Earth Science Informatics (2021) 14:133–140
https://doi.org/10.1007/s12145-020-00533-x

RESEARCH ARTICLE

Neural network boosted with differential evolution for lithology identification based on well logs information

Camila Martins Saporetti[1] · **Leonardo Goliatt**[2] · **Egberto Pereira**[3]

Received: 26 March 2020 / Accepted: 25 September 2020 / Published online: 10 October 2020

Abstract

Lithology identification of geological beds in the subsurface is fundamental in reservoir characterization. Recently, automated log analysis has an increasing demand in reservoir research and the oil industry. In this context, Machine Learning (ML) techniques arise as a surrogate model to provide lithology identification in a fast way. However, to achieve suitable performance, ML techniques require the adjustment of some parameters, and that can become a hard task, depending on the difficulty of the problem to be solved. This paper presents an Artificial Neural Network (ANN), assisted by an adaptive Differential Evolution (DE) algorithm to classify petrophysical data in the Southern Provence Basin. The main contribution is searching for a competent ANN configuration, including architecture, activation functions, regularization, and training algorithms. The proposed approach outperformed four classifiers and two results previously published. The computational methodology proposed here is able to assist in the classification of petrophysical data, helping to improve the procedure of reservoir characterization and the idealization of the development of production.

Keywords Lithology identification · Neural networks · Evolutionary algorithms · Automated machine learning

Introduction

Lithology identification of geological beds in the subsurface is fundamental in reservoir characterization, as one cannot predict the fluid content of any geological bed without knowing the lithology that the fluid is associated with. To make petrophysical properties estimations, such as porosity, clay volume, water saturation, permeability, the various lithologies of the reservoir interval must be identified, and their properties understood. Accurate determination and understanding of lithology are fundamental to other petrophysical analyses, critical for practical exploration, and production of hydrocarbon. The economic potential of an oil reservoir depends on the quality and description of lithology (Abbey et al. 2018).

The determination of lithology can be carried out by direct or indirect methods. Direct examination of underground cores sample from the intervals of interest is costly, partially reliable and biased different geologists can provide dissimilar analysis. Indirect methods use well log data to measure the physical aspects of geological formations, providing most of the data accessible to a geologist. Considering their influence on decisions, they are also fundamental tools for the mapping and determination of lithologies. However, indirect methods do not have near-performance, comparable to direct methods (Thomas et al. 1995; Burke et al. 1969). Manual analysis of well log lithologies is a labor-intensive procedure that involves a considerable amount of time spent by a competent specialist, even when assisted by graphical methods as cross-plotting (Vasini et al. 2018). The problem becomes particularly challenging as the number of synchronous logs to be evaluated increases. Therefore, would be useful to automate the reservoir characterization process. Thus,

Communicated by: H. Babaie

✉ Leonardo Goliatt
leonardo.goliatt@ufjf.edu.br

Camila Martins Saporetti
camilasaporetti@ice.ufjf.br

Egberto Pereira
egberto@uerj.br

[1] UEMG, Divinopolis, Brazil

[2] UFJF, Juiz de Fora, Brazil

[3] UERJ, Rio de Janeiro, Brazil

computational technologies are employed in lithology to assist geologists and increase accuracy (Yang et al. 2016; Pour et al. 2017; Xie et al. 2018; Tian et al. 2020). As a result, geologists can develop better quantitative analytical models of distinct rock properties, which can also improve overall assessment.

The use of Deep Learning has grown in recent years in a variety of tasks such as image recognition, speech recognition, and machine translation. Novel neural architectures are fundamental to this progress. Generally, the architectures applied are developed manually, resulting in a time-consuming and error-prone process. In view of this demand, there is a growing use of automated neural architecture search methods (Elsken et al. 2019). Automated Machine Learning (AutoML) is the process of taking training data with a defined label target, and iterating through hibridizations of techniques and feature selections to automatically select the best model or set of models for your data based on the training scores. In the context of lithology identification, AutoML approaches are a kind of computational techniques that can potentially make the procedure of reservoir and rock formation identification more efficient, agile and automatic, searching for the proper settings to perform the data identification as best as possible (Min et al. 2020; Zeng et al. 2020). Neural Architecture Search (NAS) (Elsken et al. 2019), which is the process of architecture engineering automation, is being considered as the logical step in automating machine learning. NAS can be seen as a subfield of AutoML and has a significant overlap with hyperparameters' optimization.

The aim of this study is to apply an AutoML approach using Differential Evolution (DE) (Storn and Price 1997) for searching the optimal model able to classify data from the South Provence Basin, taken from Fournier and Borgomano (2009), using the paper by Saporetti et al. (2018) as a baseline reference. The neural network presented in Saporetti et al. (2018) had only one layer. This paper presents an improvement in the model's structure, where the number of layers and neurons per layer are chosen by a Differential Evolution algorithm, configuring a Neural Architecture Search. Five classifiers were evaluated in the aforementioned database to verify the best-fit approach for the petrographic classification.

