Journal of Petroleum Exploration and Production Technology (2021) 11:3103–3127
https://doi.org/10.1007/s13202-021-01199-x

ORIGINAL PAPER-PRODUCTION ENGINEERING

Application of nature-inspired algorithms and artificial neural network in waterflooding well control optimization

Cuthbert Shang Wui Ng[1] · Ashkan Jahanbani Ghahfarokhi[1] · Menad Nait Amar[2]

Received: 20 April 2021 / Accepted: 26 May 2021 / Published online: 17 June 2021

Abstract

With the aid of machine learning method, namely artificial neural networks, we established data-driven proxy models that could be utilized to maximize the net present value of a waterflooding process by adjusting the well control injection rates over a production period. These data-driven proxies were maneuvered on two different case studies, which included a synthetic 2D reservoir model and a 3D reservoir model (the Egg Model). Regarding the algorithms, we applied two different nature-inspired metaheuristic algorithms, i.e., particle swarm optimization and grey wolf optimization, to perform the optimization task. Pertaining to the development of the proxy models, we demonstrated that the training and blind validation results were excellent (with coefficient of determination, R^2 being about 0.99). For both case studies and the optimization algorithms employed, the optimization results obtained using the proxy models were all within 5% error (satisfied level of accuracy) compared with reservoir simulator. These results confirm the usefulness of the methodology in developing the proxy models. Besides that, the computational cost of optimization was significantly reduced using the proxies. This further highlights the significant benefits of employing the proxy models for practical use despite being subject to a few constraints.

Keywords Waterflooding optimization · Machine learning · Artificial neural network · Data-driven proxy modeling · Nature-inspired algorithms

Introduction

For the past decades, waterflooding or water injection has been one of the most prevalent techniques applied to increase the hydrocarbon production. Waterflooding is termed as the secondary production method that is conducted following the primary production, which is also known as natural depletion. During the phase of natural depletion, hydrocarbon fluid is recovered from the reservoirs by natural forces, such as expansion of fluid and rock, and influx of aquifer. Besides that, tertiary recovery methods, which are known as enhanced oil recovery (EOR), can be another option if secondary recovery is not effective to produce the remaining hydrocarbon. Examples of EOR methods include steam injection and polymer flooding. Pertaining to waterflooding, due to the costs of water production and injection, it is essential for oil and gas companies to carefully plan the schemes of waterflooding to achieve higher economic returns. Such planning is understood as a part of production optimization which is a very vital aspect in reservoir management (Thakur 1996; Udy et al. 2017). Therefore, optimization of waterflooding has been one of the most widely researched topics in the field of petroleum engineering (Van Essen et al. 2009; Zhang et al. 2014; Ogbeiwi et al. 2018; Hong et al. 2019).

Fundamentally, waterflooding optimization involves the adjustment of some relevant variables to maximize the predefined objective function, like net present value (NPV), total oil production, etc., over a period. Additionally, this period can be at least in the horizon of several years or decades. Hence, it is considered as a long-term optimization problem. More intriguingly, different types of algorithms can perform this optimization as discussed in Udy et al. (2017). In general, these algorithms can be either derivative based or derivative free. Udy et al. (2017) further expounded the benefits and drawbacks of implementing derivative-based algorithm, like adjoint method and derivative-free algorithms,

✉ Cuthbert Shang Wui Ng
cuthbert.s.w.ng@ntnu.no

[1] Department of Geoscience and Petroleum, Norwegian University of Science and Technology, Trondheim, Norway

[2] Département Etudes Thermodynamiques, Division Laboratoires, Sonatrach, Boumerdes, Algeria

including particle swarm optimization (PSO), simulated annealing (SA), and genetic algorithm (GA). Moreover, waterflooding optimization can in general be categorized into three different types, namely well control optimization (generally comprising either bottom-hole pressure (BHP) or rates optimization) (Sarma et al. 2008; Zhang et al. 2014; Lu et al. 2017), well placement optimization (Guyaguler et al. 2002; Forouzanfar and Reynolds 2013; Volkov and Bellout 2017), and combination of these methods together or with other variables, such as number of wells (Bellout et al. 2012; Forouzanfar and Reynolds 2014; Pouladi et al. 2020). In this context, the potential of waterflooding optimization for continuous improvement for more practical applications has been demonstrated.

Numerical reservoir simulation (NRS) is one of the most standardized tools utilized in the oil and gas industry to conduct the subsurface or reservoir modeling. NRS can be conveniently (and is also frequently) coupled with any mathematical algorithm to optimize waterflooding or any EOR techniques. This has also been one of the most common practices in the industry as highlighted in some literatures (Peaceman 1977; Jansen et al. 2009; Ertekin and Sun 2019; Baumann et al. 2020). Nonetheless, as perceived, NRS is developed based upon the physics to model the behavior of fluid flow in porous media. Therefore, when the system modeled becomes more complex, e.g., increased heterogeneity of the reservoir, the transport of fluid in porous media will be more difficult to be solved mathematically (Mohaghegh 2017a). This implies that the time required to complete the computation of NRS will increase drastically. Consequently, this might lead to certain level of economic loss. Fortunately, thanks to the establishment of proxy modeling, the computational challenge can be mitigated. In this aspect, the word "proxy" denotes "to act on behalf of another." This denotes that proxy models are the replica of numerical reservoir models which can be readily employed for practical applications in the industry (Mohaghegh 2011; Ertekin and Sun 2019).

With respect to this, proxy models are alternatively known as data-driven models because their building blocks are made up of different sets of data. Hence, proxy models are believed to be able to replicate the results of NRS accurately if the data used to develop them are representative of the physics being modeled. As Mohaghegh (2017b) has counseled, there are two main classes of proxy modeling, which are reduced-order models (ROMs) and response surface models (RSMs). For ROMs, the simplification of the physics is involved and one of the most used examples of ROMs is capacitance resistance models (CRMs). CRMs were developed by Bruce (1943) and reinitiated by Yousef et al. (2006) to determine inter-well connectivity. Application of CRMs in waterflooding has also been proven to be useful in some literatures (Liang et al. 2007; Sayarpour et al. 2007; Hong et al. 2017). Besides that, RSMs are considered as statistical approaches which attempt to develop a predefined form of mathematical function, e.g., linear, polynomial, etc., based on the data given (Mohaghegh 2017b). There are also some papers (Valladão et al. 2013; Babaei and Pan 2016) that discuss the use of RSMs in waterflooding. Despite this, Mohaghegh (2017a, b) has opined that these classes of proxy modeling involve underlying assumptions and simplifications that can impede capturing the actual physics from pattern recognition of data provided. Thus, he has coined another class of proxy modeling that is built based upon machine learning (ML) techniques and artificial intelligence (AI), which has been named as "smart proxy modeling" (SPM). The word "smart" indicates the ability of the models to learn the pattern of the data provided through the ML and AI techniques. He has also initialized the term "Petroleum Data Analytics" (PDA) that focuses on the use of data-driven analytics and big data in the upstream of petroleum industry (Mohaghegh 2017a, b) and SPM is undeniably a part of PDA. According to Mohaghegh (2017b), smart proxy models consist of an ensemble of neuro-fuzzy systems that can duplicate the results yielded by NRS and readily to be utilized for different purposes, like history matching (He et al. 2016; Shahkarami et al. 2018; Shahkarami and Mohaghegh 2020), uncertainty quantification (Mohaghegh 2006; Mohaghegh et al. 2006, 2012), utilization of CO_2 (Shahkarami et al. 2014; Amini and Mohaghegh 2019; Vida et al. 2019; Shahkarami and Mohaghegh 2020), waterflooding (Alenezi and Mohaghegh 2017), and analysis of shales (Kalantari-Dahaghi and Mohaghegh 2011; Mohaghegh 2013; Mohaghegh et al. 2017). Also, it is very important to understand that SPM is an objective-directed task in which the purpose of the proxies needs to be notified first prior to development. Having this understanding will help the modelers to have a better idea of what data can be useful in the development of proxy models.

There are also other interesting literatures (Nait Amar et al. 2018, 2020; Navrátil et al. 2019; Alakeely and Horne 2020; Ng et al. 2021) that discuss and present the use of ML methods in the establishment of proxies of numerical models in petroleum domain, especially for reservoir engineering. Regarding this, there is a riveting insight being provided by Nait Amar et al. (2018) about the modeling of proxies, which is the difference between static and dynamic proxy models. They discussed that in static proxies, the models were not developed as the function of time. Hence, these models were built to yield the results of a predefined variable, such as NPV and total oil production at a particular time (normally at the end of simulation). In this context, Guo and Reynolds (2018) applied support vector regression (SVR) to build a static proxy model that predicted the NPV as a function of control sets by considering different geological realizations. Then, the static proxy was maneuvered to

