Predicting oil field production using the Random Forest algorithm

Isabel F. A. Gonçalves, Thiago M. D. Silva, Abelardo B. Barreto and Sinesio Pesco
Pontifical Catholic University of Rio de Janeiro
Department of Mathematics, Rio de Janeiro, Brazil
Email: isabelfagoncalves@gmail.com, thiagoomenez@gmail.com, abelardo.puc@gmail.com and sinesio@puc-rio.br

Abstract—**Precisely forecasting oil field performance is essential in oil reservoir planning and management. Nevertheless, forecasting oil production is a complex nonlinear problem due to all geophysical and petrophysical properties that may result in different effects with a bit of change. All decisions to be made during an exploitation project needs to be made considering different efficient algorithms to simulate data, providing robust scenarios to lead to the best deductions. To reduce the uncertainty in the simulation process, researchers have efficiently introduced machine learning algorithms for solving reservoir engineering problems because they can extract the maximum information from the dataset. Accordingly, this paper proposes using a Random Forest model to predict the daily oil production of an offshore reservoir. In this study, the oil rate production is considered a time series and was pre-processed and restructured to fit a supervised learning problem. We use the Random Forest model to forecast a one-time step, which is an extension of decision tree learning, widely used in regression and classification problems for supervised machine learning. For testing the robustness of the proposed model, we use the Volve oil field dataset as a case study to conduct the experiments. The results indicate that the Random Forest model could adequately estimate the one-time step of the oil field production.**

I. Introduction

Predicting oil field performance plays a vital position in reservoir engineering. The decisions to develop and manage the reservoir depend on this information to generate good oil production results. The risks involved are considerably high, demanding proper uncertainty administration during an exploitation project. Likewise, reservoir characterization is essential when forecasting an oil deposit's performance and administering uncertainty. As a result, constructing a robust reservoir model is, therefore, an important task. The process of characterizing the reservoir may be executed by incorporating observed dynamic data from a real field in a model, which is a popular technique called history matching.

The primary tool for history matching algorithms is reservoir simulation, which demands the creation of a theoretical reservoir model in which the user inputs the static properties and the simulation process computes the dynamical data as output. Reservoir simulation processes are crucial for reservoir management, which enables the testing of mixed production plans for forecasting. At the end of the simulation and characterization process, the model is expected to compute output dynamical data similar to the observed data. Nevertheless, reservoir characterization using history matching procedures requires many reservoir simulations, simplified by changing the reservoir model properties until the output data match the observed one. This procedure may take too long and need advanced computational knowledge. Moreover, the algorithms often used in history matching demand complex mathematical or statistical background. As a result, machine learning and artificial neural networks have been the focus of research worldwide to provide proxy models to replace the necessity of flow simulators in some steps of the history matching problems [5].

History matching algorithms [12] are widely known to be efficient in predicting reservoir dynamical properties and oil field production. Moreover, many studies prove that using optimization algorithms may obtain good results. We can mention the nonlinear least square methods, e.g., the Gauss-Newton and Levenberg-Marquardt algorithms [11], and the ensemble-based methods, e.g., the ensemble Kalman filter [12] and the ensemble smoother with multiple data assimilation [19]. The study of Emerick and Shirangi [17] compares the results obtained by applying the Levenberg-Marquardt (LM) and the Gauss-Newton (GN) method. Their results suggest that the LM approach could bring better results when predicting reservoir properties due to the more negligible influence of minimal singular values in the computation of the update vector compared to the GN application. Considering ensemble-based methods, the study of Silva *et al.* [18] offers a good characterization of the damage zone in a multilayered reservoir using the ensemble smoother with multiple data assimilation.

A powerful oil production forecasting tool involves using machine learning algorithms. This technique has become very popular in the last few years due to the easy manipulation and understanding of the mathematical formulation of such algorithms. Moreover, the statistical background of machine learning enables the algorithm to extract the maximum available information from the dataset, which may be unfeasible when not using any data-driven procedures. These algorithms are split into two ample categories: supervised and unsupervised machine learning. We can also mention the algorithms based on reinforcement learning, which recently gained much attention.