The main contribution of this work is the development of a neuro-evolutionary search framework to lithology classification using well log information. The remainder of this paper is organized as follows. "Material and methods" presents the experimental data used in this study, the research methodology and evaluation methods. "Computational experiments" addresses the models' implementation, results, and discussions. Concluding remarks are given in the final section.

Material and methods

Dataset

The well under consideration is located within the South Provence Basin in the vicinity of Cassis and La Ciotat, called La Ciotat-1 (Fig. 1). This dataset, obtained from Fournier and Borgomano (2009) and used in several studies (Silva et al. 2015; Brigaud et al. 2010; Matonti et al. 2012; De Ceia et al. 2015), combines (1) ultrasonic measurements of P- and S-wave velocities at various effective pressures, (2) density and porosity measurements, (3) quantitative mineralogic analyses using X-ray diffraction (XRD), (4) detailed petrographic studies of thin sections, and (5) critical porosity and elastic properties of microporous mixed carbonate-siliciclastic rocks. The data were separated into seven petrographic classes according to Fournier and Borgomano (2009) as shown in Table 1. The dataset has 40 samples, and 19 features were analyzed (input variables).

Neuro-evolutionary search strategy

The choice of the appropriate ANN architecture and its parameters can lead to significant gains in the solution of the problem. However, it is hard to create a hand-designed architecture that improves the performance as best as possible. The automatic construction of neural network architectures has been growing the interest of several researchers (Kitano 1990; Fahlman and Lebiere 1990; Liu et al. 2017), using different types of techniques

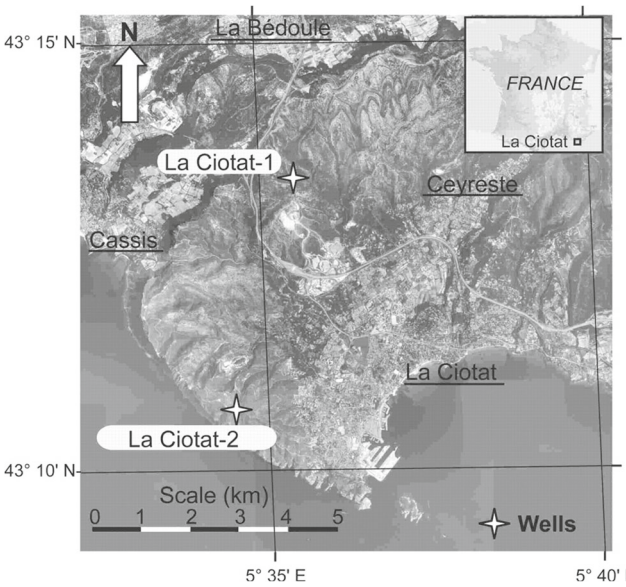

Fig. 1 Location of La Ciotat-1 and La Ciotat-2 wells (extracted from Fournier and Borgomano (2009))

Table 1 Petrographic classes and their descriptions according to Fournier and Borgomano (2009)

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 $> 50\%$)

such as Reinforcement or Hierarchical Learning, Bayesian optimization, and Evolutionary Algorithms (Xie and Yuille 2017; Zoph et al. 2018; Jaafra et al. 2019).

An alternative to building the most appropriate ANN is using an Evolutionary Algorithm (EA). In this context, the ANN search space comprises candidate solutions encoded in vectors $\boldsymbol{\theta_i} = \{\theta_{i,1}, \theta_{i,2}, \theta_{i,3}, \theta_{i,4}, \theta_{i,5}, \theta_{i,6}, \theta_{i,7}, \theta_{i,8}, \theta_{i,9}\}$ that control the neural network architecture, training algorithm, L_2 regularization, and activation functions. We highlight the proposed coding is dependent on the implementation of the neural network. The implementation employed here supports regularization, represented by parameter $\theta_{i,3}$, that can not be used for other neural networks. Also, increasing the number of layers (parameter $\theta_{i,4}$) will lead to an increase in the dimensionality of the vector. The ANN encoding is described in Tables 2 and 3 while Fig. 2 displays some examples of candidate solutions using this encoding. Using this encoding, it is possible to perform a search on the parameter space, including the activation function, training algorithm, regularization coefficient of the loss function used to train the network, the number of layers, and the number of neurons per layer.

Table 2 Encoding of ANN solutions

DV	Description	Range/Set
$\theta_{i,1}$	Activation function	0: I(x), 1: L(x), 2: T(x), 3: R(x)
$\theta_{i,2}$	ANN training algorithm (Solver)	0: L-BFGS, 1: SGD, and 2: Adam
$\theta_{i,3}$	L_2 Regularization coeff.	[0, 0.1]
$\theta_{i,4}$	Number of layers	[1, 5]
$\theta_{i,5:9}$	No. neurons, 1st-5th layers	[1, 50]

The column DV indicates the Decision Variable in the candidate solution

Table 3 Output activation functions used in ANN

ID	Name	Activation Function (φ)
0	Identity	$\mathrm{I}(x) = x$
1	Logistic	$\mathrm{L}(x) = 1/(1 + e^{-x})$
2	Tanh	$\mathrm{T}(x) = \tanh(x)$
3	ReLU	$\mathrm{R}(x) = \max(0, x_i; i = 1, \cdots, n)$