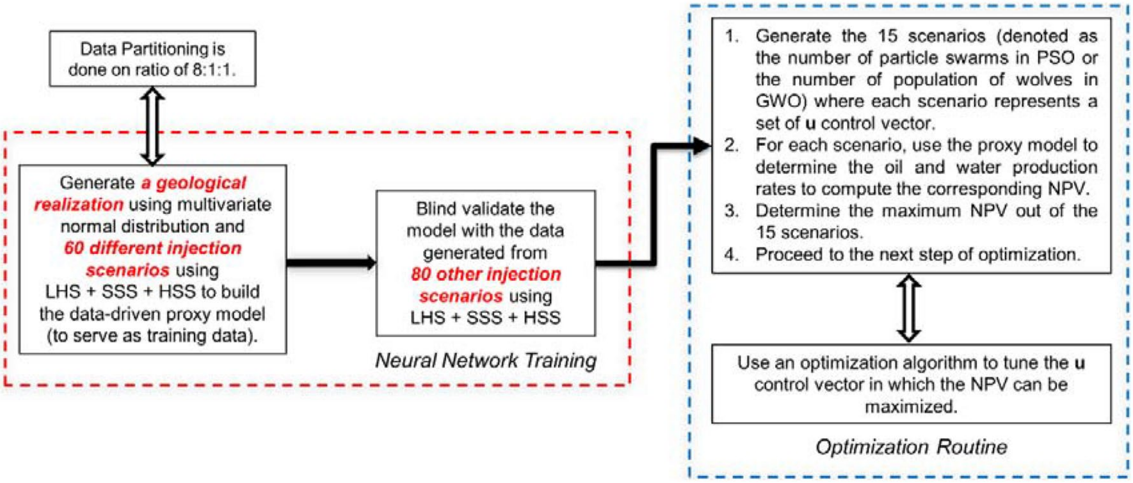

Data Partitioning is done on ratio of 8:1:1.
Generate a geological realization using multivariate normal distribution and 60 different injection scenarios using LHS + SSS + HSS to build the data-driven proxy model (to serve as training data).
Blind validate the model with the data generated from 80 other injection scenarios using LHS + SSS + HSS
Neural Network Training
1. Generate the 15 scenarios (denoted as the number of particle swarms in PSO or the number of population of wolves in GWO) where each scenario represents a set of u control vector.
2. For each scenario, use the proxy model to determine the oil and water production rates to compute the corresponding NPV.
3. Determine the maximum NPV out of the 15 scenarios.
4. Proceed to the next step of optimization.
Use an optimization algorithm to tune the u control vector in which the NPV can be maximized.
Optimization Routine

Fig. 1 General workflow of the methodology of data-driven proxy modeling and optimization

perform robust production optimization. Moreover, Wang et al. (2021) presented the application of PSO in tuning the hyperparameters of SVR that was developed as static proxies to forecast the NPV and cumulative oil production. Thereafter, this PSO–SVR model was coupled with non-dominated sorting genetic algorithm-II (NSGA-II) to conduct the Pareto optimization. Albeit the use of static proxies has been successfully shown, they articulated that the dynamic proxies (established as a function of time) offered more practical applicability and flexibility to be used, notably under time-dependent constraints (Nait Amar et al. 2018). In the context of waterflooding, Golzari et al. (2015) applied artificial neural network (ANN) modeling to build a dynamic proxy and coupled it with GA to optimize the production. They also integrated cross-validation and Jackknife Variance to evaluate the quality of the proxies and perform adaptive sampling to add new training data if necessary. Also, Teixeira and Secchi (2019) employed ANN to develop two dynamic proxies: one that could forecast the oil production rates as a function of injection rates and past oil rates and another one was to approximate the same output by having injection rates and BHP of producers as inputs.

In this paper, one of the goals is to present how dynamic proxy models can be developed based upon the data generated by the NRS models. There are two different NRS models, 2D and 3D reservoirs, being analyzed in this work. The purpose of the proxies is to be employed to carry out the well control optimization. About the proxy modeling, the ML technique that has been applied is ANN and the corresponding training algorithm is adaptive moment estimation (Adam). Furthermore, we couple these proxies with two different nature-inspired metaheuristic algorithms, namely particle swarm optimization (PSO) and grey wolf optimization (GWO) to run the respective optimization. These algorithms would also be utilized with the NRS models for comparative analysis. After this introduction, we will explicate the formulation of the optimization problem and the methodology used to establish the proxies. In this aspect, we also provide brief discussion about ANN, PSO, and GWO. Thereafter, we explain the background of the reservoir models and illustrate the respective results of the ANN training as well as the optimization study. The discussion will then follow prior to proceeding to the conclusions.

Methods

The entire workflow utilized to build and apply the data-driven models for the optimization of waterflood is summarized in Fig. 1. The workflow can be classified into two main parts, which include neural network training (also known as proxy modeling) and optimization routine. Prior to developing the data-driven proxies, it is essential to identify the purpose of these models as proxy modeling is an objective-directed task. In this paper, the objective is to maximize the NPV of a waterflooding project by adjusting the control of injection rate of each well periodically (every 150 days) over 3000 days. Besides that, the control of each injector is tuned within the range of 40 m^3/day and 100 m^3/day.

The NPV is expressed as shown in Eq. (1). Since the reservoir models presented in this paper are only oil–water systems, gas production rate is not considered in the formulation of NPV.

$$\mathrm{NPV}(\mathbf{u}) = \sum_{j=1}^{n_{total}} \frac{\left(Q_o^j(\mathbf{u})P_o - Q_w^j(\mathbf{u})C_w - Q_{wi}^j(\mathbf{u})C_{wi}\right) \times \Delta t_j}{(1+b)^{t_j/D}} \tag{1}$$

$$\mathbf{u} = \left[u_1, u_2, u_3, \ldots, u_M\right]^T \tag{2}$$

where **u** is the control vector (e.g., control rates or BHP), M is the number of control variables, Q_o^j is the field oil production rate at timestep j, Q_w^j is the field water production rate at timestep j, Q_{wi}^j is the field water injection rate at timestep j, Δt_j is the time difference between timestep j and previous timestep, t_j is the cumulative time until timestep j that is used to discount the cashflow, and D is the reference period for discounting. In this paper, D is set to be 365 days because interest rate, b is in the unit of fraction per year and the cashflow is discounted every day. P_o, C_w, and C_{wi} correspondingly mean oil price, cost of water production, and cost of water injection. According to Eq. (1), there are two important parameters we aim to obtain, either directly or indirectly, from the proxy models. These parameters are field oil and water production rates. Based on our analysis and investigation, we established two different proxy models, where one could predict the field liquid production rates at a specific timestep whereas the other one could estimate the field water cut at a particular timestep.

For both proxies, the input variables include the number of days at each timestep j, t_j; the harmonic mean of grid absolute permeability for each reservoir layer, $\overline{k}$; the standard deviation of grid absolute permeability for each reservoir layer, k_{SD}; the permeabilities of perforated grid blocks (injectors), $k_{injector}$; the permeabilities of perforated grid blocks (producers), $k_{producer}$; the field water injection rate (control vector); the output at the previous timestep, y_{j-1}. The mathematical formulation of the proxies[1] built in this paper in general can be expressed as Eq. (3). Besides that, the harmonic mean of permeability for each reservoir layer is presented as Eq. (4). Nonetheless, regarding the input variables of the permeabilities of perforated grid blocks (producers and injectors), they are case-dependent in this paper. This implies that we have applied different approaches of formulation to incorporate them as parts of the inputs relying upon the reservoir models investigated. It will be discussed in detail later. In this work, the data-driven models were represented as the ANNs. The topologies of the ANNs developed here will be divulged in the next section.

$$y_j = f\left(t_j, \overline{k}, k_{SD}, k_{injector}, k_{producer}, \mathbf{u}, y_{j-1}\right) \tag{3}$$

$$\overline{k} = \frac{\sum_{i=1}^{n} L_i}{\sum_{i=1}^{n} \frac{L_i}{k_i}} \tag{4}$$

where n is the number of grid blocks, L_i is the depth at the top of grid block i, and k_i is the grid absolute permeability.

About the development of the proxies, the first step is to generate the spatiotemporal database. To develop the database, we apply three different sampling techniques, which are Latin Hypercube sampling (LHS), Sobol Sequence sampling (SSS), and Hammersley Sequence sampling (HSS) to generate 60 sets of samples of control rates (each set consists of 20 injection rates which corresponds to one injection scenario). Respectively, peruse McKay et al. (1979), Sobol' (1967), and Hammersley and Handscomb (1964) for more information about LHS, SSS, and HSS. Each sampling method is implemented to, respectively, generate 20 sets of samples. Thereafter, each set would be fed into the reservoir simulator to yield the reservoir responses. This denotes that 60 reservoir simulations are run in total. After finishing the simulations, we extract the dynamic inputs and combined them with the static inputs to create the spatiotemporal database. It is of paramount importance to have the database normalized and arrange in a consistent format before it is supplied to the neural network for training. The fundamental ideas of the neural network training will be delineated later. Before the neural network training commences, the normalized database is partitioned into three different groups, namely training, validation, and testing, based on a ratio of 8:1:1. In this case, only the training data is used to build the data-driven models. However, after each epoch (iteration) of training, validation data would be simultaneously fed into the neural network to elude the issue of overfitting (Mohaghegh 2017a; Shahkarami and Mohaghegh 2020). A heathy training can be ensured by having the simultaneous decreasing trends of the training and validation errors as shown in Fig. 2. After the training is completed, the testing data would be used to evaluate the predictability of the models.

Upon the completion of these three stages, the data-driven proxies ought to undergo the blind validation before being practically employed. The data used in blind validation should not be part of the above-mentioned spatiotemporal database. Therefore, to conduct the blind validation, we utilize LHS, SSS, and HSS to generate other 80 sets of samples of control rates. Then, 80 reservoir simulations are run to produce the outputs of field liquid production rates and field water cut. These outputs are compared with the predicted outputs yielded by the proxy models. Only when the comparative study shows excellent results, we can safely infer that the proxy models can practically be employed. By having successfully established these two proxies, the field oil and water production rates required for the optimization purpose can be obtained. In this paper, we implemented PSO and GWO to optimize the well control. The information about the algorithms and the parameters used to carry out the optimization will be presented later. We did not only

[1] The permeability used here is the horizontal permeability. The data of the vertical permeability can also be included in the development of the proxies here. However, it has not been considered in this paper as the current formulation already yielded very good results.