The use of machine learning techniques applied in reservoir engineering problems is not entirely new, having a good number of studies published since the beginning of 1990. We can mention the study of Zhou *et al.* (1993) [25], which presents a field example for recognizing lithology from well logs using a fuzzy neural network approach. The authors decided to use this strategy due to the uncertainty, fuzziness, and incompleteness of reservoir engineering problems. The study of Mohaghegh *et al.* (1994) [10] presents an application of artificial neural networks for predicting reservoir heterogeneities such as permeability, porosity, and liquid saturation. This study may be one of the first to introduce neural networks in forecasting reservoir dynamical properties. Another interesting study that uses machine learning procedures applied to reservoir engineering problems is the one presented by Ahmadi et all. [1]. The main objective is to predict the thermodynamics properties of the reservoir fluids. More precisely, they offer an approach to monitoring dew point pressure in retrograde gas condensate reservoirs. To test the method's robustness, they compare the results obtained with the proposed artificial neural network with the classical fuzzy system.

This study presents a data-driven solution for oil production forecasting using a popular machine learning algorithm called Random Forest. The application is included in the supervised machine learning category. The methodology is analogous to the one presented in time-series forecasting studies, where we create a subset X containing vectors X_i, $i \in \mathbb{N}$, of size $N > 0$, corresponding to the past N steps of the time series. For the target set Y, we create subsets Y_i, $i \in \mathbb{N}$, of length one, corresponding to the immediate forward step of the time series related to the vector X_i. In this study, the length of each vector in the variables set X is denoted by *look back*. More precisely, we use supervised machine learning, inputting the target and the variables in the training dataset for prediction. In this study, we evaluate the sensitivity of the Random Forest algorithm, how its formulation deals with large and small training datasets, and the efficiency of predicting a large number of days ahead. We use a real field case to test the proposed technique, containing the production data of the offshore Volve field located in the Norwegian North Sea, available by Equinor.

This study is separated into four sections. In section II, we present the algorithm Random Forest and how it can improve the results of forecasting oil field production. Section III presents some previous works that efficiently used similar techniques as the one proposed by this study. In section IV, the Volve dataset is explained in detail. Finally, section V presents the results of the proposed method.

II. Random Forest Algorithms

CART stands for Classification and Regression Trees. This machine learning algorithm generates binary trees, i.e., a popular data structure in computer science in which each node has exactly two children. These two edges (children) are defined as the left and right children. The splitting decision is made by using an appropriate impurity criterion. The most popular ones are defined as *Gini* or *entropy*. For regression, CART introduced variance reduction using least squares LS (Equation (1)) and Mean Absolute Error MAE (Equation (2)). In Equations (1) and (2) y_i, refers to the prediction for an instance, N is the number of instances and μ is the mean given by $\frac{1}{N}\sum_{i=1}^{N} y_i$.

$$LS = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mu)^2 \tag{1}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \mu| \tag{2}$$

Random Forest [4] is a supervised learning algorithm with an ensemble of size N decision trees built and trained with the input training dataset. One can apply this model for diverse classification and regression problems. The main concern with using decision trees to solve real-case problems is the lowly variance problem widely described in the machine learning literature. It means that if the input data have a slight change compared to the one used as training, one may not compute exactly the change in the output, which can be substantial. Moreover, it is reasonable to expect the model to present a common issue in machine learning applications called overfitting. This point indicates that the model learned too much from the training data but could not generalize the result for other datasets. A traditional procedure that alleviates the low-variance problem of decision trees is called *bagging*, which builds a forest of decision trees and trains each one with the input training dataset. However, instead of training all trees with the whole dataset, it draws a sample of the entire dataset and determines this small sample as the tree's input. This procedure is executed for each tree in the forest. More precisely, given N cases in the training dataset, it samples, with replacement, $k < N$ subsets among all possible cases. In addition, another crucial technique is implemented in each tree of the forest to reduce the problems of low variance and increase the model's generalization capability, which is called *randomized subspace*. The randomized subspace is also applied to the bagging strategy for each tree in the forest. A random sample of the input features and training data is employed in each tree. It is straightforward to expect that each tree in the forest captures slightly different information from the whole dataset, constructing a forest that could provide more variability and robustness to the model. For classification problems, the result of a random forest would be the class with the most appearance in the forest. For regression, the result would be the mean of the outcome of each forest. Figure 1 shows a simple diagram explaining how the random forest algorithm works.