Differential Evolution (DE) (Storn and Price 1997) is one of the most efficient EAs. In this paper, a self-adaptive DE based on Brest et al. (2006), referred to as jDE, uses a self-adaptive mechanism during the evolutionary process. The goal of DE is to minimize an objective function $f(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the parameter vector. DE works iteratively with a population of NP ordered sequences $\{(\boldsymbol{\theta}_i, F_i, CR_i, V_i) \mid i = 1, 2, \ldots, NP\}$. The vector $\boldsymbol{\theta}_i$ controls the neural network parameters, F_i is called scaling factor, CR_i is the crossover rate, and V_i is the mutation variant. The role of each element in the sequence is explained below. Each individual in the population is generated randomly according to

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_L + \boldsymbol{r}(\boldsymbol{\theta}_U - \boldsymbol{\theta}_L) \quad (1)$$

$$F_i = 0.1 + 0.9 rand_i \quad (2)$$

$$CR_i = rand_i \quad (3)$$

$$V_i = V_{k_{rand}},\ k_{rand} \in \{1, 2, 3, 4, 5\} \quad (4)$$

where $\boldsymbol{r} = (r_1, r_2, \ldots, r_D)^T$ is a D-dimensional random vector generated uniformly in [0,1], $\boldsymbol{\theta}_L$ and $\boldsymbol{\theta}_U$ are respectively the lower and upper bounds of the parameters, $rand_i \in [0, 1]$ is a uniform random number, and $k_{rand} \in [1, 5]$ is a random integer. At each iteration G ($G = 1, 2, ..., G_{max}$) the following operations are performed on these vectors:

1. Mutation operation: For each $\boldsymbol{\theta}_i$, a mutant vector $\boldsymbol{v}_i$ is generated according to one of the following V_i variants:

$$V_1 : \boldsymbol{v}_i = \boldsymbol{\theta}_{r_1} + F_i(\boldsymbol{\theta}_{r_2} - \boldsymbol{\theta}_{r_3}) \quad (5)$$

$$V_2 : \boldsymbol{v}_i = \boldsymbol{\theta}_i + F_i(\boldsymbol{\theta}_{best} - \boldsymbol{\theta}_i) + F_i(\boldsymbol{\theta}_{r_1} - \boldsymbol{\theta}_{r_2}) \quad (6)$$

$$V_3 : \boldsymbol{v}_i = \boldsymbol{\theta}_{r_1} + F_i(\boldsymbol{\theta}_{r_2} - \boldsymbol{\theta}_{r_3}) + F_i(\boldsymbol{\theta}_{r_4} - \boldsymbol{\theta}_{r_5}) \quad (7)$$

$$V_4 : \boldsymbol{v}_i = \boldsymbol{\theta}_{r_1} + F_i(\boldsymbol{\theta}_{r_2} - \boldsymbol{\theta}_{r_3} + \boldsymbol{\theta}_{r_4} - \boldsymbol{\theta}_{r_5} + \boldsymbol{\theta}_{r_6} - \boldsymbol{\theta}_{r_7}) \quad (8)$$

$$V_5 : \boldsymbol{v}_i = \boldsymbol{\theta}_{r_1} + F_i(\boldsymbol{\theta}_{r_2} - \boldsymbol{\theta}_i) + F_i(\boldsymbol{\theta}_{r_3} - \boldsymbol{\theta}_{r_4}) \quad (9)$$

where $V_i, \ldots, V_5$ are the mutation variants, the indices $r_1, \ldots, r_7$ are mutually exclusive integers randomly generated within the range $[1, NP]$, which are also different from the index i. These indices are randomly generated once for each mutant vector. F_i is the scaling factor parameter that controls the magnitude of the difference vectors. $\boldsymbol{\theta}_{best}$ is the best individual vector with the best fit value in the population at generation G.