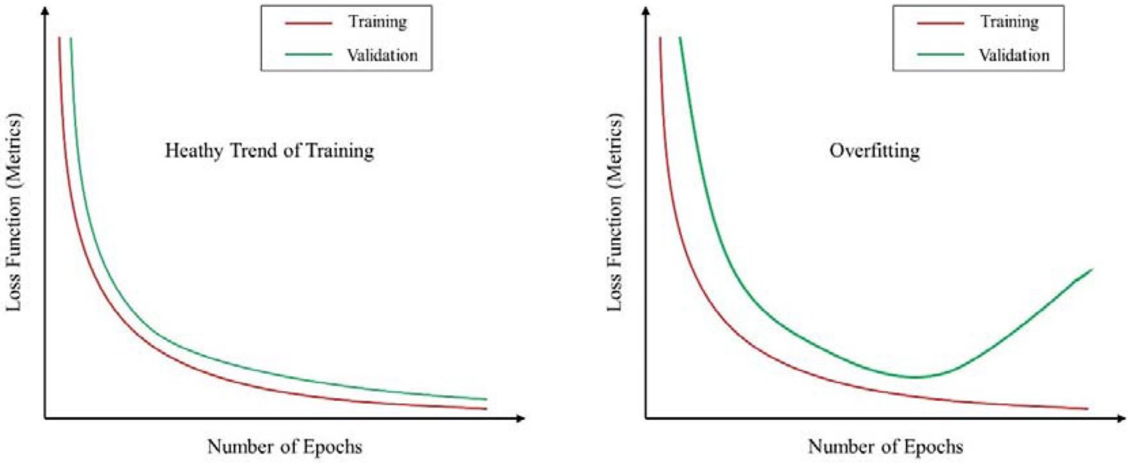


Fig. 2 Comparison between healthy trend of training and overfitting issue. Adapted from Shahkarami and Mohaghegh (2020)

couple these algorithms with the proxies developed here, but also applied them along with the numerical reservoir simulation. The optimal well controls resulted from the two approaches were then compared. Additionally, the proxy-optimized well controls were fed into the reservoir simulator to yield the results that could be used to further illustrate the robustness of the proxy models.

Artificial neural network

ANN is a famous ML method that is established based on the inspiration from the working process of the biological neural networks in human brains. ANN consists of a lot of computing elements which are termed as nodes or artificial neurons. It has been proven to be useful and effective in capturing and learning the sophisticated relationship between input and output data derived from any physical process as in traditional regression approaches. Examples of ANNs include feedforward neural network (FNN), convolutional neural network, recurrent neural network, radial basis function networks, and adaptive neuro-fuzzy inference system. Different types of activation functions can also be employed to develop an ANN and the common ones are the sigmoid function, the hyperbolic tangent, and the rectified linear unit (ReLU) function (Buduma and Locascio 2017).

In this paper, FNN with the ReLU function as its activation function was utilized. In general, FNN, also called multilayer perceptron (MLP), has three layers, e.g., the input layer, the hidden layer, and the output layer. To guarantee that the MLP can study the relationship between input and output data provided, it must undergo the training stage. During the training stage, the learning ability of the MLP is achieved by adjusting the sets of weights and biases to reduce the predefined loss function, including mean squared error (MSE) and mean absolute percentage error. In this paper, MSE was chosen as the loss function. Such optimization is generally conducted through the backpropagation (BP) approaches. These methods involve the application of different derivative-based algorithms, for instance, steepest descent gradient, the Levenberg–Marquardt algorithm, the Powell-Beale conjugate gradient, and Adam. In this work, Adam was applied as the training algorithm. For the details of Adam, refer to this literature (Kingma and Ba 2015). The relevant parameters used for the training of all the neural network proxies in this paper are tabulated in Table 1.

Prior to entering the MLP, the data have to be normalized to improve the training performance of the MLP as recommended in Hemmati-Sarapardeh et al. (2020). In this paper, we used Eq. (5) to normalize the data between 0 and 1. After normalization of data, the forward propagation of the input data will happen to compute the outputs. The resulting output data will thereafter be compared with the actual output data to determine the errors. After this, the errors are propagated back through the MLP to iteratively tune the weights and biases to reach the optimal point. The architecture of an arbitrary FNN is demonstrated in Fig. 3 in which the red node acts as the bias node between the input and hidden layers whereas the node between the hidden and output layers is shown in green.

Table 1 Parameters used to conduct neural network training using Adam

Adam parameters	Values
Number of iterations (epochs)	2000
Learning rate	0.001
Exponential decay rates for the 1st moment estimates, β_1	0.9
Exponential decay rates for the 2nd moment estimates, β_2	0.999
Numerical stability constant, ε	10^{-7}

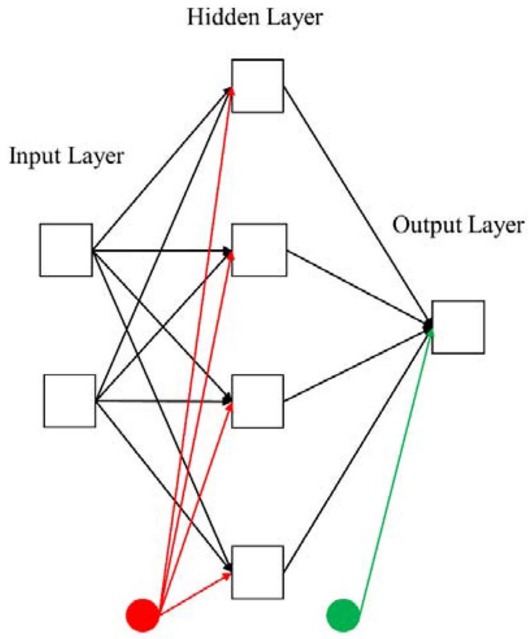


Fig. 3 The structure of a simple FNN model

$$x_{norm} = \frac{x_j - x_{min}}{x_{max} - x_{min}} \quad (5)$$

where x_{norm} represents the normalized data point, x_j refers to a data point, x_{min} corresponds to the data point with the lowest value, and x_{max} is the data point with the highest value. To assess the quality of the prediction done by the proxies, we used coefficient of determination, R^2 as the performance metrics. The respective formula is shown in Eq. (6).

$$R^2 = 1 - \frac{\sum_{j=1}^{N}\left(y_j^{real} - y_j^{pred}\right)^2}{\sum_{j=1}^{N}\left(y_j^{pred} - \overline{y^{real}}\right)^2} \quad (6)$$

where y_j^{real} means the actual data point, y_j^{pred} denotes the predicted data point, $\overline{y^{real}}$ refers to the mean of all the actual data points, and N is the total number of data points. As explained, we have built four neural network proxies in this paper, two for each of the reservoir models studied. The topologies of the neural proxies are presented in Table 2 for 2D reservoir model and Table 3 for 3D reservoir model. These architectures were determined via the trial and error approach.

Particle Swarm Optimization

The PSO algorithm is one of the most popular swarm-based metaheuristic algorithms that has been initiated by Kennedy and Eberhart (1995) through simulating the social habit of flying birds. Mathematically, this flock of birds is represented as a population of particles known as a swarm of particles. Each particle indicates a potential position (solution) in a search space and it is updated iteratively according to its position and velocity at previous iteration step. The motions of the particles are regulated by their own most optimal position (the local best position) and their most optimal position in the entire swarm (the global best position). After some iterations, the convergence of the particles in the swarm to an optimal point (the best solution) will occur. The position and velocity of the j^{th} particle in a k dimensional space at step t are formulated as follows:

Table 2 The architecture of ANN for 2D Reservoir Model

Layers	Field liquid production rate		Field water cut	
	Number of layers	Number of nodes	Number of layers	Number of nodes
Input	1	23	1	23
Hidden	1	100	2	50
Output	1	1	1	1

Table 3 The architecture of ANN for 3D Reservoir Model

Layers	Field liquid production rate		Field water cut	
	Number of layers	Number of nodes	Number of layers	Number of nodes
Input	1	29	1	29
Hidden	1	100	2	50
Output	1	1	1	1

$$x_{j,t} = \{x_{j1,t}, x_{j2,t}, x_{j3,t}, \ldots, x_{jk,t}\} \quad (7)$$

$$v_{j,t} = \{v_{j1,t}, v_{j2,t}, v_{j3,t}, \ldots, v_{jk,t}\}. \quad (8)$$

Thereafter, the velocity of each particle at next step t + 1 is updated based on Eq. (9) and the position of a particle at the next iteration t + 1 is updated by using Eq. (10).

$$\mathbf{v_{jk,t+1}} = \omega \mathbf{v_{jk,t}} + c_1 r_1 \left(\mathbf{pbest_{jk,t}} - x_{jk,t}\right) + c_2 r_2 \left(\mathbf{gbest_{k,t}} - \mathbf{x_{jk,t}}\right) \quad (9)$$

$$x_{jk,t+1} = x_{jk,t} + v_{jk,t+1} \quad (10)$$

where $v_{jk,t}$ and $x_{jk,t}$ indicate the velocity of the jth particle at step t and its corresponding position in the kth dimension quantity, respectively. Apart from this, $pbest_{jk,t}$ refers to the kth dimension quantity of the individual j at the local best position at iteration t. $gbest_{k,t}$ is the kth dimension quantity of the swarm at the global best position at iteration t. c_1 and c_2, respectively, denote the cognitive and social learning factors. ω is known as the inertial weight which was introduced by Shi and Eberhart (1998) to enhance the convergence condition. r_1 and r_2 are randomly retrieved between 0 and 1. In terms of the minimization problem, a cost function f (to be minimized) is defined. Then, to find out the local best solution at t + 1, Eq. (11) is used. To determine the global optimal solution at t + 1, Eq. (12) is applied.

$$\mathrm{pbest}_{jk,t+1} = \begin{cases} \mathrm{pbest}_{jk,t}, \text{if } f\left(\mathrm{pbest}_{jk,t}\right) \le f(x_{jk,t+1}) \\ x_{jk,t+1}, \text{otherwise} \end{cases} \tag{11}$$

$$\mathrm{gbest}_{k,\,t+1} = \min\left[f\left(\mathrm{pbest}_{jk,\,t+1}\right)\right]. \tag{12}$$

The procedure described above is repeated until the stopping condition is satisfied. During the optimization process, 15 particle swarms were initially generated, 100 iterations were run, and the values of ω, c_1, and c_2 were, respectively, set to be 0.8, 1.1, and 1.1.