III. Previous works

Random Forest has become a popular machine learning ensemble algorithm in recent years. The diverse applications with good performance in several problems in the industry, including the oil field, may explain this fast popularity increase. Among many applications, we can mention the study of Zhang *et al.* (2019) [24], which developed a hybrid scoring system process by combining conventional screening guidelines and

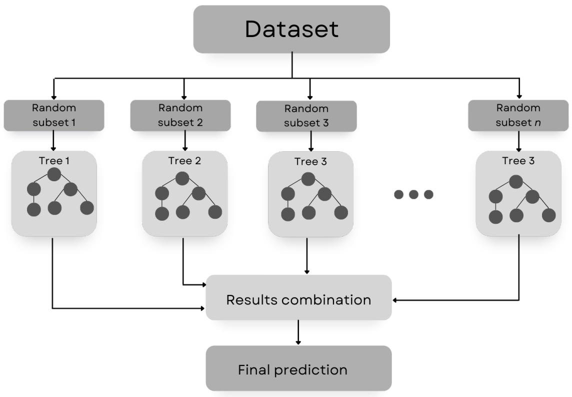

Fig. 1. Simple diagram of how the Random Forest algorithm works..

the random forest algorithm for enhanced oil recovery processes. Wang *et al.* (2021) [22] implemented the random forest algorithm to predict the time-lapse oil saturation profiles at well locations. They used a specific strategy of giving the field-wide production and injection data as the only input parameters. Moreover, they could improve the algorithm using the feature importance module and the Pearson correlation coefficient to optimize the feature selection. Feng *et al.* (2021) [7] applied the random forest algorithm to predict missing well log data. Moreover, they analyze the uncertainty of the proposed method by using prediction intervals through the quantile regression tree. Marins *et al.* (2021) [9] used the random forest algorithm to classify faulty events during the practical operation of oil and gas wells. In their study, the proposed methodology could detect the faults at the beginning of the transient stage. Rahimi and Riahi (2022) [15] operated the random forest algorithm for reservoir facies classification. They concluded that using a decision tree helped to select facies classification for efficient computation. Moreover, they combined geostatistic knowledge with a random forest algorithm for the first time.

The random forest algorithm is also widely used for time series forecasting in different fields of science. Among all available studies, we can mention the study of Qiu and Zhang [14], which predicted electricity load demand from the Australian Energy Market Operator. Wu et al. [23] implemented the random forest algorithm to forecast the weekly upper respiratory infection rate using clinical data from the Shenzhen Health Information Center. Papacharalampous and Tyralis [13] used past streamflow observations and precipitation information to forecast daily streamflow up to seven days ahead with the random forest algorithm. Altinçop and Oktay [2] stated that the random forest algorithm produces accurate results and performs better than artificial neural networks in forecasting time series analysis of air pollution indicators.

IV. Dataset description

The offshore Volve reservoir [6], located in Norwegian North Sea, was discovered in 1993. The field is in the sandstone of Middle Jurassic age at the depth around 2900m. The plan for development was approved in 2005 and productions started in 2008, achieving a peak oil rate of 56,000 bbl/day. The field was decommissioned in 2016 with a cumulative oil production of 63 million barrels.

Equinor and the Volve license partners, ExxonMobil and Bayerngas, have disclosed all seismic records and oil production data from this reservoir in an open repository [6]. Since real data are often prohibitive or challenging to be obtained, the multi-terabyte Volve dataset, containing lots of information on a complete lifetime of a reservoir exploitation project, has been widely used by data scientists and reservoir engineers in their work involving the oil and gas industry. Moreover, academic researchers could test the different complex models they develop in a real field case using the Volve dataset. It has been a substance for research in drilling data, geometric modeling, scientific visualization, and Petroleum reservoir modeling. Tunkiel *et al.* [21] explored the dataset, described common obstacles found in the Volve dataset, and presented approaches for overcoming all the issues. Gupta *et al.* [8] developed a complex workflow to identify the formation type around the bit from surface drilling data. Sun *et al.* [20] build a 3D mechanical earth model of the Volve field. Ravasi *et al.* [16] created a real target-oriented seismic images for Volve reservoirs.

This study uses only the oil production data from well 15/9-F-1 C, contained in the Volve Field. Due to the high number of errors and inaccuracies in the data, it was necessary to pre-process the whole data, guaranteeing that there were no missing values or non-numerical ones. The final data was an uninterrupted 746 days sequence of the oil production information, shown in Figure 2.