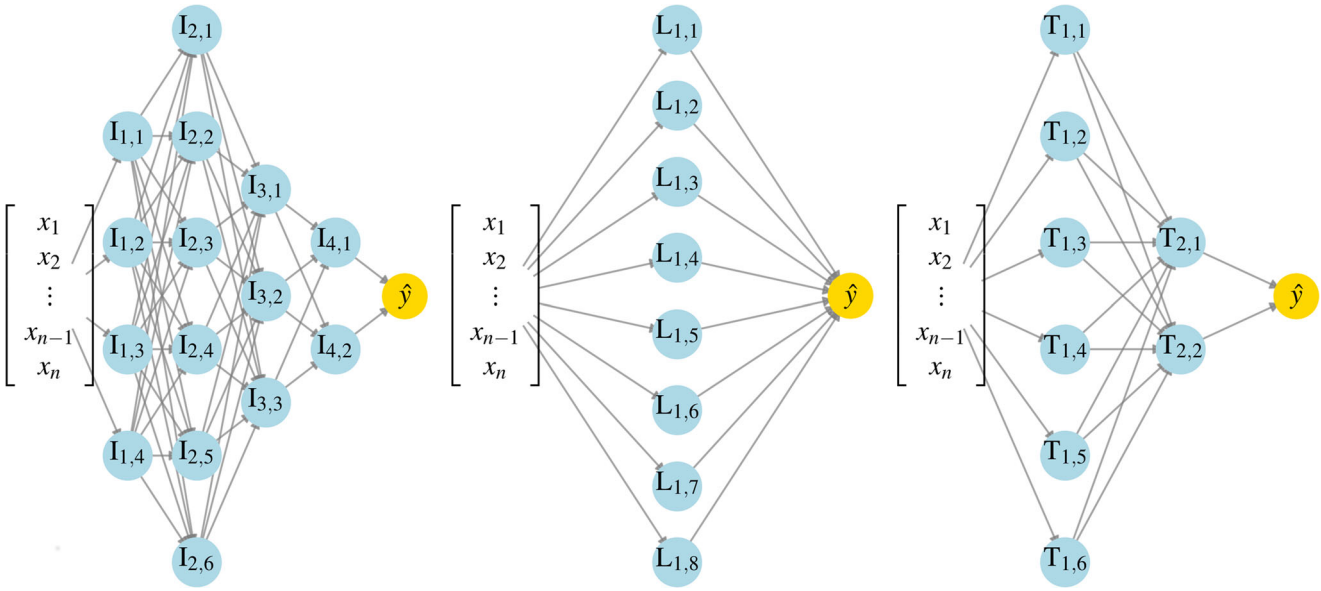

Fig. 2 Examples of ANN search space: candidate solutions implementing SGD training algorithm ($\boldsymbol{\theta}_{i,2} = 1$) and using a regularization coefficient equals to 0.05 as described in Table 2. Left: candidate solution $\boldsymbol{\theta}$ =[0,1,0.05,4,4,6,3,2,–], which represents a neural network with Identity activation function and 4 hidden layers with 4, 6, 3 and 2 neurons respectively. Center: candidate solution $\boldsymbol{\theta}$ =[3,1,0.05,1,8,–,–,–,–], ReLU activation function with eigth neurons on the first layer. Right: candidate solution $\boldsymbol{\theta}$ =[2,1,0.05,2,6,2,–,–,–], hyperbolic tangent activation function, six neurons in the first hidden layer and two neurons in the second one

2. Crossover operation: In this step, a trial vector $\boldsymbol{\mu}_i$ is generated using a stochastic operation given by

$$\mu_{i,j} = \begin{cases} v_{i,j}, & \text{if } rand_{ij} \leq CR_i \text{ or } j = j_{rand} \\ \theta_{i,j}, & \text{otherwise} \end{cases} \tag{10}$$

where $j = 1, 2, ..., D$, $v_{i,j}$ is the value of j-th variable of vector $\boldsymbol{v}_i$ produced by mutation operation $j_{rand} \in \{1, 2, ..., D\}$ is a random integer value produced for each solution and CR_i is the crossover rate.

3. Control parameters update operation: For each individual $\boldsymbol{\theta}_i$ the new control parameters $F_{i,G}^{trial}$, $CR_{i,G}^{trial}$ and $V_{i,G}^{trial}$ are calculated as

$$F_{i,G}^{trial} = \begin{cases} 0.1 + rand_1 \cdot 0.9, & \text{if } rand_2 \leq \tau_1 \\ F_{i,G}, & \text{otherwise} \end{cases} \tag{11}$$

$$CR_{i,G}^{trial} = \begin{cases} rand_3, & \text{if } rand_4 \leq \tau_2 \\ CR_{i,G}, & \text{otherwise} \end{cases} \tag{12}$$

$$V_{i,G}^{trial} = \begin{cases} V_{k_{rand}}, \; k_{rand} \in [1, 5] & \text{if } rand_5 \leq \tau_3 \\ V_{i,G}, & \text{otherwise} \end{cases} \tag{13}$$

where $rand_j, j \in \{1, 2, 3, 4, 5\}$ are uniform random values $\in [0, 1]$, $k_{rand} \in \{1, 2, 3, 4, 5\}$ is a random integer value, $\tau_1 = \tau_2 = \tau_3 = 0.1$ represent the probabilities to adjust the parameters F_i, CR_i, and V_i respectively.

4. Selection operation: If offspring vector $\boldsymbol{\mu}_{i,G}$ is better than $\boldsymbol{\theta}_{i,G}$, then $\boldsymbol{\theta}_{i,G+1}$ and the control parameters $F_{i,G+1}$, $CR_{i,G+1}$, and $V_{i,G+1}$ will be replaced. Otherwise, the old values will be maintained and copied to the next generation as follows

$$\begin{aligned} &\left(\boldsymbol{\theta}_{i,G+1}, F_{i,G+1}, CR_{i,G+1}, V_{i,G+1}\right) \\ &= \begin{cases} (\boldsymbol{\mu}_{i,G}, F_{i,G}^{trial}, CR_{i,G}^{trial}, V_{i,G}^{trial}), & \text{if } f(\boldsymbol{\mu_{i,G}}) \leq f(\boldsymbol{\theta_{i,G}}) \\ (\boldsymbol{\theta}_{i,G}, F_{i,G}, CR_{i,G}, V_{i,G}), & \text{otherwise} \end{cases} \end{aligned} \tag{14}$$