Grey wolf optimization

The GWO is another well-known metaheuristic algorithm that was established by Mirjalili et al. (2014). This algorithm was developed in accordance with the natural inspiration derived from the social hierarchy of leadership and hunting style of grey wolves (Mirjalili et al. 2014). Pertaining to the paradigm of this algorithm, it is essential to recognize that the population of grey wolves is divided into four different classes, such as alpha (α), beta (β), delta (δ), and omega (ω). Based upon the social hierarchy, ω wolves are the lowest among others and they are preceded by δ, β, and α. To mathematize the mechanism of GWO, a population of wolves is expressed as a set of random solutions. The fitness value of this set of solutions is then calculated and assessed by applying a predefined objective function (Xu et al. 2020). After that, the wolves' populations are divided into the four previously stated classes based on the computed fitness value. When the optimization takes place, the three most optimal wolves: α, β, and δ, would eventually guide the other ω wolves toward the prey that acts as the global solution in the search space. This procedure is carried out via the iterative update of the positions of the wolves as shown below:

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_p}(t) - \vec{X}(t)\right| \tag{13}$$

$$\vec{X}(t+1) = \left|\overrightarrow{X_p}(t) - \vec{A}.\vec{D}\right| \tag{14}$$

$$\vec{A} = 2\vec{a}.\overrightarrow{r_1} - \vec{a} \tag{15}$$

$$\vec{C} = 2\overrightarrow{r_2} \tag{16}$$

where t means the current iteration step, $\vec{X}$ implies the position of a grey wolf, $\overrightarrow{X_p}$ is the position of the prey, $\vec{a}$ is normally lowered from 2 to 0. Also, $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ are the random vectors between 0 and 1. In GWO, the position of the prey (the global optimal solution) is not exactly known. Hence, it is assumed that the positions of α, β, and δ are considered as the optima. Then, the other ω wolves re-calibrate their positions with respect to those of α, β, and δ as follows:

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1}.\overrightarrow{X_\alpha}(t) - \vec{X}(t)\right| \tag{17}$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{C_2}.\overrightarrow{X_\beta}(t) - \vec{X}(t)\right| \tag{18}$$

$$\overrightarrow{D_\delta} = \left|\overrightarrow{C_3}.\overrightarrow{X_\delta}(t) - \vec{X}(t)\right| \tag{19}$$

where $\overrightarrow{X_\alpha}(t)$ corresponds to the position of α wolves at step t, $\overrightarrow{X_\beta}(t)$ is the position of β wolves at step t, and $\overrightarrow{X_\delta}(t)$ represents the position vector of δ wolves at iteration t. α, β, and δ wolves will then update their positions at iteration t + 1 based on Eqs. (20), (21), and (22). The position of the solution at step t + 1 is thereafter determined based upon Eq. (23).

$$\overrightarrow{X_1} = \left|\overrightarrow{X_\alpha}(t) - \overrightarrow{A_1}.\overrightarrow{D_\alpha}\right| \tag{20}$$

$$\overrightarrow{X_2} = \left|\overrightarrow{X_\beta}(t) - \overrightarrow{A_2}.\overrightarrow{D_\beta}\right| \tag{21}$$

$$\overrightarrow{X_3} = \left|\overrightarrow{X_\delta}(t) - \overrightarrow{A_3}.\overrightarrow{D_\delta}\right| \tag{22}$$

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3}. \tag{23}$$

These steps are repeated until the stopping condition is met. During the optimization process, 15 populations of grey wolves were initially generated, and 100 iterations were run.

Results

Case study 1: 2D reservoir model

We first illustrate the development of a data-driven proxy model of a 2D heterogeneous and 2-phase (water and black oil) reservoir model. The heterogeneity only applies to the permeability in this case study. Besides that, the horizontal permeabilities in both x and y directions are assumed to be the same whereas the vertical permeability is set to be 10 times smaller. Also, homogeneity applies to porosity and it is assumed to be 0.4. Regarding the size of the grid blocks,

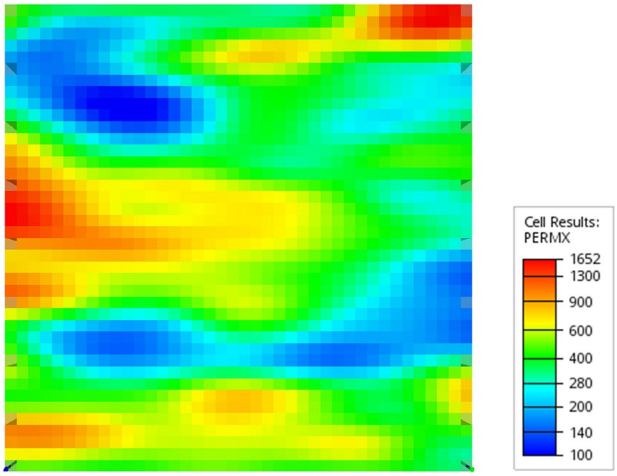


Fig. 4 The overview of the 2D reservoir model. The color bar indicates the values of horizontal permeability in x-direction in the units of millidarcy (mD)

it is 10 m × 10 m × 10 m and the total number of grid blocks is 40 × 40 × 1. Therefore, the dimension of the whole model is 400 m × 400 m × 10 m. Its top is at the depth of 1500 m. Pertaining to the configuration of well, there are only two wells being drilled, namely one horizontal injector and one horizontal producer. The injector is drilled at the left edge whereas the producer is placed at the right edge. The 2D reservoir model is illustrated in Fig. 4. As briefly discussed, the performance of the injector is controlled by the rate within the range of 40 m^3/day and 100 m^3/day whereas the producer is controlled by the BHP with the lower limit of 180 bar. Regarding the perforation in the x-direction, the injector is perforated at the 1^{st} grid block whereas the producer is perforated at the 40th grid block. However, for y-direction, both wells are completed at 1st, 5th, …, 35th, 40th grid blocks. Permeabilities of these grid blocks (constituting 18 variables in total) are directly retrieved and used as the input parameter for neural network training. The numerical simulation is performed using ECLIPSE 100 software Schlumberger.

After running the required numerical reservoir simulations and extracting the input and output data, the neural network training was correspondingly performed on the data-driven proxies of field liquid rate and field water cut by using the specification listed in Table 2. Based on Eq. (3) and Table 2, there are 23 input parameters applied to train the proxies. The performances of training, validation, and testing of both proxies are evaluated by using the coefficient of determination, R^2, and shown in Table 4. Besides that, for the blind validation phase, the proximity of the actual and targeted outputs is assessed by applying 80 injection schedules. Thereafter, the mean of the respective coefficient of determination is calculated for each proxy and tabulated in Table 5. For illustration purpose, only the result for a randomly selected injection schedule of blind validation (out of 80) is demonstrated for each sampling method. In this case, the comparison between the actual and the predicted field liquid production rates (also field water cuts) is, respectively, plotted as shown in Fig. 5 for LHS, in Fig. 6 for SSS, and in Fig. 7 for HSS. According to these results, it is inferred that these proxy models are ready for practical application.

Table 4 R^2 of training, validation, and testing results of the data-driven proxies

Dataset	Field liquid production rate	Field water cut
Training	0.9999	0.9999
Validation	0.9999	0.9999
Testing	0.9999	0.9999

Table 5 Mean R^2 of blind validation of proxies based on different sampling techniques

Sampling methods	Field liquid production rate	Field water cut
LHS	0.9999	0.9995
SSS	0.9999	0.9995
HSS	0.9999	0.9995

In this aspect, we defined the economic parameters as depicted in Table 6 to be used in the optimization process. As mentioned earlier, PSO and GWO would be employed to conduct the optimization of NPV. The optimized controls of field water injection rates are, respectively, illustrated in Fig. 8 for PSO and in Fig. 9 for GWO. Pertaining to this, the resulted optimal NPV of three different scenarios is demonstrated in Table 7 in which Scenario 1 represents the optimization by only using the reservoir simulator (NPV_{sim}), Scenario 2 denotes the optimal NPV obtained by feeding the proxy-optimized control into the simulator ($NPV_{sim\text{-}proxy}$), and Scenario 3 means the optimization by only using the proxies (NPV_{proxy}). Pertaining to the optimal NPVs yielded from three different scenarios, it can be noted that the data-driven proxies have in general overestimated the optimal NPV for both algorithms. However, the absolute percentage error between NPV_{proxy} and either NPV_{sim} or $NPV_{sim\text{-}proxy}$ is miniscule.

