRIEDMAN, J. H. Multivariate adaptive regression splines. *ann. stat.*, v. 19, n. 1, p. 1–67, 1991.

[42] JONES, E.; OLIPHANT, T.; PETERSON, P. {*SciPy*}*: open source scientific tools for* {*Python*}. 2014.

[43] AKUSOK, A. et al. High-performance extreme learning machines: a complete toolbox for big data applications. *IEEE Access*, v. 3, n. 1, p. 1011–1025, 2015.

[44] PRICE RAINER M. STORN, J. A. L. K. *Differential evolution a practical approach to global optimization*. 1. ed. Berlin, Germany: Springer, 2005. v. 1. (Natural Computing Series, v. 1).

[45] SAPORETTI, C. M. et al. Machine learning approaches for petrographic classification of carbonate-siliciclastic rocks using well logs and textural information. *J. Appl. Geophys.*, v. 155, n. 1, p. 217 – 225, 2018.

[46] POLLOCK, D. W.; BARRON, O. V.; DONN, M. J. 3d exploratory analysis of descriptive lithology records using regular expressions. *Comput. Geosci-uk.*, v. 39, n. 1, p. 111 – 119, 2012.

[47] GRAY, J. M.; BISHOP, T. F.; WILFORD, J. R. Lithology and soil relationships for soil modelling and mapping. *CATENA*, v. 147, n. 1, p. 429 – 440, 2016.

1. Dataset

Table 10. La-Ciotat1 dataset, available from [25], in accordance with petrographic class predictions shown in Table 1. Grain size code: 1- very coarse; 2- coarse; 3- medium; 4- fine; 5- very fine.