Computational experiments

The computational experiments described here were conducted based on scikit-learn framework (Pedregosa et al. 2011). The parameter settings used in the DE-based neuroevolutionary process is displayed in Table 4. The lower and upper limits of the ANN encoding, $\boldsymbol{\theta}_L$ and $\boldsymbol{\theta}_U$ respectively, also appear in Table 4. A 5-fold cross-validation was employed to evaluate each candidate solution: the dataset is randomly divided into five subsets; from the five subsets, 4 out of them were used in the training step, and the remaining set was used in the testing step. The objective function (to be maximized) is F1 given in Table 5. A total of 50 individuals were evolved under 100 generations for each independent run, in a total of 50 runs.

Table 4 DE parameter settings used in the optimization of classifiers hyperparameters

Param.	Name	Value/Range
NP	Population size	50
G_{max}	No. generations	100
$\boldsymbol{\theta}_L$	Lower bounds	$(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8, \theta_9) = (0, 0, 0.0, 1, 1, 1, 1, 1, 1)$
$\boldsymbol{\theta}_U$	Upper bounds	$(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8, \theta_9) = (3, 2, 0.1, 5, 50, 50, 50, 50, 50)$
$f(\boldsymbol{\theta})$	Objective function	F1 from Table 5

OF is the objective function. Boldface indicate that the variable is a vector

Table 6 presents the mean and standard deviation of the Accuracy, F1, Precision, and Recall by ANN, Extreme Learning Machine (ELM), Decision Tree (DT), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). It can be observed that the ANN presented superior performance when compared to similar methods commonly used in the literature.

The method reached F1 = 0.720, achieving better performance than the methodology previously proposed in Saporetti et al. (2018) (F1 = 0.615) and in Saporetti et al. (2018) (F1 = 0.696). The same evolutionary search procedure was performed for all methods. In Saporetti et al. (2018) the parameters were chosen through the exhaustive search (Grid-Search), while in Saporetti et al. (2018) a hybrid ELM was used to explore the parameters through evolutionary algorithm. As cited before, in this paper the neural network can have up to 5 layers, improving the model presented in Saporetti et al. (2018), which used a single layer. The approach proposed in Saporetti et al. (2019) presents higher computational complexity and needs more time for training. The method in this paper is able to produce a similar performance using a less complex model.

The grid search can become a very time-consuming step when the number of parameters increases or when we have a large set of user-defined options of each parameter. In order to survey the effectiveness of the proposed methodology, consider for example a five discrete options for every single one of the nine parameters in Table 2. In an exhaustive search, the total of possible candidate settings generated is $5^9 = 1953125$, while the use of DE algorithm with a population size of 50 individuals evolving along 100 generations produce a total of $50 \times 100 = 5000$ candidate settings, representing 0.256% of the cost of the exhaustive search procedure . In such a case, the evolutionary approach for model selection arises as an interesting alternative to save computing time while producing optimal or near-optimal solutions for the model selection procedure.

Figure 3 presents the confusion matrix of the seven petrographic classes, where rows represent a classification while the columns represent the reference class. The main diagonal shows the correctness of the classification. In this plot it can be observed which petrographic classes are misclassified to other classes. Overall, classes C5, C4, and C2 have the highest prediction accuracy.

For the class C1, 47% of the samples were classified as C2. In contrast, for class C2, 14% were predicted as C1, 9% as C3, and 2% as C4. For C3 samples, 19% as C4, 20% as C2, 1%, and 2% as C1 and C7, respectively. For

Table 5 Performance metrics: $f(x_k)$ is the predicted class of a test sample, c_k is the true class of this sample, $I(true) = 1$ and $I(false) = 0$

Name	Expression	Description
Accuracy	$\frac{1}{N}\sum_{k=1}^{N} I(f(x_k) = c_k)$	Percentage of corrected predicted classes when compared with those classified by the manual method.
Precision	$\frac{TP_k}{PP_k}$	Rate of Predicted Positive (PP) cases that are correctly True Positives (TP).
Recall	$\frac{TP_k}{TP_k + FN_k}$	Percentage of actual positive samples that were classified as positive.
F1	$\frac{2TP_k}{2TP_k + FP_k + FN_k}$	Weighted average of the Accuracy and Recall; F1 reaches its best value at 1 and worst score at 0.