For PSO, the absolute percentage error between NPV_{sim} and NPV_{proxy} is around 0.14%. This shows that when the data-driven proxies are coupled with PSO, they can yield reasonable results to approximate the NPV calculated with the results from simulator. Furthermore, to understand whether the proxies produce accurate results for the calculation of NPV, the absolute percentage error between NPV_{sim} and $NPV_{sim\text{-}proxy}$ is found out to be about 0.13%. Additionally, the absolute percentage error between $NPV_{sim\text{-}proxy}$ and

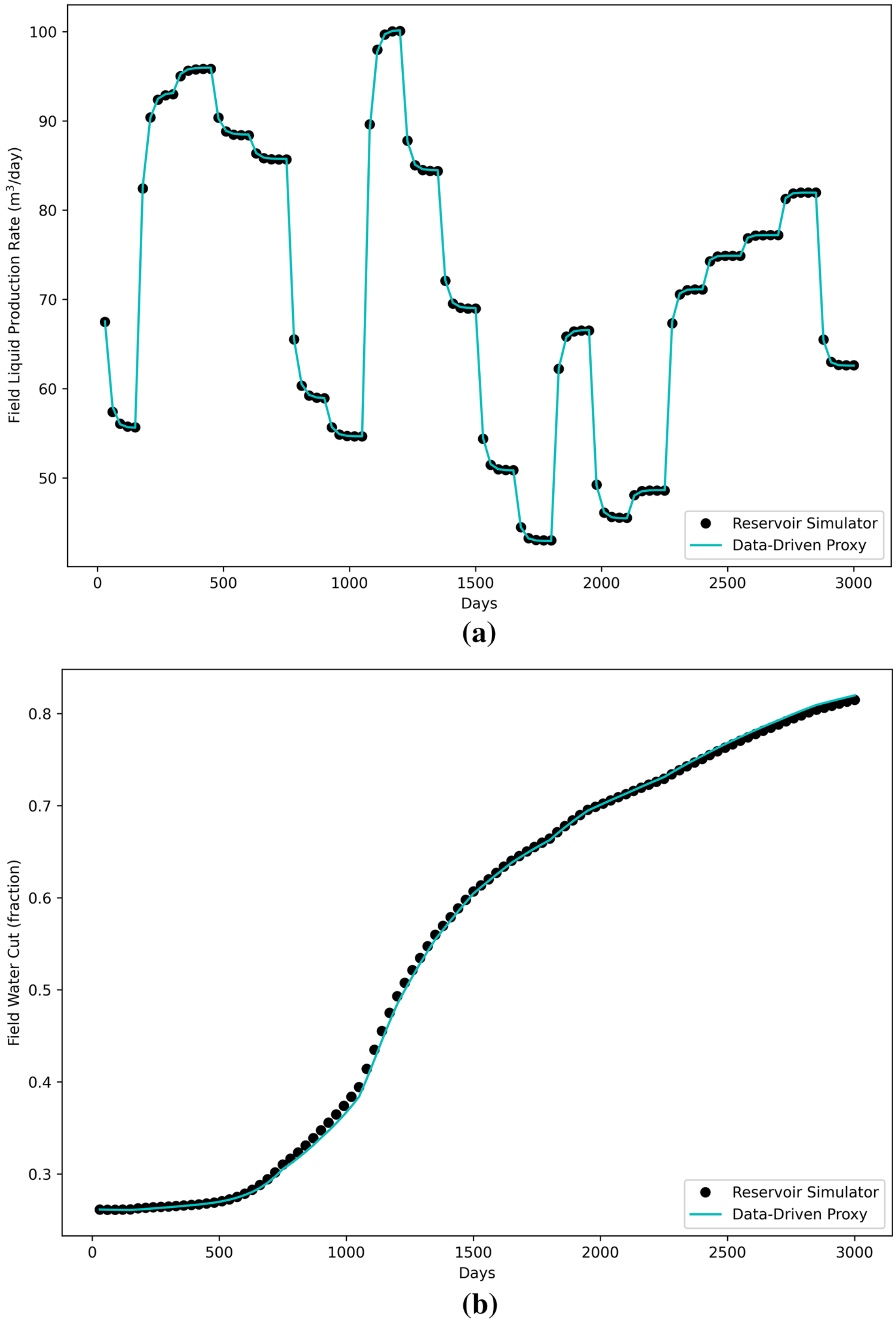
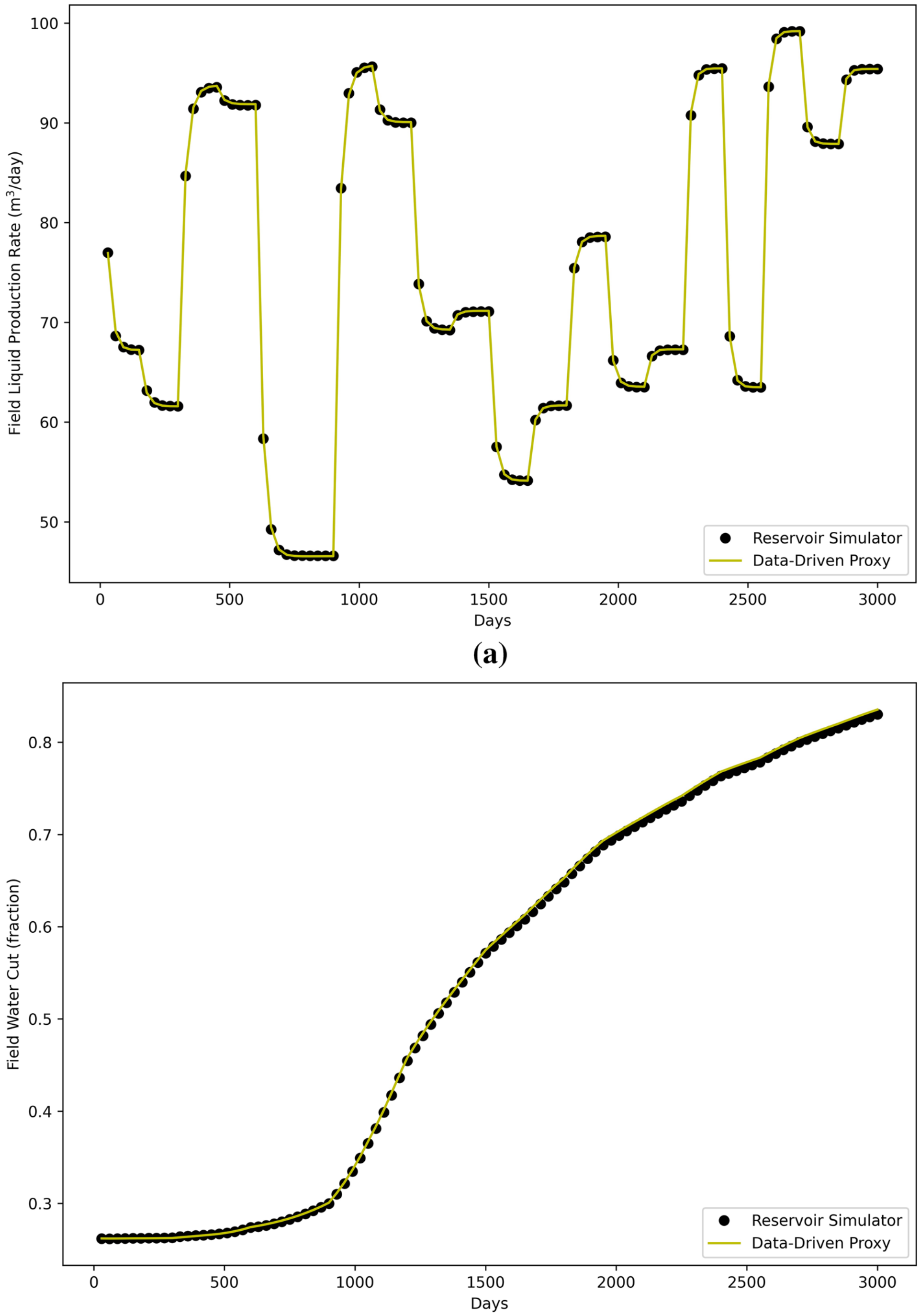


Fig. 5 Results of blind validation of data-driven proxies of LHS sample set 11 (out of 80). **a** Field liquid production rate. **b** Field water cut

NPV_{proxy} is 0.27%. This proves the reliability of the developed proxies. Thereafter, for GWO, the absolute percentage error between NPV_{sim} and NPV_{proxy} is 0.34% while it is 0.40% between NPV_{sim} and $NPV_{sim\text{-}proxy}$. Also, the absolute percentage error between $NPV_{sim\text{-}proxy}$ and NPV_{proxy} is 0.74%. Despite the higher accuracy of optimization results portrayed by PSO, it can be noted that GWO generally performs better than PSO in the context of optimization in this case study. To further demonstrate the high proximity of the data-driven models, the plots of PSO-optimized and GWO-optimized field water production rates of Scenario 2 against Scenario 3 are correspondingly illustrated in Figs. 10 and 11. R^2 obtained for Fig. 10 is 0.9998 whereas that of Fig. 11 is 0.9989. The similar plots for field oil production rates are presented for PSO in Fig. 12 and GWO in Fig. 13 Then, the values of R^2 calculated for Figs. 12 and 13 are 0.9999.