V. Results

The 746 daily oil production information from Volve well 5/9-F-1 C was used as input. In all following results, we use the first 500 days as a training set and the remaining 246 days as a test set. In particular, the original data set is restructured into a sliding window dataset, where t time steps are used to predict time step $t + 1$. We refer to t as *look-back* since it is the number of previous time steps to make one-time step prediction ahead.

The Random Forest Regression in the Scikit-Learn package includes some default parameters. If not specified, the number of trees in the forest, usually known as n_estimators, is 100. The function to measure the quality of a split is the Squared error shown in equation 3, where N is the number of samples being tested, y_i is the model prediction, and $\hat{y}$ is the actual value.

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2 \tag{3}$$

By default, the minimum number of samples required to split an internal node is two, and all the nodes are expanded until all leaves are pure or until all leaves contain less than

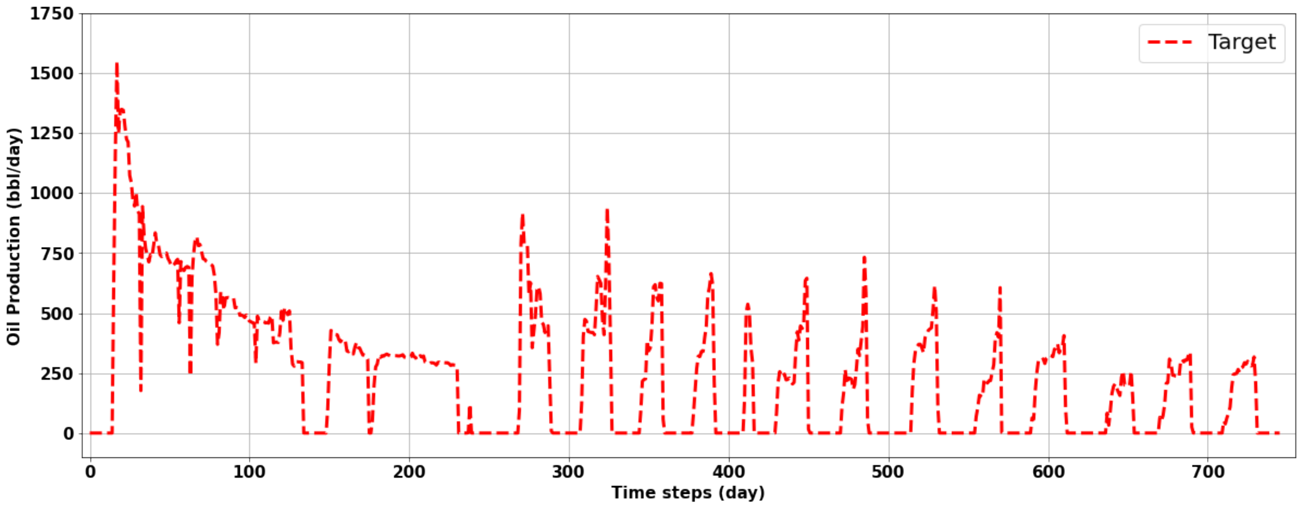

Fig. 2. Well "15/9-F-1 C" daily oil production data contained in the Volve dataset.

two samples. The minimum number of samples required at a leaf node is 1. The minimum weighted fraction of the total sum of weights (of all the input samples) required at a leaf node is 0. *look-back* is the number of time steps to consider when looking for the best split. If not specified, there is no limit for the number of leaf nodes. A node will be split if this split induces a decrease of the impurity greater than ni that is calculated as shown in equation 4, where ni_j is the importance of node j, W_j is weighted number of samples reaching node j, C_j is the impurity value of node j, $W_{left(j)}$ (resp. $W_{right(j)}$) is the weighted number of samples reaching child node from left (resp. right split) on node j and $C_{left(j)}$ (resp. $C_{right(j)}$) is is the impurity value of child node from left (resp. right split) on node j. By default, bootstrap samples are used when building trees, and the number of samples in a subset, usually called max_samples, is equal to the number of samples in the original dataset.

$$ni_j = W_j C_j - W_{left(j)} C_{left(j)} - W_{right(j)} C_{right(j)} \qquad (4)$$

