A comparison of machine learning algorithms as surrogate model for net present value prediction from wells arrangement data

João Roberto Bertini Junior  
School of Technology  
University of Campinas  
Limeira-SP, Brazil  
bertini@ft.unicamp.br

Mei Abe Funcia  
School of Technology  
University of Campinas  
Limeira-SP, Brazil  
m211891@dac.unicamp.br

Antonio Alberto S. Santos  
Center for Petroleum Studies  
University of Campinas  
Campinas-SP, Brazil  
alberto@cepetro.unicamp.br

Denis J. Schiozer  
Faculty of Mechanical Eng.  
University of Campinas  
Campinas-SP, Brazil  
denis@cepetro.unicamp.br

Abstract—Net Present Value (NPV) measures whether an investment will be profitable within a given period of time. In oil production planning, it consists of an important indicator to evaluate different production strategies. The NPV estimate is calculated on the basis of the production data which are generally obtained by means of numerical simulations, which consider the strategy details and the physical reservoir model. However, the simulator demands high computational resource, which may take hours or days of processing time to evaluate a single strategy, depending on the size of the reservoir model. To speed up this process a simpler model, referred to as a surrogate model, can be used to approximate the simulator output. In this work, we hypothesize that it is possible to predict the NPV using only wells arrangement data as predictors. Moreover, we present a comparison among six machine learning algorithms used as a surrogate model: Linear Regression, K-Nearest Neighbor, Multi-Layer Perceptron, Kernel Ridge Regression, Support Vector Regression, and Gradient Tree Boosting. Results confirm it is viable to predict NPV from wells arrangement data, in special with kernel-based methods.

Index Terms—Surrogate model, NPV estimation, Oil reservoir simulation, Oil production strategy, Regression.

I. INTRODUCTION

The arrangement of wells in an oil reservoir is one of the main characteristics of the production strategy since it greatly influences the NPV and determines the quantity of oil that will be extracted within a given period of time. Due to the geological formation of the reservoir, certain areas are easier to drain than others, thus these areas might be more profitable to add a well. Moreover, when a set of wells is considered, the interaction among them must also be taken into account [1]. Therefore, every possible well arrangement in the field produces a different amount of oil and it is important to have ways of quantifying how good the production strategy will be before making a decision.

NPV is the main indicator to evaluate the economic performance of a production strategy. It accounts for the time value of money, and it is useful to analyze the profitability of an investment planning within a given period of time. Its calculation considers the costs and earns for a given period of analysis. In the case of oil production, cash inflow examples are oil and gas price, while cash outflow comprises costs with water injection and oil and water production, and also the value invested in wells and platforms to perform extraction, and so on [2].

As decision makers are interested in the economic performance of strategies, it is necessary to, somehow, project the NPV from them to base the decision. Nowadays, NPV prediction is carried out by complex computer simulations which are based on solving nonlinear dynamic equations representing the geological model of the reservoir. Also, well position optimization demands thousands of simulator evaluations, many of them resulting in very low NPV. Although this approach is effective, it is only viable to analyze a small number of strategies, especially when the size of reservoir models increases [3]. Therefore, to provide an efficient oil reservoir production management and analysis, a simpler model, referred to as a surrogate model may be used to speed up the selection of a production strategy [4]. Fast models aid the optimization process by replacing the simulator when searching for a production strategy. Thus, good strategies can be confirmed by the simulator, while bad ones can be ignored.

The use of a surrogate models, or proxies, to estimate some value - or set of values - is common in applications in which the measure of interest is somehow hard to obtain. Machine learning methods are among the most frequent alternative to build a surrogate model. Particularly due to the possibility to learn from historical data [5]. In this paper, it is studied the use of a surrogate model to partially replace the numerical simulation in the task of evaluating the NPV given information of well arrangement on the field. It is expected that the surrogate model speeds up the evaluation process, allowing to evaluate many more strategies, and thus enhancing the search space for a good production strategy.

The contribution of this work is twofold. First, to access the possibility to infer the NPV from simple wells arrangement data on the petroleum field, i.e. whether a specific well is
considered or not. And second, to establish a comparison of some machine learning methods when used as a surrogate model under these assumptions. The remainder of the paper is organized as follows. Section II presents some recent related works. Section III presents the reservoir model and the production strategies for the case study presented in this work. Section IV details the methodology adopted to conduct this work along with a brief description of the machine learning methods employed. Section V shows the obtained experimental results and comparative analyses. At last, Section VI draws the conclusions and mentions some future works.