TP_k is the number of samples that were correctly classified, FP_k is the total number of misclassified samples and FN_k is the number of misclassified samples for class c_k

Table 6 Mean and standard deviation of the Accuracy, F1, Precision and Recall for 5-fold-cross-validation

Est.	Accuracy	F1	Precision	Recall
ANN	0.736 (0.038)	0.720 (0.041)	0.728 (0.038)	0.736 (0.038)
ELM	0.693 (0.041)	0.677 (0.047)	0.682 (0.056)	0.693 (0.041)
DT	0.684 (0.054)	0.673 (0.056)	0.685 (0.062)	0.684 (0.054)
SVM	0.624 (0.043)	0.590 (0.041)	0.591 (0.057)	0.624 (0.043)
KNN	0.616 (0.047)	0.587 (0.044)	0.599 (0.051)	0.616 (0.047)
Saporetti et al. (2018)	0.628 (0.065)	0.615 (0.064)	0.625 (0.063)	0.628 (0.065)
Saporetti et al. (2018)	0.696 (0.044)	0.696 (0.044)	0.630 (0.054)	0.696 (0.044)

The first column shows the estimator

class C6 28% of samples were predicted to be C7 while for class C7 12% were considered as C6. 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 ANN learned the training sample from C5, it did not record how to generalize new situations, representing an overfitting problem. The classes C6 and C7 are sandstones, and the method was able to separate from the rest of the carbonate classes, except for C5 where 22% was classified as C7 due to the percentage of orthoclase, albite, and clays.

In the barplot presented in Fig. 4 one can observe the highest frequency is for the Relu function and shows the frequencies to each layer sizes. Note that the higher frequencies for two or three layers. Solver frequencies also presented in this Figure, where one can verify that Adam and L-BFGS are competitive.

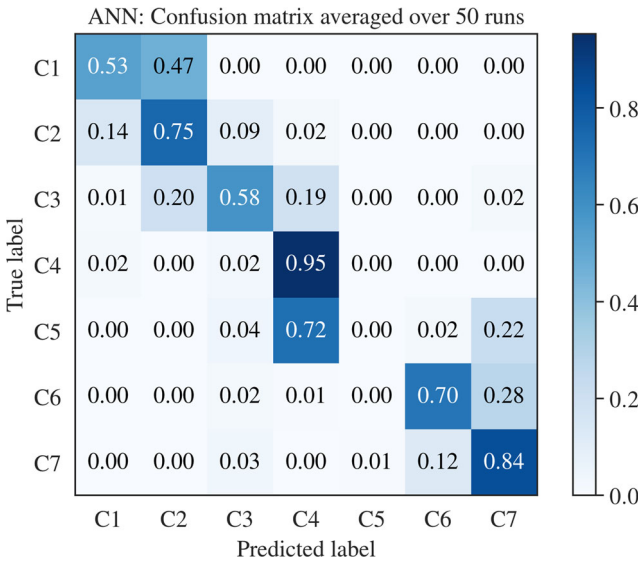

Fig. 3 Confusion matrix for petrographic classes

Figure 5 shows a comparison between the lithologic interpretation of well, according to Fournier and Borgomano (2009) and the classes found by classifiers. It can be noticed the similarity between the classes found and those that were proposed by Fournier and Borgomano (2009).

The approach proposed here is flexible when considering the evolutionary algorithm employed to perform the search procedure since other EAs (Wu et al. 2018; Yang et al. 2017) can be easily integrated into the computational framework. Some classes had performance compromised by data unbalance, such as class C5, which has only one sample. An alternative to enhance interpretability is to use feature selection approaches, aiming at removing some irrelevant and/or redundant features (Tuv et al. 2009; Tsai et al. 2013). Alternatively, polynomial combinations of the attributes (Geron 2017) can be used to improve the models' performance.

Several neural architecture search approaches focus on searching models that achieve highest performance regardless of the model complexity. In this context, neural architecture search approaches for multi-objective problems (Dong et al. 2018) pose an even more significant challenge. Besides, synthetic samples can be created to achieve a less unbalanced distribution of samples between classes, and it is essential to obtain more data from class C5 to be able to extract their characteristics properly.

Conclusions

Aiming to solve the problem of lithology identification, the present study presented an automated technique to find the best configuration of an Artificial Neural Network using a Differential Evolution Algorithm. The results presented here show that the search of the best network configuration also led to the best set of performance metrics, thus proving its robustness in solving the lithology identification problem. In addition, the proposed approach produced a model with good classification accuracy, which