Scikit-learn package also includes GridSearch, a tuning technique that finds optimum parameters. It is an exhaustive search performed with specific sets of parameter values. GridSearch builds a model for every combination of parameters specified in the collection and evaluates each model, returning the best one. To measure the accuracy of the random forest algorithm, we use root mean square error (RMSE), which is a classical way to evaluate the error of a forecasting model. It represented the square root of the mean of the differences between predicted values and observed values, as shown equation 5. where N is the number of time steps in the tested set, $m_{true,k}$ is the observed value in time step k and $m_{j,k}$ is the predicted value for time step k.

$$RMSE = \left(\frac{1}{N_m} \sum_{k=1}^{N_m} (m_{true,k} - m_{j,k})^2 \right)^{1/2}, \qquad (5)$$

In this study, we settle n_estimators as 10, 25, or 50 and max_samples as 50%, 80%, or 100% of the original dataset. All other parameters are used as default. Besides, we structure the dataset with three different *look-back* values: 10, 25, and 50. Hence, for each *look-back*, we use GridSearch to find the best n_estimators and max_samples. Thus, for each *look-back* value, GridSearch builds nine different models and returns the best one, according to RMSE metrics. Note that when *look-back* is settled as t, it is necessary t time steps to make a one-time step ahead prediction, so different values for *look-back* generate different amounts of predictions even if the test set always has the same size.

When *look-back* is 10, GridSearch returns n_estimators as 100 and max_samples as 50%. The final prediction and the target values are shown in Figure 3. RMSE in this test is 0.07. When *look-back* is 25, GridSearch returns n_estimators as 1000 and max_samples as 50%. The final prediction and the target values are shown in Figure 4. RMSE in this test is 0.06. When *look-back* is 50, GridSearch returns n_estimators as 500 and max_samples as 50%. The final prediction and the target values are shown in Figure 5. RMSE in this test is 0.06.

VI. Conclusion

This study proposes applying the random forest algorithm for oil production forecasting. The strategy used was similar to the one presented in the time series forecasting, where the whole dataset is considered time-dependent, and the training dataset is split into small pieces of the entire data. The size of these series pieces is referred to as *look-back*. The target data is the oil production related to that series piece at the immediate forward time step. We use the Volve dataset

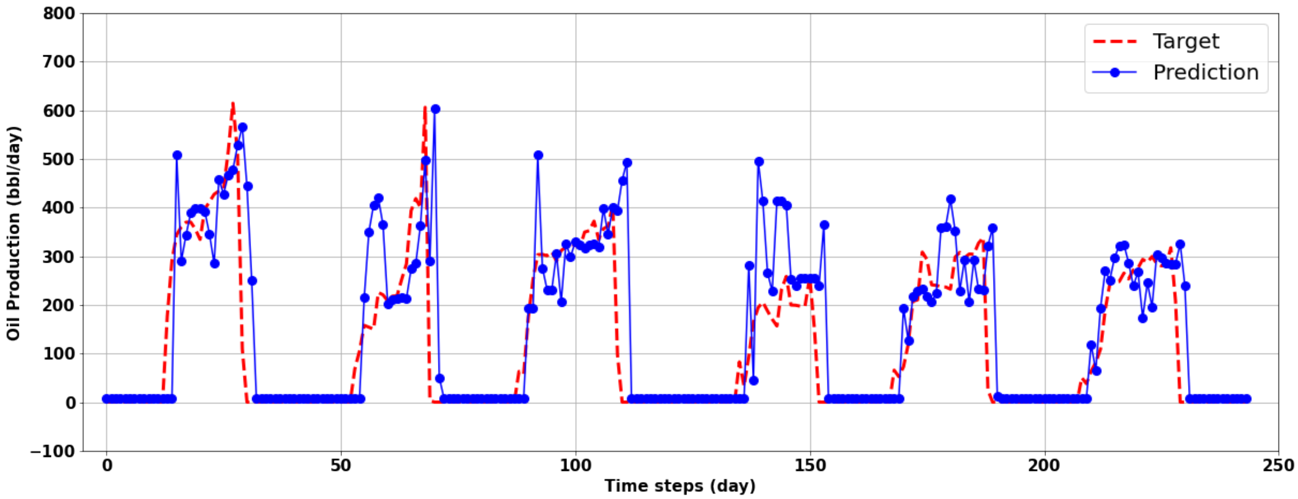

Fig. 3. Oil production forecast in test set with *look-back* =10, n_estimators=100 and max_samples as 50%