Depth	Class	Porosity	Dry bulk density	VP	VS1	VS2	K1	K2	m1	m2	Grain size	Carbonate fraction	Calcite	Dolomite	Quartz	Orthoclase	Albite	Clays	Pyrite	α
(m)		(%)	(g/cm3)	(m/s)	(m/s)	(m/s)	(GPa)	(GPa)	(GPa)	(GPa)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)		
21.1	1	0.44	2.69	6375	3196	3209	72.58	72.28	27.44	27.66	3	69.86	99.8	0	0.2	0	0	0	0	0.7
22.1	1	0.66	2.68	6368	3217	3230	71.81	71.51	27.78	28	2	69.5	99.3	0	0.7	0	0	0	0	0.7
30.4	1	1.05	2.67	6341	3260	3244	69.62	69.99	28.41	28.14	4	69.72	99.6	0	0.5	0	0	0	0	0.7
36.45	1	2.65	2.62	5702	3043	3030	52.89	53.16	24.28	24.08	3	58.75	97.9	0	2.1	0	0	0	0	0.6
114.4	1	2.01	2.64	6240	3332	3230	63.74	66.09	29.32	27.55	3	66.15	97.9	0	2.1	0	0	0	0	0.7
2.7	2	0.39	2.69	6494	3498	3400	69.47	71.89	32.87	31.06	3	70	98.9	0	1.1	0	0	0	0	0.7
9.85	2	0.62	2.68	6344	3231	3258	70.54	69.91	27.97	28.44	3	64.27	98.9	0	1.1	0	0	0	0	0.65
15.7	2	0.18	2.69	6172	3242	3268	64.87	64.27	28.32	28.77	3	69.43	99.2	0	0.8	0	0	0	0	0.7
25.55	2	0.24	2.69	6207	3221	3206	66.44	66.78	27.91	27.65	2	67.9	100	0	0	0	0	0	0	0.7
32.35	2	1.73	2.65	6110	3261	3269	61.36	61.18	28.18	28.32	2	71.35	99.1	0	0.9	0	0	0	0	0.72
34.4	2	0.93	2.67	6404	3446	3332	67.27	70.03	31.73	29.66	3	68.72	98.2	0	1.8	0	0	0	0	0.7
42.4	2	4.44	2.58	5738	3091	3159	52.02	50.56	24.62	25.72	4	84.17	99	0	1	0	0	0	0	0.85
56.5	2	3.39	2.61	5846	3338	3296	50.34	51.31	29.03	28.31	4	34.38	98.2	0	1.8	0	0	0	0	0.35
0.5	3	0.82	2.7	6238	3202	3188	68.15	68.48	27.68	27.44	3	64.4	92.1	0	7.9	0	0	0	0	0.7
112.4	3	3.07	2.59	5543	3495	3286	37.41	42.3	31.65	27.98	4	23.6	92.1	0	7.9	0	0	0	0	0.4
116.4	3	6.71	2.5	5058	2909	2853	35.78	36.86	21.17	20.37	4	20.2	65.5	0	34	0	0	0.55	0	0.3
132.5	3	7.35	2.48	4599	2567	2611	30.67	29.91	16.34	16.91	5	49.64	89.6	1.2	8.3	0	0	0.9	0	0.55
64.4	3	1.68	2.65	5463	3105	3066	45.07	45.92	25.57	24.94	2	13.33	66.6	0	33.4	0	0	0	0	0.2
136.4	3	2.4	2.63	5640	3142	3102	49.07	49.94	25.98	25.32	3	17.77	88.8	0	11.2	0	0	0	0	0.2
66.4	4	15.2	2.24	3773	2368	2431	15.13	14.23	12.55	13.23	5	52.5	75.7	0	23.4	0	0	0.9	0	0.7
74.4	4	5.62	2.53	4623	2507	2547	32.89	32.2	15.91	16.42	4	36.5	75.7	0	23.4	0	0	0.9	0	0.5
76.45	4	8.61	2.46	4334	2471	2479	26.17	26.04	15.02	15.11	5	67.2	75.7	0	23.4	0	0	0.9	0	0.8
78.45	4	16.2	2.21	3730	2084	2248	17.93	15.84	9.59	11.16	5	46.9	75.7	0	23.4	0	0	0.9	0	0.7
110.4	4	6.7	2.49	4745	2585	2626	33.88	33.17	16.64	17.17	5	53.84	75.7	0	23.4	0	0	0.9	0	0.7
126.4	4	6.84	2.48	5274	2940	2899	40.34	41.13	21.4	20.81	4	79.83	77.17	0	19.37	0.89	0.59	0.89	1.19	0.9
140.4	4	8.61	2.43	3810	2302	2378	18.14	16.98	12.9	13.77	4	47.49	77.17	0	19.37	0.89	0.59	0.89	1.19	0.6
146.4	4	5.43	2.55	5097	2903	2892	37.62	37.83	21.5	21.34	3	45.64	94.9	0	5.1	0	0	0	0	0.48
84.4	4	5.14	2.54	5123	2942	2943	37.41	37.39	22.02	22.04	3	52.8	75.7	0	23.4	0	0	0.9	0	0.8
64.4	5	6.4	2.49	4644	2741	2810	28.78	27.5	18.72	19.67	5	43.96	56.4	0	35.1	1.5	2.5	4.5	0	0.75
96.4	6	5.03	2.53	4730	2938	2883	27.46	28.54	21.82	21.01	4	23.58	36.6	0	63.4	0	0	0	0	0.6
6.3	6	23.9	2.02	3496	2157	2139	12.15	12.35	9.39	9.23	2	38.03	12.4	0	87.6	0	0	0	0	0.95
20.45	6	21.2	2.1	3535	2204	2140	12.62	13.39	10.18	9.6	1	29.24	12.4	0	87.6	0	0	0	0	0.95
34.4	6	7.39	2.48	4544	2829	2815	24.73	25	19.84	19.65	2	28.98	12.4	0	87.6	0	0	0	0	0.6
80.4	7	20.8	2.06	3307	2032	2060	11.21	10.9	8.53	8.76	5	46	11.5	0	85.5	0	0	3	0	0.92
92.4	7	11.7	2.33	4036	2585	2523	17.22	18.2	15.59	14.85	4	20.58	11.5	0	85.5	0	0	3	0	0.95
118.4	7	7.39	2.44	4662	2705	2755	29.21	28.32	17.84	18.51	4	28.91	40	0	55	0	0	3.5	1.5	0.65
124.4	7	16.6	2.2	4162	2666	2582	17.27	18.57	15.65	14.68	5	34.55	29	0	61	1.7	2	3.9	2.5	0.85
134.4	7	20.2	2.1	3620	1980	1967	16.56	16.7	8.24	8.13	4	51.6	23.9	0	75.6	0	0	0.5	0	0.6
136.4	7	14.5	2.26	3771	2344	2306	15.57	16.1	12.41	12.01	5	31.15	23.9	0	75.6	0	0	0.5	0	0.9
78.4	7	5.22	2.54	5026	3046	3068	32.78	32.32	23.59	23.94	2	17.69	41.7	0	58.3	0	0	0	0	0.4