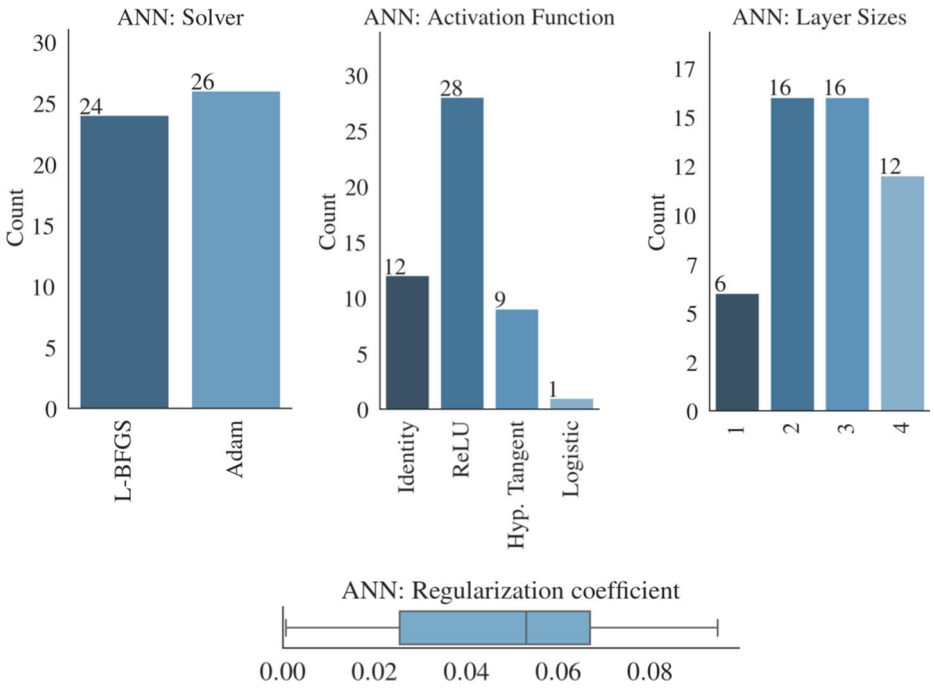

Fig. 4 Distribution of ANN parameters ver 50 independent runs

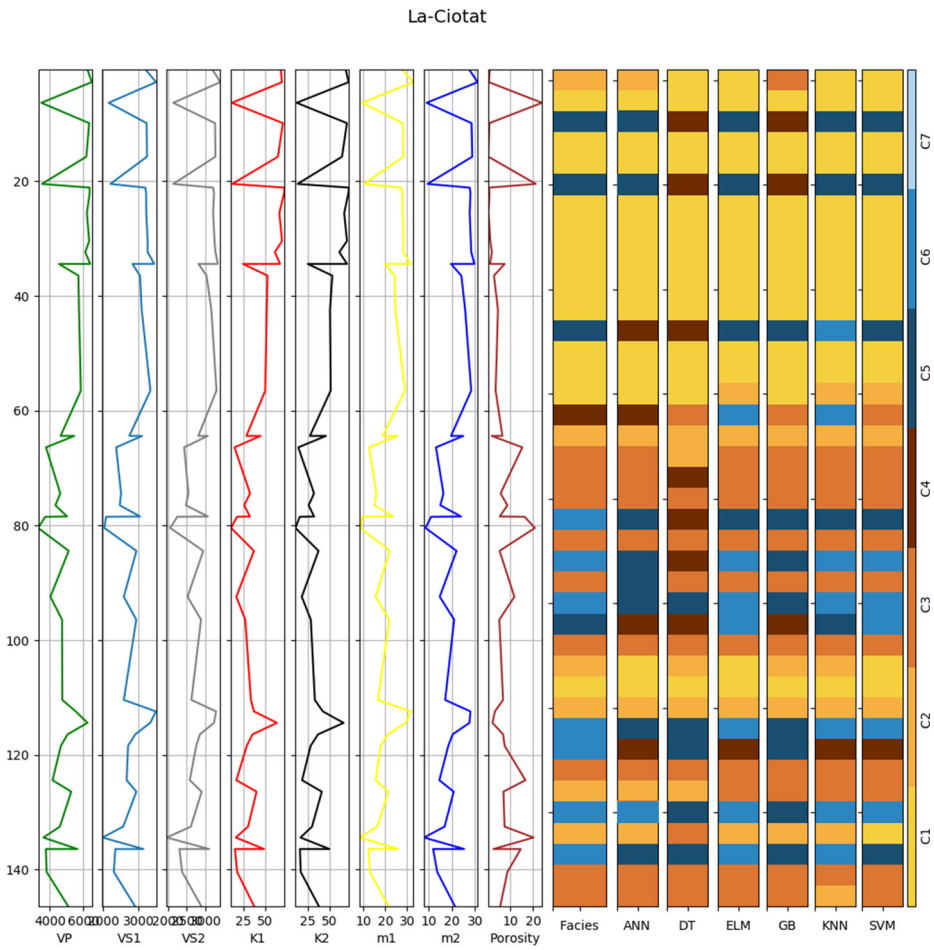

Fig. 5 Comparison of the the current results and those found in Fournier and Borgomano (2009)

can potentially help geologists/petrologists determine the heterogeneity of a reservoir.

Acknowledgments This work was supported by the Federal University of Juiz de Fora (UFJF) and CAPES (Finance Code 001). L.G. thanks FAPEMIG (grant 01606/15) and CNPq (grant 429639/2016-3) for financial support. E.P. thanks CNPq and FAPERJ by research support.