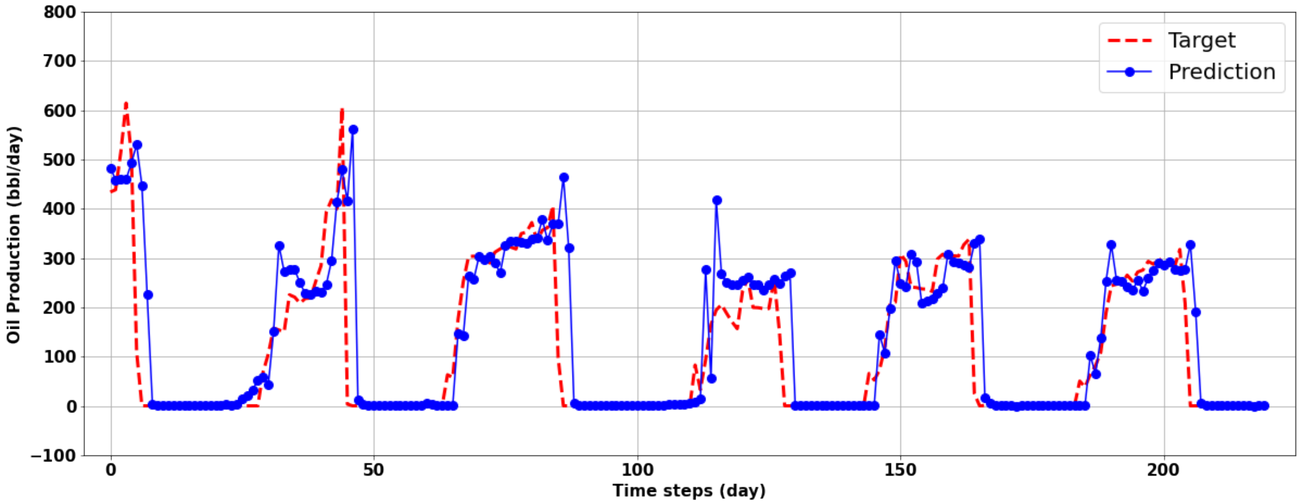

Fig. 4. Oil production forecast in test set with *look-back* =25, n_estimators=1000 and max_samples as 50%

provided by Equinor to evaluate the robustness and the efficiency of the method in predicting high nonlinear physical dynamic data. Moreover, we test the method's performance by applying different values for the *look-back* parameter. We also experiment with some hyperparameters of the theoretical background of the random forest algorithms, such as the forest's number of trees and the bootstrap sampling ratio, to proceed with the bagging technique. We use the root mean squared error (RMSE), a classical error measuring practice, to assess the results. The results obtained by this study suggest that the random forest algorithm could get accurate results when predicting oil field performance with a small value assigned for the RMSE. Moreover, the *look-back* value of 25 was enough to yield good results. The bootstrap sampling ratio of 50% showed to be enough to acquire good final results. Considering the number of trees in the forest, we tested 100, 500, and 1000. However, the result was inconclusive because some experiments found the best fit for using 100 trees and others using 1000 trees. Therefore, we leave this hyperparameter open for further research.

For future works, we plan to continue this work by implementing the random forest algorithm to forecast more than a one-time step ahead. Also, we plan to implement other machine learning techniques to make more accurate predictions considering additional information from a reservoir as inputs.

ACKNOWLEDGMENT

The authors would like to thank PETROBRAS and CAPES for partially support this research.