(a)

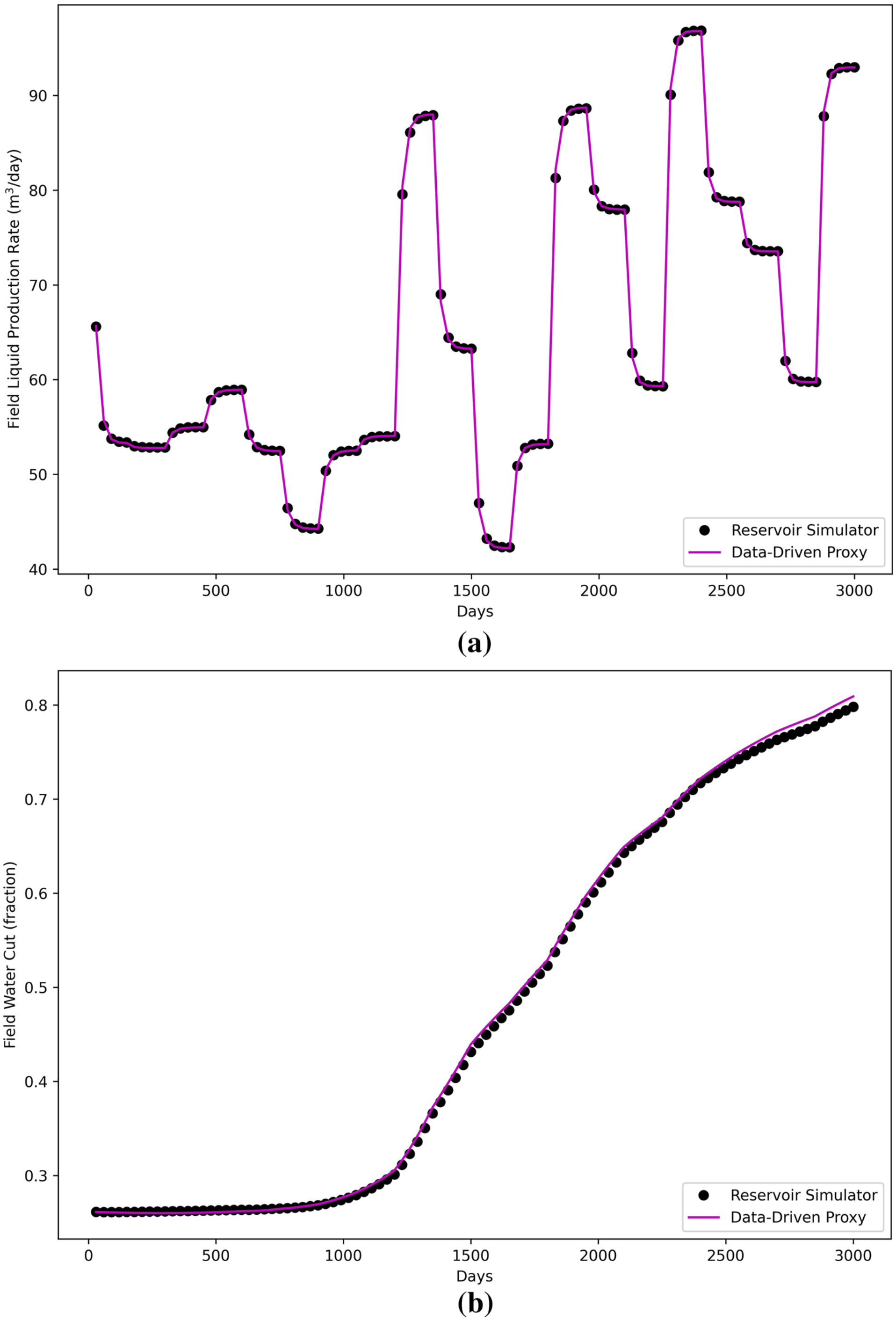
(b)

Fig. 6 Results of blind validation of data-driven proxies of SSS sample set 32 (out of 80). **a** Field liquid production rate. **b** Field water cut

Case study 2: 3D reservoir model (Egg Model)

To further demonstrate the methodology of proxy modeling proposed in this paper, we present the use of a more sophisticated reservoir model as another case study. The model was initiated by Jansen et al. (2014) and termed as "Egg Model." It was employed as case study in several papers (Van Essen et al. 2009; Hong et al. 2017). In general, it is considered as a channelized depositional model where the heterogeneity only pertains to permeability. However, porosity is homogeneous and set to be 0.2. Besides that, the initial water saturation is 0.1 and it applies to all grid blocks. The size of the grid blocks is 8 m × 8 m × 4 m and the total number of grid blocks is 60 × 60 × 7. However, the

Fig. 7 Results of blind validation of data-driven proxies of HSS sample set 69 (out of 80). **a** Field liquid production rate. **b** Field water cut



total number of active grid blocks is 18533. The horizontal permeability map of Egg Model is illustrated in Fig. 14. The details of the geological properties of this model can be found in Jansen et al. (2014). Besides that, the model comprises eight vertical injectors and four vertical producers. The only modification done on the Egg Model to fulfill the need of the analysis here is changing the control of injectors. Since there are eight injectors and each of them is controlled by the rate within the range of 40 m^3/day and 100 m^3/day, the field injection rates are altered between 320 m^3/day and 800 m^3/day. Regarding the four producers, each of them is controlled by the BHP with the lower limit of 395 bar.

About the completion, all the wells are perforated in seven layers. If we apply the formulation presented in the case study of 2D reservoir model to include the grid block permeability as the input variables, then this will result in 84

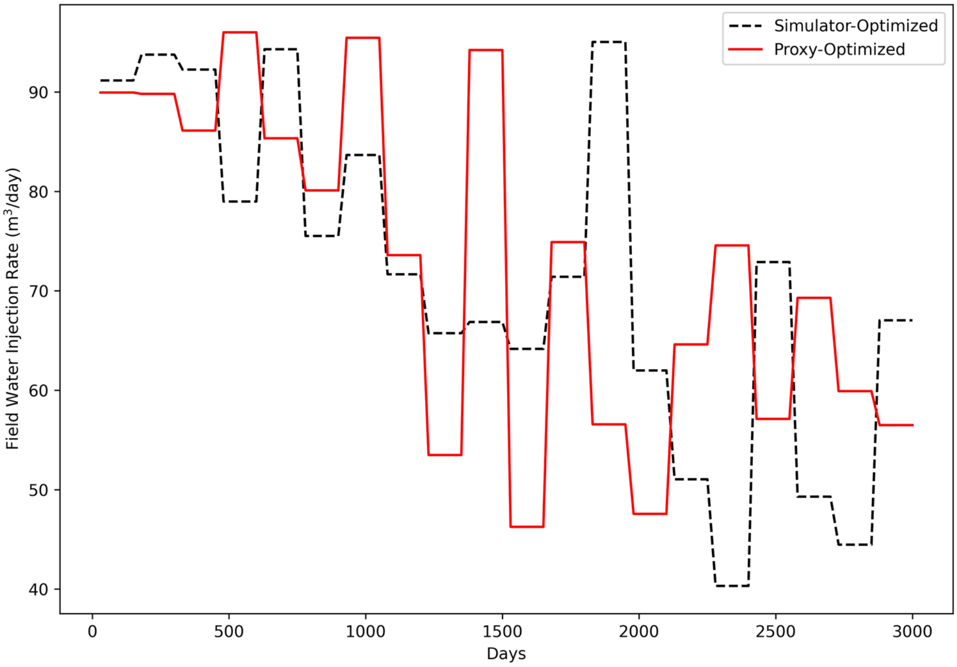


Fig. 8 Optimized control rates by using simulator and data-driven proxies with the implementation of PSO

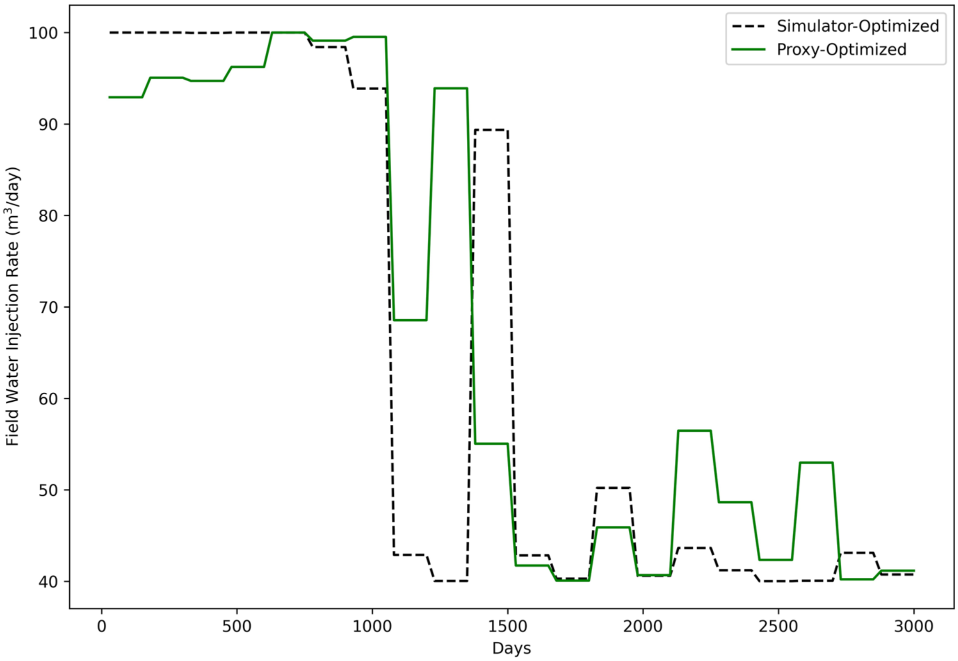


Fig. 9 Optimized control rates by using simulator and data-driven proxies with the implementation of GWO

Table 6 Economics parameters used for NPV calculation

Variables	Values	Units
Oil price, P_o	314.50	USD/m^3
Cost of produced water, C_w	37.50	USD/m^3
Cost of injected water, C_{wi}	37.50	USD/m^3
Discount rate	0.10	per year

variables of $k_{injector}$ and $k_{producer}$. Thus, for practical purpose of eluding the curse of dimensionality, we determined the arithmetic mean of the permeability of the perforated grid blocks for each well. This could reduce the number of permeability variables from 84 to 12. Given there are 7 layers in this Egg Model, there will be a total of 14 variables of $\overline{k}$ and k_{SD}. According to Eq. (3), there are 29 input variables to be included to train the neural network. By employing the same