References

Abbey CP, Okpogo EU, Atueyi IO (2018) Application of rock physics parameters for lithology and fluid prediction of TN field of niger delta basin, Nigeria. Egypt J Pet 27(4):853–866

Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10(6):646–657

Brigaud B, Vincent B, Durlet C, Deconinck JF, Blanc P, Trouiller A (2010) Acoustic properties of ancient shallow-marine carbonates: effects of depositional environments and diagenetic processes (middle jurassic, paris basin, france). J Sed Res 80(9):791–807

Burke J, Campbell RJr, Schmidt A et al (1969) The litho-porosity cross plot a method of determining rock characteristics for computation of log data. In: SPE Illinois basin regional meeting. society of petroleum engineers

De Ceia MA, Misságia RM, Neto IL, Archilha N (2015) Relationship between the consolidation parameter, porosity and aspect ratio in microporous carbonate rocks. J Appl Geophys 122:111–121

Dong JD, Cheng AC, Juan DC, Wei W, Sun M (2018) Dpp-net: evice-aware progressive search for pareto-optimal neural architectures. In: The european conference on computer vision (ECCV)

Elsken T, Metzen JH, Hutter F (2019) Neural architecture search. Springer International Publishing, Cham, pp 63–77

Fahlman SE, Lebiere C (1990) The cascade-correlation learning architecture. In: Advances in neural information processing systems, pp 524–532

Fournier F, Borgomano J (2009) Critical porosity and elastic properties of microporous mixed carbonate-siliciclastic rocks. Geophysics 74(2):E93–E109

Geron A (2017) Hands-on machine learning with Scikit-Learn and tensorflow: concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Sebastopol CA

Jaafra Y, Laurent JL, Deruyver A, Naceur MS (2019) Reinforcement learning for neural architecture search: a review. Image Vis Comput 89:57–66

Kitano H (1990) Designing neural networks using genetic algorithms with graph generation system. Complex Syst 4(4):461–476

Liu H, Simonyan K, Vinyals O, Fernando C, Kavukcuoglu K (2017) Hierarchical representations for efficient architecture search. arXiv:1711.00436

Matonti C, Lamarche J, Guglielmi Y, Marié L (2012) Structural and petrophysical characterization of mixed conduit/seal fault zones in carbonates: example from the castellas fault (se france). J Struct Geol 39:103–121

Min X, Pengbo Q, Fengwei Z (2020) Research and application of logging lithology identification for igneous reservoirs based on deep learning. J Appl Geophys 173:103929

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830

Pour AB, Hashim M, Hong JK, Park Y (2017) Lithological and alteration mineral mapping in poorly exposed lithologies using landsat-8 and aster satellite data: North-eastern graham land, antarctic peninsula Ore Geology Reviews

Saporetti CM, da Fonseca LG, Pereira E (2019) A lithology identification approach based on machine learning with evolutionary parameter tuning. IEEE Geosci Remote Sens Lett 16(12):1819–1823

Saporetti CM, Duarte GR, Fonseca TL, Da Fonseca LG, Pereira E (2018) Extreme learning machine combined with a differential evolution algorithm for lithology identification. RITA 25(4):43–56

Saporetti CM, Da Fonseca LG, Pereira E, De Oliveira LC (2018) Machine learning approaches for petrographic classification of carbonate-siliciclastic rocks using well logs and textural information. J Appl Geophys 155:217–225

Silva AA, Neto IAL, Misságia RM, Ceia MA, Carrasquilla AG, Archilha NL (2015) Artificial neural networks to support petrographic classification of carbonate-siliciclastic rocks using well logs and textural information. J Appl Geophys 117:118–125

Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Glob Opt 11(4):341–359

Thomas E, Sneider R, Neasham J, Vinegar H (1995) A catalog of petrophysical and geological properties of typical reservoir rocks. Update Published as Technical Progress Report BTC, pp 12–95

Tian M, Omre H, Xu H (2020) Inversion of well logs into lithology classes accounting for spatial dependencies by using hidden markov models and recurrent neural networks. J Petrol Sci Eng p 107598

Tsai CF, Eberle W, Chu CY (2013) Genetic algorithms in feature and instance selection. Knowl-Based Syst 39:240–247

Tuv E, Borisov A, Runger G, Torkkola K (2009) Feature selection with ensembles, artificial variables, and redundancy elimination. J Mach Learn Res 10:1341–1366

Vasini EM, Battistelli A, Berry P, Bonduà S, Bortolotti V, Cormio C, Pan L (2018) Interpretation of production tests in geothermal wells with t2well-EWASG. Geothermics 73:158–167

Wu G, Shen X, Li H, Chen H, Lin A, Suganthan P (2018) Ensemble of differential evolution variants. Inf Sci 423:172–186

Xie L, Yuille A (2017) Genetic cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1379–1388

Xie Y, Zhu C, Zhou W, Li Z, Liu X, Tu M (2018) Evaluation of machine learning methods for formation lithology identification: a comparison of tuning processes and model performances. J Pet Sci Eng 160:182–193

Yang H, Pan H, Ma H, Konaté AA, Yao J, Guo B (2016) Performance of the synergetic wavelet transform and modified k-means clustering in lithology classification using nuclear log. J Pet Sci Eng 144:1–9

Yang H, Xu Y, Peng G, Yu G, Chen M, Duan W, Zhu Y, Cui Y, Wang X (2017) Particle swarm optimization and its application to seismic inversion of igneous rocks. Int J Mining Sci Technol 27(2):349–357

Zeng L, Ren W, Shan L (2020) Attention-based bidirectional gated recurrent unit neural networks for well logs prediction and lithology identification. Neurocomputing 414:153–171

Zoph B, Vasudevan V, Shlens J, Le QV (2018) Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8697–8710