REFERENCES

[1] M. A. Ahmadi, M. Ebadi, A. Yazdanpanah, ”Robust intelligent tool for estimating dew point pressure in retrograded condensate gas reservoirs:

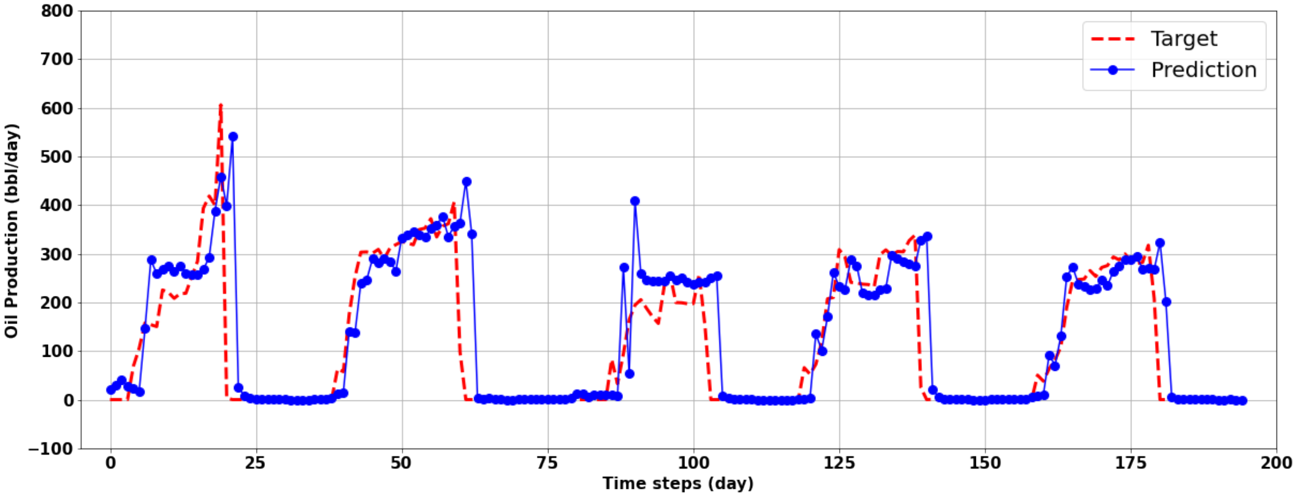

Fig. 5. Oil production forecast in test set with *look-back* =50, n_estimators=500 and max_samples as 50%

Application of particle swarm optimization", *Journal of Petroleum Science and Engineering*, Vol. 123, November 2014, Pages 7-19, DOI: 10.1016/j.petrol.2014.05.023

[2] H. Altinçop, A. B. Oktay, "Air Pollution Forecasting with Random Forest Time Series Analysis", *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, pp. 1-5, 2019, doi: 10.1109/IDAP.2018.8620768

[3] L. Breiman, "Bagging predictors", *Machine Learning*, vol. 24, pp. 123-140, aug. 1996, doi: 10.1007/BF00058655.

[4] L. Breiman, "Random Forests", *Machine Learning*, vol. 45, pp. 5-32, oct. 2001, doi: 10.1023/A:1010933404324.

[5] L. A. N. Costa, C. Maschio, D. J. Schiozer, "Application of artificial neural networks in a history matching process", *Journal of Petroleum Science and Engineering*, Vol 123, November 2014, Pages 30-45, DOI: 10.1016/j.petrol.2014.06.004

[6] Equinor, Volve dataset, https://www.equinor.com/en/news/ 14jun2018-disclosing-volve-data.html, 2018.Accessed: 2022-04-13.

[7] R. Feng, D. Grana, N. Balling, "Imputation of missing well log data by random forest and its uncertainty analysis", *Computers & Geosciences*, vol. 152, 104763, mar. 2021, doi: 10.1016/j.cageo.2021.104763

[8] I. Gupta, N. Tran, D. Devegowda, V. Jayaram, C. Rai, C. Sondergeld, H. Karami. "Looking Ahead of the Bit Using Surface Drilling and Petrophysical Data: Machine-Learning-Based Real-Time Geosteering in Volve Field", *SPE Journal*, vol. 25, pp. 990–1006, 2020, doi: 10.2118/199882-PA

[9] M. A. Marins, B. D. Barros, I. H. Santos, D. C. Barrionuevo, R. E.V. Vargas, T. M. Prego, A. A. Lima, M. L.R. Campos, E A. B. Silva, S. L. Netto, "Fault detection and classification in oil wells and production/service lines using random forest", *Journal of Petroleum Science and Engineering*, vol. 197, 107879, 2021, doi: 10.1016/j.petrol.2020.107879.