II. RELATED WORKS

This section describes some recent works found in the literature that uses a surrogate model in some aspect of oil production.

Ashleyy et al. [6] have done performance evaluation of surrogate models in heterogeneous reservoirs (reservoirs with variations in rock properties with location). Their surrogate models comprise mathematical functions for the gas-oil recovery ratio. To carry out the study, a generic heterogeneous reservoir was created following a specific normal distribution. The surrogate models were constructed through several simulations for different permeability distributions, with different mean and variance. They found that logarithmic equations for oil recovery and the power law equation for the gas-oil ratio were the model that yielded the best results. In the same study, they also have used Monte Carlo simulations to predict the production of any input permeability distribution. For low values of permeability, a small difference in oil recovery and the gas-oil ratio has been observed among the surrogate models. The models developed in their work had performed very well in an intermediate permeability range to predict oil recovery and the cumulative gas-oil ratio for any input distribution. In addition, the models can be used to study the production forecast and perform sensitivity analyses in heterogeneous reservoirs.

Three different models of artificial intelligence were considered in [7], with the aim of comparing them to predict the production of hydrocarbons from hydraulically fractured wells. The models used were: Least Square Support Vector Machine (LSSVM), Artificial Neural Networks (ANN), and the Response Surface Model (RSM). To carry out the comparison among the models, second-order polynomial equations were used to determine the production of shales. All unknown parameters in the surrogate models (i.e. RSM, LSSVM, and ANN) were obtained using Particle Swarm Optimization PSO. Their research considered eight input parameters for the construction of the model: reservoir permeability, initial dissolved gas ratio, rock compressibility, selective gas permeability, gasoline inclination ratio, initial reservoir pressure, the lower hole, and hydraulic fracture spacing. The outputs of the experiment were the oil recovery factor and the gas-oil ratio produced. All models were evaluated by adjusting the coefficient of determination ($R^2$) and the mean of the Root Mean Square Error (RMSE). After all models were trained with the same data, they were used to predict the production of different scenarios. The conclusion of the researchers was that RSM and LSSVM have better predictive capabilities for oil recovery than ANN. In addition, the LSSVM exhibited the highest precision in relation to the prediction of the gas-oil ratio.

An efficient data-based framework for the Model Predictive Control (MPC) was proposed in [8], using Echo State Networks (ESN) as a predictive model. In their work, the ESN model for MPC was partially linearized, i.e., the free response of the system was maintained completely nonlinear and the forced response was linearized. The authors proposed a method called Practical Nonlinear Model Predictive Controller – PNMPC-ESN, used to control a simplified model of an oil and gas well. The method was able to successfully control the plant, following established constraints and maintaining set point tracking. The authors have obtained two advantages with the PNMPC-ESN structure, which comprise the data capacity of the ESN for efficient identification, even without previous knowledge, and the PNMPC guarantees a more precise response besides not requiring inference algorithms. With this, they found that the use of ESNs for oil and gas production systems is appropriate.

To construct adaptive surrogate models for oil optimization problems, the authors in [9] used ANNs as an approximation function and performed training using an adaptive sampling algorithm. Afterward, training examples were added in by an adaptive sampling algorithm with the aim of increasing the prediction accuracy of the surrogate model. The authors used jackknifing and cross-validation as sampling methods during the recursive training and evaluation steps of the network, respectively. The authors have compared the performance between the actual reservoir model and the surrogate model when used to perform the objective function evaluation. The authors have concluded that the surrogate model overcomes the conventional approach and obtains greater precision of prediction, as well as reducing computational cost.

In [10] the authors have developed an effective technique from the combination of a PSO algorithm with adaptive constraint handling and a static Kriging model (i.e. Gaussian process prediction) to automatically obtain the best water injection management in an oil field in order to increase the NPV. Since PSO does not deal well with constraints alone, a constraint manipulation technique has been integrated with the optimizer, whose goal is to look for the best surrogate model. Another feature of the PSO algorithm is the fact that a high number of objective function evaluations is required. To address this issue, the authors used a surrogate Kriging model, which is trained offline through high-fidelity simulator evaluations. The authors have applied the proposed method to nine problems arising from the management of water in two reservoirs of oil with different operating conditions and control cycles. The results of this experiment demonstrated that the optimizer converged to the best approximations.