Table 7 Optimization results of three scenarios for PSO and GWO

Optimization Algorithm	Scenario 1		Scenario 2		Scenario 3	
	GWO	PSO	GWO	PSO	GWO	PSO
$NPV_{optimal}$ (million USD)	16.52	16.26	16.46	16.24	16.58	16.29

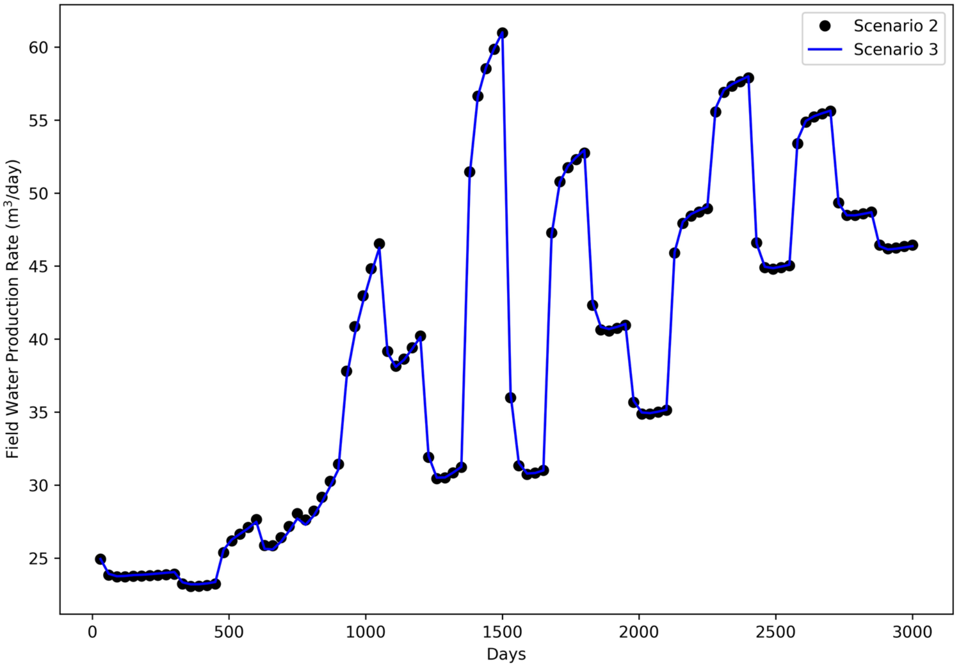


Fig. 10 Plot of PSO-optimized field water production rates, comparison between Scenarios 2 and 3

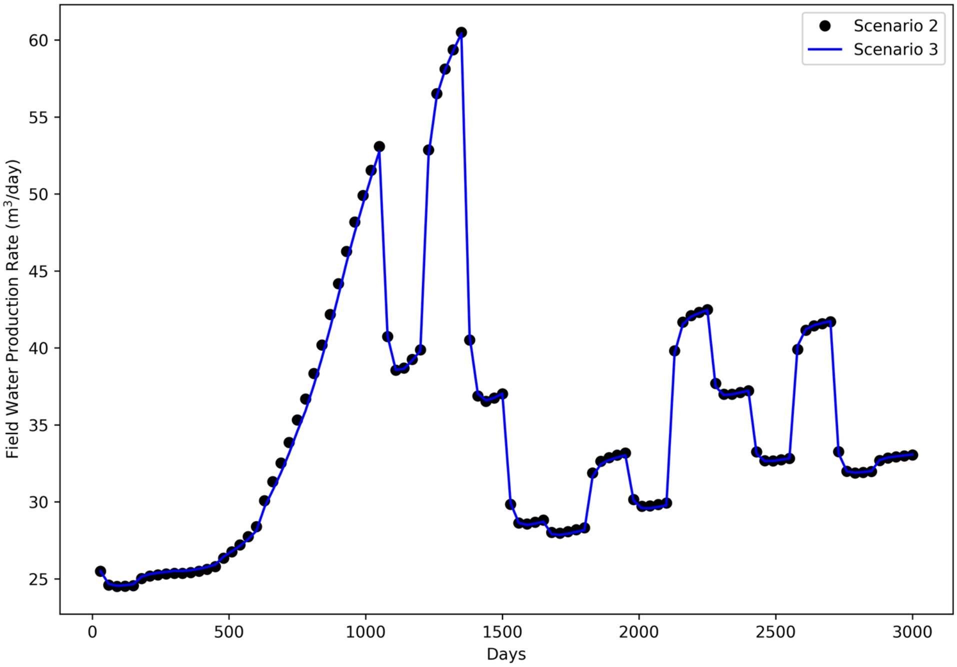


Fig. 11 Plot of GWO-optimized field water production rates, comparison between Scenarios 2 and 3

methodology as explained earlier and the specifications presented in Table 3, the neural network training is conducted. Performance metrics of training, validation, and testing of the proxies of the Egg Model are presented in Table 8. Also,

Fig. 12 Plot of PSO-optimized field oil production rates, comparison between Scenarios 2 and 3

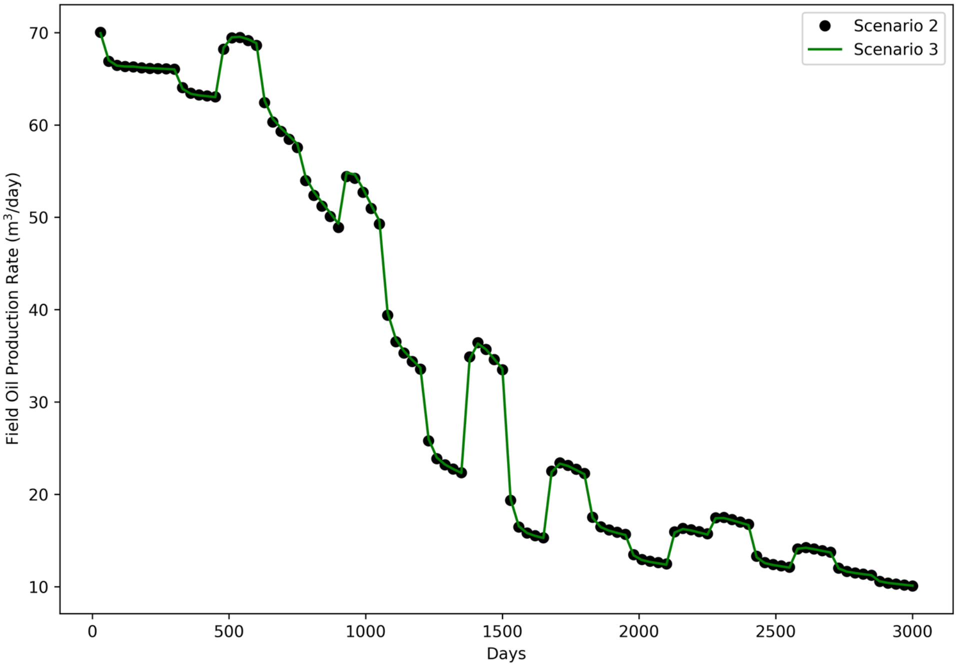
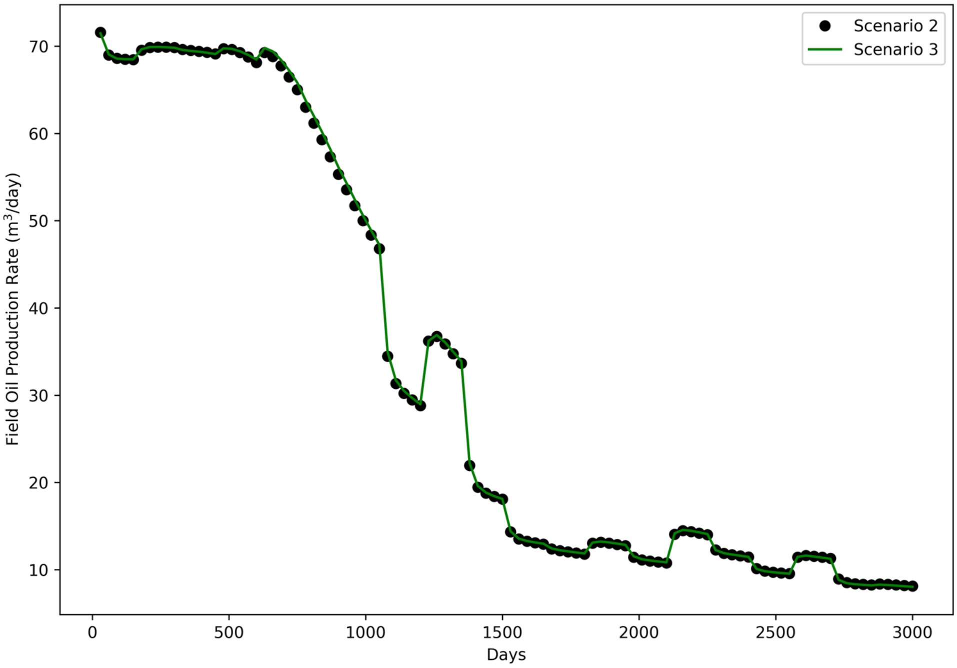


Fig. 13 Plot of GWO-optimized field oil production rates, comparison between Scenarios 2 and 3



the mean of the corresponding coefficient of determination is computed for each proxy and shown in Table 9. For illustration purpose, like Figs. 5, 6, and 7, the graphs of the comparison between the actual and the predicted field liquid production rates (also field water cut) are shown in Fig. 15 for LHS, in Fig. 16 for SSS, and in Fig. 17 for HSS. To perform the optimization of NPV, different economic parameters, as shown in Table 10, are used because using the parameters in Table 6 will result in a mathematically trivial solution in this case study. Table 11 illustrates the results of optimization of the three scenarios. The optimized controls of field water injection rates are shown in Fig. 18 for PSO and in Fig. 19 for GWO. In this case study, the overestimation of NPV by the data-driven proxies for both algorithms is also noticed. Nevertheless, this overestimation is practically infinitesimal in which the application of these proxies is still feasible.