[10] S. Mohaghegh, R. Arefi, S. Ameri, M. H. Hefner, "A Methodological Approach for Reservoir Heterogeneity Characterization Using Artificial Neural Networks", *SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, September 1994*, SPE-28394-MS

[11] J. Nocedal, S. Wright, "Numerical Optimization", *Springer-Verlag New York*, 2006

[12] D. S. Oliver, A. C. Reynolds, N. Liu, "Inverse Theory for Petroleum Reservoir Characterization and History Matching", *Cambridge: Cambridge University Press*, 2008

[13] G. A. Papacharalampous, H. Tyralis, "Evaluation of random forests and Prophet for daily streamflow forecasting", *Adv. Geosci.*, vol. 45, pp. 201–208, 2018, doi: 10.5194/adgeo-45-201-2018

[14] X. Qiu, L. Zhang, P. N. Suganthan, G. A. J. Amaratunga, "Oblique Random Forest Ensemble via Least Square Estimation for Time Series Forecasting", *Information Sciences*, vol. 420, pp. 249-262, dec. 2017, doi: 10.1016/j.ins.2017.08.060

[15] M. Rahimi, M. A. Riahi, "Reservoir facies classification based on random forest and geostatistics methods in an offshore oilfield ", *Journal of Applied Geophysics*, vol. 201, 104640, april, 2022, doi: 10.1016/j.jappgeo.2022.104640.

[16] M. Ravasi, I. Vasconcelos, A. Kritski, A. Curtis, C. A. C. Filho, G. A. Meles, "Geophysical Journal International", *55th U.S. Rock Mechanics/Geomechanics Symposium*, vol. 205, pp. 99–104, 2016, doi: 10.1093/gji/ggv528.

[17] M. G. Shiranji, A. A. Emerick, "An Improved TSVD-Based Levenberg-Marquardt Algorithm for History Matching and Comparison with Gauss-Newton", *Journal of Petroleum Science and Engineering*, Vol. 143, July 2016, Pages 258-271, DOI: 10.1016/j.petrol.2016.02.026

[18] T. M. D. Silva, R. V. Bela, S. Pesco, A. B. Barreto, "ES-MDA applied to estimate skin zone properties from injectivity tests data in multilayer reservoirs", *Computers & Geosciences*, Vol. 146, January 2021, 104635, DOI: 10.1016/j.cageo.2020.104635

[19] T. M. D. Silva, S. Pesco, A. B. Barreto, "Influences of the inflation factors generation in the main parameters of the ensemble smoother with multiple data assimilation", *Journal of Petroleum Science and Engineering*, Vol. 203, August 2021, 108648, DOI: 10.1016/j.petrol.2021.108648

[20] Z. Sun, A. Garza, R. Salazar-Tio, A. Fager, B. Crouse, "A Novel 3D Mechanical Earth Modeling of the Volve Field and Its Application to Fault Stability Analysis", *55th U.S. Rock Mechanics/Geomechanics Symposium*, 2021.

[21] A. T. Tunkiel, T. Wiktorski, D. Sui, "Drilling Dataset Exploration, Processing and Interpretation Using Volve Field Data", *ASME 2020 39th International Conference on Ocean, Offshore and Arctic Engineering*, 2020, doi: 10.1115/OMAE2020-18151

[22] B. Wang, J. Sharma, J. Chen, P. Persaud, "Ensemble Machine Learning Assisted Reservoir Characterization using Field Production Data-An Offshore Field Case Study", *Energies*, vol. 14, 1052, feb. 2021, doi: 10.3390/en14041052

[23] H. Wu, Y. Cai, Y. Wu, R. Zhong, Q. Li, J. Zheng, D. Lin, Y. Li, "Time series analysis of weekly influenza-like illness rate using a one-year period of factors in random forest regression", *BioScience Trends*, vol. 11, 3, 2017, doi: 10.5582/bst.2017.01035

[24] N. Zhang, M. Wei, J. Fan, M. Aldhaheri, Y. Zhang, B. Bai, "Development of a hybrid scoring system for EOR screening by combining conventional screening guidelines and random forest algorithm", *Energies*, vol. 256, 115915, aug. 2019, doi: 10.1016/j.fuel.2019.115915

[25] C. D. Zhou, X. Wu, J. Cheng, "Determining Reservoir Properties in Reservoir Studies Using a Fuzzy Neural Network", *SPE Annual Technical Conference and Exhibition, Houston, Texas, October 1993*, SPE-26430-MS