Aiming at predicting the economic performance of an uncertain oil reservoir under irrigation, Zanbouri and Salahshoor
[4] have developed a robust modeling methodology with the objective of identifying a set of robust surrogate models with unstructured uncertainties. In their work, the oil reservoir was treated as a Multiple Input and Multiple Output (MIMO) system. This system is integrated with a net present value (NPV) function to synthesize a new robust surrogate model. They have conducted experiments in ten different permeability scenarios so to perform a set of uncertain models, which are processed in several stages to determine a set of intercept models. After modeling, the method has been evaluated as a three-dimensional synthetic reservoir with eight injector wells (water) and four producers wells in egg model. Their model has presented robust performance and satisfactory results for the prediction of the economic model. Moreover, even in the worst scenarios, uncertainty has been limited, and it also has been observed that the proposed model is useful to compensate the impacts of uncertainties on the total economic performance of the oil reservoir.

Using an ANN, the authors in [11] have proposed an application for the surrogate reservoir model to perform the prediction of the well Bottom Flow Pressure (BFP) in a reservoir initially considering different time intervals. The output of the simulation consisted of: oil production, gas rate, mean reservoir pressure, and BFP. The proposed surrogate model was based on a Radial Basis Neural Network (RBNN) to predict the BFP based on the extracted data from the Black Oil Applied Simulation Tool (BOAST). As a conclusion, they realized that their model is capable of supporting rapid analysis, decision optimization and managing the results in less time when compared to conventional reservoir models.

III. THE RESERVOIR MODEL AND PRODUCTION STRATEGIES

The reservoir model used in this work is the UNISIM-I-D, a model that encompasses uncertainties regarding the geological properties of the reservoir. The simulation model UNISIM-I-D was based on the reference model UNISIM-I-R, which has been built with geological data from the Namorado Field, Campos Basin, Brazil. Both, the reference and the simulation model, were developed by UNISIM Reservoir and Simulation Group at Center of Petroleum Studies (Cepetro-Unicamp / Brazil). A detailed description of both models can be found in [12].

The UNISIM-I-R was constructed based on structural, facies and petrophysical model, using data from the Namorado Field. Aiming at predicting reservoir performance by small scale heterogeneities, a grid cell resolution was defined as $25 \times 25 \times 1$ m, discretized into a corner point grid with $326 \times 234 \times 157$ cells (3,408,633 total active cells). Figure 1 shows the structural model and the grid cell resolution for the UNISIM-I-R.

For the creation of the UNISIM-I-D benchmark case, data were collected during four years of production (2013-2017), considering the available information of four historical production wells. Thus the UNISIM-I-D benchmark comprises production data with data from the reference model, UNISIM-I-R.

Moreover, as the geomodel used to build the UNISIM-I-D is the same as the UNISIM-I-R, they had the same resolution. To decrease the computational effort during numeric simulations it was necessary upscaling UNISIM-I-D from a high resolution grid to a medium reservoir scale. The cell scale of the simulation model is defined to reflect the reservoir behavior properly. Therefore, the UNISIM-I-D simulation grid cells measure $100 \times 100 \times 8$m, totaling $(81 \times 58 \times 20$ cells, with 36,739 total active cells), as shown in Fig. 2.

A. Production strategy

In the case study, a production strategy refers to a possible arrangement of wells in the field. Each strategy may consider the combination of three kinds of wells, which are:

- **Historical wells**: Are those wells that had been drilled early to test the reservoir and to collect data so to place other wells. They are basically producer wells which have been placed in the field based on geological data.

- **Producer wells**: These are the wells where oil and gas extraction takes place. At the beginning of the extraction of oil, these wells have a natural pressure, which drives the oil to the surface of the well, but with the exploitation and withdrawal of the raw material, this pressure decreases naturally, requiring the use of external resources to get oil to continue. For this, injector wells are used.

- **Injector Wells**: The pressure regulation of producer wells is controlled by injector wells throughout the life cycle of an oil field. It is possible to carry out injection with several materials, among which water is the most used, due to the low cost and characteristics. According to the intrinsic characteristics of the field, studies should be...
carried out to determine the optimal number of producer and injector wells besides the location of them.

To define the strategies considered in this work, a set of 100 wells have been considered as the search space. Among them, four are historical wells, 47 of them are producer wells and 49 are injector wells. The wells are defined in a fixed location and distributed almost uniformly in the reservoir. For simulation purposes, each well is characterized by coordinates \((i, j, k)\) in the Cartesian plane; where \(i\) and \(j\) correspond to abscissa and ordinate, respectively, and \(k\) represents the depth of the well. Historical wells, noted as \(Hi\), have a depth of one block, orientation in the axis \(k\) direction and length of 19 blocks. The producing wells, noted as \(Pr\), have a depth of two blocks and occupy six blocks in the direction of either \(i\) or \(j\). Producer wells from \(Pr1\) to \(Pr23\) are defined in the direction of axis \(i\), with a fixed \(j\); while producing wells from \(Pr24\) to \(Pr47\) are defined in the direction of axis \(j\), having a fixed value for \(i\). At last, injector wells, noted as \(In\), have a depth of 15 blocks and also are defined with six blocks, in either direction, \(i\) or \(j\). Wells from \(In1\) through \(In27\) are defined in the direction of axis \(i\), while wells from \(In28\) to \(In49\) are defined in the direction of axis \(j\).

A production strategy is composed of a subset of these wells. Figure 3 shows a sketch of a possible production strategy represented as an arrangement of wells.

![Fig. 3: A sketch of a production strategy with seven producer wells and three injector wells.](image)

In the figure it is depicted a configuration of ten wells, where seven of them are producer wells and three of them are injector wells. Each well has a fixed coordinate, which means that \(Pr23\) will always be in the same place, for every strategy to be evaluated including that well. The difference among strategies are whether a particular well is considered or not.

The information of well arrangement together with the reservoir model is then passed to the simulator, which evaluates a strategy according to several measures of interest. The most important indicator to decide among several production strategies is the NPV. The NPV considers all expenses and production revenues, fixed costs, taxes and depreciation of the goods used in the project and the production. Therefore, it is considered the main economic objective function in oil production, and the factor responsible for investment decision in a particular oil field. The NPV can be calculated through Eq. (1).

\[
NPV = \sum_{i=0}^{n} \frac{R_t}{(1 + i)^t}
\]  

(1)

In the equation:

- \(R_t\): the net cash,
- \(i\): the discount rate,
- \(t\): the time of the cash flow,
- \(n\): the final period (total number of periods).

The objective is to maximize NPV by minimizing costs and maximizing earnings. In terms of well arrangement it means that it is desirable a strategy with a reduced number of wells, placed in the position that maximizes extraction. That is why the simulator is used to aid decision process.

IV. METHODOLOGY

This section presents the adopted methodology to conduct the experiments. In Section IV-A, it is detailed how the data set has been defined. Section IV-B presents a brief overview of each machine learning algorithms considered as a surrogate model. In the following, Section IV-C describes how the results have been obtained and how the hyper-parameters of the algorithms under comparison have been adjusted.

A. The data set

The data used in this study were provided by UNISIM-CEPETRO. It corresponds to a set of production strategy, totaling 2,000. Each strategy is a combination of wells, that is, when the well is considered for a certain strategy, it receives a value of 1 and when it is not considered, it receives a value of 0. For each strategy, a value of NPV have been determined by the simulator, which is assumed to be the ground truth and is the objective function to be approximated by the surrogate models.

In this work, however, the only information from wells used as predictors are whether each of them is present or not in a given strategy. The data set has 100 attributes, each representing a different well. Among the 100 wells, four are historical wells, 47 of them are producer wells and 49 injector wells. Each row denotes a different strategy, and it is composed of 0s and 1s, indicating the absence or presence of the well, respectively. Associated to each strategy is the NPV evaluated by the simulator. Figure 4 shows a sketch of how the data set is organized.

In the figure, \(Hi1\) to \(Hi4\), represent the Historical wells, \(Pr1\) to \(Pr47\) represent the producer wells and \(In1\) to \(In49\) the injector wells. The number of wells per strategy varies uniformly in the range \([29, 62]\), while the NPV varies in the range \([9.6010^8, 2.6810^8]\).
<table>
<thead>
<tr>
<th></th>
<th>Hi1</th>
<th>...</th>
<th>Hi4</th>
<th>Pr1</th>
<th>...</th>
<th>Pr47</th>
<th>In1</th>
<th>...</th>
<th>In49</th>
<th>NPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2.5</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Fig. 4: Sketch of the data set organization used in this work.

Still, prior to train a regression model, the responses (NPV), noted as $y_i$, were adjusted according to Eq. (2). And, scaled back prior to compute RMSE and $R^2$.

$$y'_i = \frac{y_i - \min_y}{\max_y - \min_y}$$  \hspace{1cm} (2)

For the experiments, besides the original data set, we also considered reducing its dimensionality using Principal Component Analysis (PCA) [13] with 10 components. PCA uses an orthogonal transformation to convert a set of examples, whose attributes might be correlated, into a preferably smaller set of linearly uncorrelated attributes called principal component. PCA is one of the most popular methods for dimensionality reduction in pattern recognition, therefore it has been selected for the experiments.

B. Machine learning algorithms used as a surrogate model

Prior to introduce the regression methods under comparison, let the data set be represented as $\{x_i\}^N_{i=1}$, where each $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,p})$ is associated to the desired output $y_i \in [0, 1]$, and $f(x_i)$ is the model estimate for $y_i$.

The machine learning regression algorithms considered for the comparison are listed ahead.

1) **Linear Regression (LR):** is one of the most widely used models for regression [14]. LR asserts that the response is a linear function of the inputs and can be written as in Eq. (3).

$$f(x) = \sum_{j=1}^{p} w_j x_j + \epsilon$$  \hspace{1cm} (3)

In the equation, $w$ is the weight vector and $\epsilon$ is the residual error between the linear predictions and the true values. In practice, $w$ can be defined analytically by the least squares method or iteratively by the gradient descent method, for instance. In this work, a simple linear regression has been considered, i.e. an hyperplane has been fitted to the data.

2) **K-Nearest Neighbor (KNN):** for a given unlabeled data example, KNN assigns the mean of the labels of its K-nearest neighbors [14]. Basically the neighbors’ importance, or weights, can be considered uniformly, i.e. each example in the local neighborhood contributes uniformly to the label estimation of a query example; or these weights can be used to favor nearby examples to estimate the response for the query example as in Eq. (4).

$$f(x) = \frac{w_{(1)} y_{(1)} + \cdots + w_{(K)} y_{(K)}}{\sum_{j=1}^{K} w_j}$$  \hspace{1cm} (4)

In the equation, $w_{(1)}$ is the weight associated with the nearest neighbour of the query example $x$ and $y_{(1)}$ is the label of the nearest neighbor of $x$. In this work, the weight between the query example and its j-th nearest neighbor has been defined as $w_j = 1/\text{dist}(x, x_{(j)})$.

For model selection, $K$ was selected in the range $K \in \{1,2,\ldots,11\}$. When considering the original data, the Hamming distance has been used. While the Euclidian distance has been used for the data reduced by PCA.

3) **Multi-layer Perceptrons (MLP):** is a feed-forward neural network composed of Perceptron neurons, which are units that perform a weighted sum of the inputs, and outputs accordingly to a given activation function [15]. Neurons in a MLP are organized in layers; generally, each layer receives inputs only from the previous one. The input layer receives the input connections, the output layer gives the network response, and all layer within, called hidden layers, carries possible different numbers of processing neurons. MLP neural network structures are usually trained with the back-propagation algorithm, which updates the network weights’ iteratively by propagating the error from the output to the input. Often, an algorithm to adaptively control the learning rate is used together with back-propagation in order to improve performance. In this work we have employed the Adam algorithm [15].

The following parameters have been adjusted during model selection: the initial learning rate varied in the set $\{0.001, 0.01, 0.1\}$, the hidden layers structures were $\{(25), (50), (100), (150), (50,50), (100,50), (100,100), (150,100)\}$ where (25) means one hidden layer with 25 neurons. Also, three activation functions have been considered: logistic: $g(x) = 1/(1 + \exp(-x))$, the hyperbolic tangent: $g(x) = (\exp(x) - \exp(-x))/(\exp(x) + \exp(-x))$ and the rectified linear unit function, $g(x) = \max(0,x)$. At last, the L2 penalty parameter varied in the set $\{0.0001, 0.001\}$.

4) **Kernel Ridge Regression (KRR):** associates a kernel function to ridge regression, which allows for a more robust function approximation [13]. The ridge regression relies on finding $w$ and $\lambda$ that minimizes the equation:

$$J(w, \lambda) = \sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda ||w||^2$$  \hspace{1cm} (5)

where $f(x_i) = w^\top x_i + w_0$ and $L$ is the quadratic loss function. It can be shown that $w$ that minimizes Eq. (5) has the form: $w = X^\top(X X^\top + \lambda I_N)^{-1} y$. The kernelized version is then obtained by defining the dual variables, as in Eq. (6), where $K$ is the Gram matrix.

$$\alpha = (K + \lambda I_N)^{-1} y$$  \hspace{1cm} (6)

Therefore, $w$ can be rewritten as $w = \sum_{i=1}^{N} \alpha_i x_i$ and predictions requires to sum over the entire data set, as denoted in Eq. (7).
\[ f(x) = \sum_{i=1}^{N} \alpha_i x_i^T x \] (7)

For model selection, \( \lambda \in \{1, 0.1, 10^{-2}, 10^{-3}\} \) and two kernels have been considered, the polynomial kernel \( K(x_i, x_j) = (x_i^T x_j + c)^d \), with \( c = 1 \) and degree \( d \in \{2, 3, 4\} \), and the RBF kernel \( K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2) \) with \( \gamma \in \{10^{-4}, 10^{-3}, 10^{-2}\} \).

5) Support Vector Regression (SVR): As commented earlier, KRR is a kernel-based procedure, but it is not sparse, meaning that for every new prediction a sum over all the data set is necessary. The SVR formulation replaces the quadratic loss, \( L \), with a loss function that ensure the solution is sparse, so predictions only depend on a subset of the training data, known as support vectors [16].

The loss function used by SVR does not penalize the residuals, \(|y_i - f(x_i)|\), that are smaller than a given threshold \( \epsilon \). The \( \epsilon \)-insensitive loss function is denoted in Eq. (8).

\[ L_{\epsilon}(y_i, f(x_i)) = \begin{cases} 0, & \text{if } |y_i - f(x_i)| \leq \epsilon \\ |y_i - f(x_i)| - \epsilon, & \text{otherwise} \end{cases} \] (8)

For the SVR, a slightly different cost function is considered, as shows Eq. (9), where \( C \) is a penalty parameter of the error term.

\[ J(w) = C \sum_{i=1}^{N} L_{\epsilon}(y_i, f(x_i)) + \frac{1}{2} ||w||^2 \] (9)

SVR results have been obtained by considering radial basis and polynomial kernels. For the radial basis, the kernel parameter varied in the range \( \gamma \in \{10^{-4}, 10^{-3}, 10^{-2}\} \); the degree of the polynomial kernel function were considered in the set \( d \in \{2, 3, 4\} \) and the cost parameter \( C \in \{1, 10, 100, 1000\} \).

6) Gradient Tree Boosting (GTB): For regression is an ensemble of weak base prediction models, which are typically regression trees. An ensemble is a set of base learners with a single output, which is some combination of the base learners’ output [14].

As a boosting algorithm, GTB builds the ensemble sequentially, aiming at correcting the errors of the previous learner. Thus, the next tree to be added in the ensemble is trained with the residuals of the last one. GTB allows optimization considering arbitrary differentiable loss function, \( L \). The residual for the \( m \)-th tree for the example \( x_i \), \( r_{im} \) is calculated as in Eq. (10).

\[ r_{im} = -\left[ \frac{\partial L(y_i, f_{m-1}(x_i))}{\partial f_{m-1}(x_i)} \right] \] for \( i = 1, \ldots, n \) (10)

A new regression tree is fitted to the targets \( r_{im} \); its leaves then yields the terminal regions \( R_{jm}; j = 1, 2, \ldots, Jm \). For each region, the tree output is defined as in Eq. (11).

\[ \gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma) \] (11)

Then, the model output is updated according to Eq. (12), where \( I \) is the indicator function, i.e. \( I(x \in R_{jm}) = 1 \), if \( x \in R_{jm} \), and \( I(x \notin R_{jm}) = 0 \) if \( x \notin R_{jm} \).

\[ f_{m}(x) = f_{m-1}(x) + \sum_{j=1}^{Jm} \gamma_{jm} I(x \in R_{jm}) \] (12)

For model selection the following hyper-parameters have been tuned. The minimum percentage of examples to consider further splitting a node \( \text{min\_split} \in \{0.05, 0.1, 0.2, 0.3\} \). The number of estimators to be added during boosting \( M \in \{50, 100, 150\} \), and the loss function which was: the least squares regression, the least absolute deviation and Huber. Please refer to [14] for details on the loss functions.

C. Model selection and evaluation

Model selection and model evaluation has been performed through a 5-times nested 5-fold cross-validation process. This strategy encompasses two sampling procedures; one nested inside the other. The outer loop, a 5-fold cross-validation, separates a training set with four folds and a testing set with one fold, five times. Each training set created in the outer loop is then re-sampled in the inner loop, which is also a 5-fold cross-validation. The inner loop is responsible for model selection, therefore a cross-validation is done for each possible hyper-parameter combination being considered; each hyper-parameter combination defines a model. The best model in the inner loop is then validated in the validation set, which is the set that had been left out by the outer loop.

The results presented here are the average performance of the best models in the validation sets, obtained in each nested cross-validation round. Such strategy avoids optimistic assumptions about the model, which would be the case when one simply chooses the best value in the test sets.

V. EXPERIMENTS RESULT AND DISCUSSION

This section presents the comparison results for the algorithms presented in IV-B when used as a surrogate model to estimate the NPV, given well arrangement data as input. The algorithms have been implemented in Python using the Scikit-learn package [17], and the experiments have been conducted in a Windows 10 environment, in a Intel Core i7 3.6 GHz and 32 GB RAM machine.

Table I presents the RMSE error and the coefficient of determination \( R^2 \) along with their standard deviations for each algorithm under comparison. Also, each algorithm has been tested with the original data and with a reduced version, produced using PCA by selecting the first ten principal components. The best results for both, original and PCA, are highlighted in bold.

Prior to compare the results among the regression models, consider establishing the worst case scenario. A naive estimator, for instance, one that always assigns the mean NPV independent of the input, has a RMSE of 2.322. Therefore, all the algorithms performed better than the naive estimator, for both, original data and when PCA has been applied.
TABLE I: Root mean squared error and $R^2$ measure comparison for NPV estimation from well arrangement data.

<table>
<thead>
<tr>
<th>Surrogate model</th>
<th>RMSE USD $10^8$</th>
<th>$R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Original</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>1.64 ± 0.21</td>
<td>0.472 ± 0.177</td>
</tr>
<tr>
<td>KNN</td>
<td>2.103 ± 0.374</td>
<td>0.126 ± 0.346</td>
</tr>
<tr>
<td>MLP</td>
<td>1.494 ± 0.339</td>
<td>0.538 ± 0.283</td>
</tr>
<tr>
<td>KRR</td>
<td>1.283 ± 0.165</td>
<td>0.689 ± 0.082</td>
</tr>
<tr>
<td>SVR</td>
<td>1.409 ± 0.149</td>
<td>0.602 ± 0.092</td>
</tr>
<tr>
<td>GTB</td>
<td>1.553 ± 0.162</td>
<td>0.536 ± 0.098</td>
</tr>
<tr>
<td><strong>PCA</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>1.641 ± 0.213</td>
<td>0.466 ± 0.161</td>
</tr>
<tr>
<td>KNN</td>
<td>1.844 ± 0.301</td>
<td>0.357 ± 0.179</td>
</tr>
<tr>
<td>MLP</td>
<td>1.633 ± 0.239</td>
<td>0.493 ± 0.132</td>
</tr>
<tr>
<td>KRR</td>
<td>1.645 ± 0.325</td>
<td>0.476 ± 0.210</td>
</tr>
<tr>
<td>SVR</td>
<td>1.678 ± 0.213</td>
<td>0.482 ± 0.141</td>
</tr>
<tr>
<td>GTB</td>
<td>1.596 ± 0.310</td>
<td>0.496 ± 0.261</td>
</tr>
</tbody>
</table>

From the results for the original data in Table I, the best performance was obtained by KRR, followed by SVR, both kernel-based methods. The first four ranked methods are known to approximate well high non-linear data. KNN has provided the worst result when applied to original data. The Hamming distance has shown to be an ineffective measure to be used in well arrangement data.

Considering the results for the PCA data, except for LR and KNN, all the algorithms have had worse performance with the data reduced by PCA. In the case of the KNN it can be explained due to the use of the Euclidian distance instead of the Hamming distance.

Figures 5 to 10 show the scatter plot of the surrogate model predictions against the simulator output for NPV estimation for each algorithm under comparison. The figures’ axis values are scaled by $10^9$. Each figure depicts the results of the best surrogate model found in the nested cross-validation process when applied to the whole data set. Also, for each best model, it is shown its RMSE and $R^2$ in the caption of the figures.

Fig. 5: Best LR surrogate model predictions against the simulator output for NPV estimation on (a) Original data (RMSE: $1.205 \times 10^8$ $R^2$: 0.637) and (b) PCA reduced data (RMSE: $1.437 \times 10^8$ $R^2$: 0.388).

Fig. 6: Best KNN surrogate model predictions against the simulator output for NPV estimation on (a) Original data (RMSE: $0.611 \times 10^8$ $R^2$: 0.927) and (b) PCA reduced data (RMSE: $1.357 \times 10^8$ $R^2$: 0.387).

Fig. 7: Best MLP surrogate model predictions against the simulator output for NPV estimation on (a) Original data (RMSE: $1.207 \times 10^8$ $R^2$: 0.629) and (b) PCA reduced data (RMSE: $1.351 \times 10^8$ $R^2$: 0.492).

Fig. 8: Best KRR surrogate model predictions against the simulator output for NPV estimation on (a) Original data (RMSE: $0.374 \times 10^8$ $R^2$: 0.971) and (b) PCA reduced data (RMSE: $1.377 \times 10^8$ $R^2$: 0.483).

Fig. 9: Best SVR surrogate model predictions against the simulator output for NPV estimation on (a) Original data (RMSE: $1.112 \times 10^8$ $R^2$: 0.621) and (b) PCA reduced data (RMSE: $1.392 \times 10^8$ $R^2$: 0.455).

The figures and the error measures for the best models corroborates the results of Table I. The best model correspond to the model that would be used in practice. It is important to highlight that most of the algorithms considered in this comparison are stochastic, thus it is not guaranteed to obtain the same model every time it is trained. Also, some of
them presents higher variability than others, which explains the difference between the average performance and the best model performance.

Now, considering analyzing the performance of the algorithms when taking the original data as input. From the Figs. 8, 9 and 10 it can be concluded that such algorithms can be effectively used as a surrogate model for NPV estimation. In particular, KRR best model has obtained $R^2 = 0.971$. Notice that, KNN has obtained the best model with $R^2 = 0.927$ mainly due to exactly predicting the training data, however, it present high variance on test data, which makes it less prone to be used as a surrogate model.

Overall, machine learning based surrogate models, as the KRR, SVR or the GBT, can be successfully used to fully replace the simulator or to help to reduce the number of simulation runs in an application that requires many executions, as when using a genetic algorithm searching for the best production strategy [18].

When considering the results obtained with the data reduced with PCA, however, the results were not good. All the algorithm had presented similar performance, which may be due to all the features describing the data having similar importance. In practice, it might be an issue when the size of the number of considered wells enhances too much, for instance to millions of possible coordinates. If that is the case, other kinds of dimensionality reduction might be tried or even another kind of data representation.

VI. CONCLUSIONS

Net present value is the main indicator in oil production planning. Decision makers make use of complex, time consuming, numerical simulations to predict the NPV of a production strategy. This work presents evidence that it is possible to efficiently infer the NPV using only well arrangement data. The case study considered simulation data from the reservoir model of the Namorado Field, Brazil. Experimental results comparing six machine learning algorithm used as a surrogate model showed that NPV estimation can be done through well arrangement data. Among the tested algorithms, Kernel Ridge Regression has yielded the provided the best fit to the data, followed by Support Vector Regression and MLP networks. Future works include considering further machine learning algorithms, especially those based on kernel, as a surrogate model together with other dimensionality reduction algorithms. Also, incremental learning algorithms should be considered to continue refining the model during application, especially for oil fields with few historical data.

REFERENCES