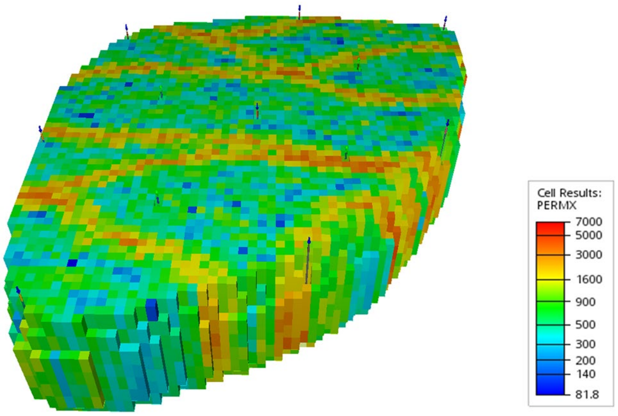


Fig. 14 The overview of the 3D Egg Model. The color bar indicates the values of horizontal permeability in x-direction in the units of millidarcy (mD)

Table 8 R^2 of training, validation, and testing results of the data-driven proxies

Dataset	Field liquid production rate	Field water cut
Training	0.9999	0.9999
Validation	0.9999	0.9999
Testing	0.9999	0.9999

Table 9 Mean R^2 of blind validation of proxies based on different sampling techniques

Sampling methods	Field liquid production rate	Field water cut
LHS	0.9999	0.9992
SSS	0.9999	0.9990
HSS	0.9999	0.9990

During the optimization using PSO, the absolute percentage error between NPV_{sim} and NPV_{proxy} is approximately 2.43%. Given the higher complexity of the Egg Model, the results produced are within satisfactory level of accuracy to approximate the NPV_{sim}. Also, the absolute percentage error between NPV_{sim} and $NPV_{sim\text{-}proxy}$, which is determined to be about 0.68%, portrays a higher confidence in the usefulness of the results obtained by the proxies. Besides that, the absolute percentage error between $NPV_{sim\text{-}proxy}$ and NPV_{proxy} is 3.13%. It can be inferred that the proxies of Egg Model are deemed reliable as well. For the case of GWO, the absolute percentage error between NPV_{sim} and NPV_{proxy} is 2.72% while it is 1.81% between NPV_{sim} and $NPV_{sim\text{-}proxy}$. Also, the absolute percentage error between $NPV_{sim\text{-}proxy}$ and NPV_{proxy} is 4.62%. In this case study, GWO can achieve a higher accuracy of optimization than PSO. In addition, GWO generally outperforms PSO in terms of optimization, except for Scenario 2. The high proximity of these data-driven models is also captured through the demonstration of the plots of PSO-optimized field water production rates of Scenario 2 against Scenario 3 in Fig. 20 and those for GWO in Fig. 21. R^2 computed for Fig. 20 is 0.9986 whereas that of Fig. 21 is 0.9978. The similar plots for field oil production rates are also shown for PSO in Fig. 22 and GWO in Fig. 23. Then, R^2 determined for Fig. 22 is 0.9991 whereas that of Fig. 23 is 0.9981.

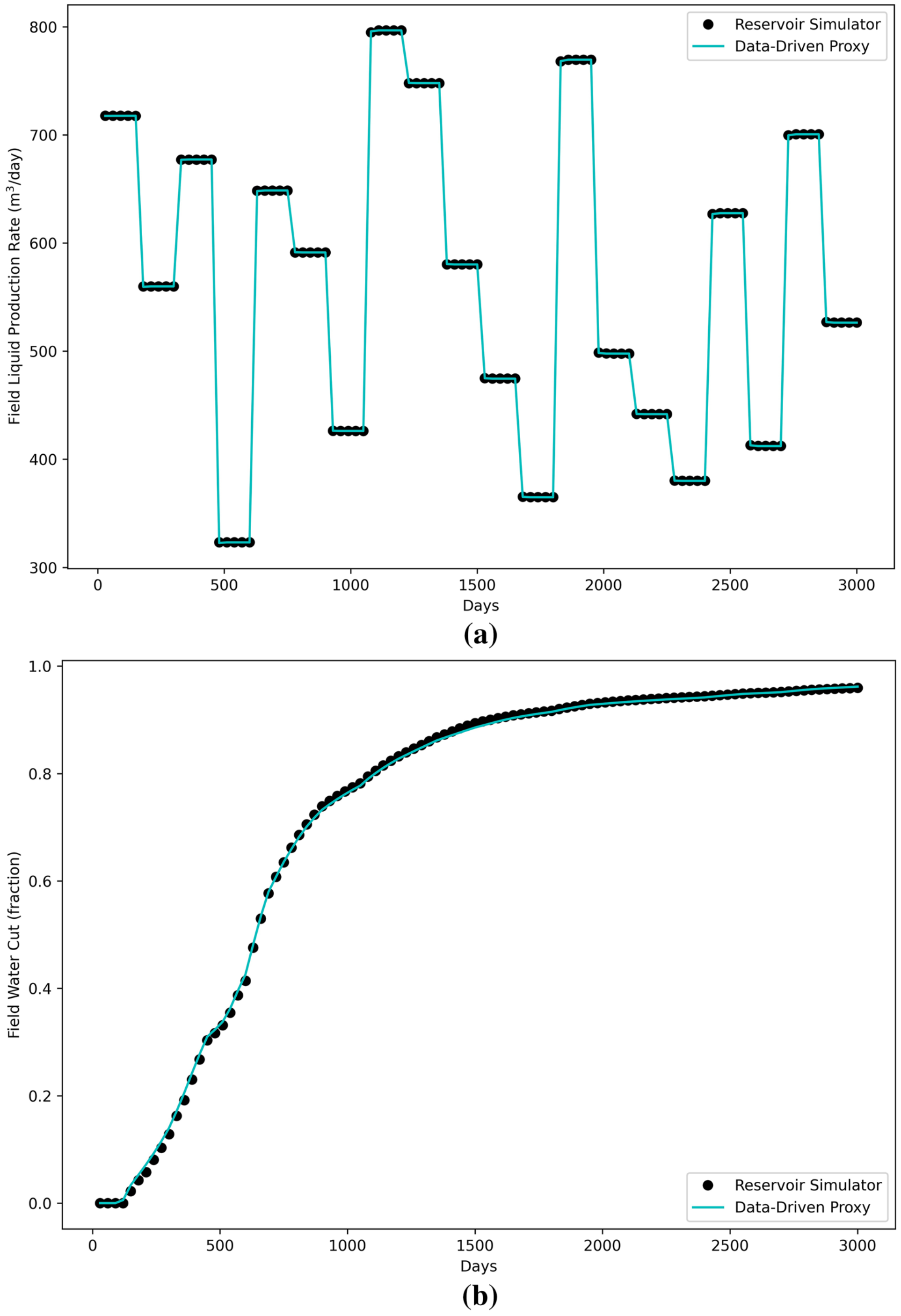
To demonstrate the accuracy and robustness of the approaches proposed in this study, the plots of field water (and oil) production rates under Scenario 2 against those under an unoptimized scheme are provided. To elude any confusion, the optimized rates used to produce these plots are derived from simulator. The unoptimized scheme, which is also known as "base case," comprises a constant field injection rate of 560 m^3/day over the whole production period. The corresponding NPV of base case is determined to be 152.57 million USD. Refer to Table 11 for the NPVs of the optimized cases (Scenario 2). Figure 24 illustrates the plot of PSO-optimized field water production rates (Scenario 2) against that of base case whereas Fig. 25 portrays the similar plot for field oil production rates. For GWO method, the plots for field water and oil production rates are, respectively, shown in Figs. 26 and 27. According to these four figures, we can fairly deduce that the optimization schemes have been performed practically well in the case study of Egg Model.

Discussion

About the results of NPV optimization in Tables 7 and 11, it is observed that in all three scenarios for both case studies, GWO reached a higher optimal NPV than PSO, except for Scenario 2 in Egg Model. This shows that GWO generally outperforms PSO to yield better optimization results based upon the analysis conducted in this paper. Despite this, the underperformance of GWO in Scenario 2 of Egg Model can be due to the lack of efficiency in the sampling of data for the neural network training. This means that the data sampled might not be efficiently extensive to cover the solution space of optimization induced by GWO. Nonetheless, based on the results presented, the data-driven proxies are still able to practically serve their objective when coupled with GWO. The sampling strategy used in this paper is deemed straightforward and still has room for improvement. This domain is not emphasized much as it is not the focus of our study here. In this aspect, the efficient sampling algorithm initiated by Dige and Diwekar (2018) can be taken into account as future work to enhance the sampling strategy in this paper.

There are a few limitations about the data-driven proxies developed in this paper. One of them pertains to the

Fig. 15 Results of blind validation of data-driven proxies of LHS sample set 11 (out of 80). **a** Field liquid production rate. **b** Field water cut



(a)

(b)

applicability of the models. As discussed in several literatures (Mohaghegh 2011, 2017a, 2017b; Ng et al. 2021), the data-driven model is only relevant to the reservoir model being studied. This denotes that it cannot be implemented as the substitute for another reservoir. In addition, the developed proxy models are only able to capture the physics of the reservoir system that is represented by the spatiotemporal database. For instance, if the proxy is established for a reservoir model that is waterflooded, then it cannot be employed for the analysis of other enhanced oil recovery (EOR) methods, such as CO_2 injection and water-alternating-gas (WAG). The elimination of the control switch problem is considered as another limitation. This means that the reservoir simulation system is designed in a way that for the injectors, the

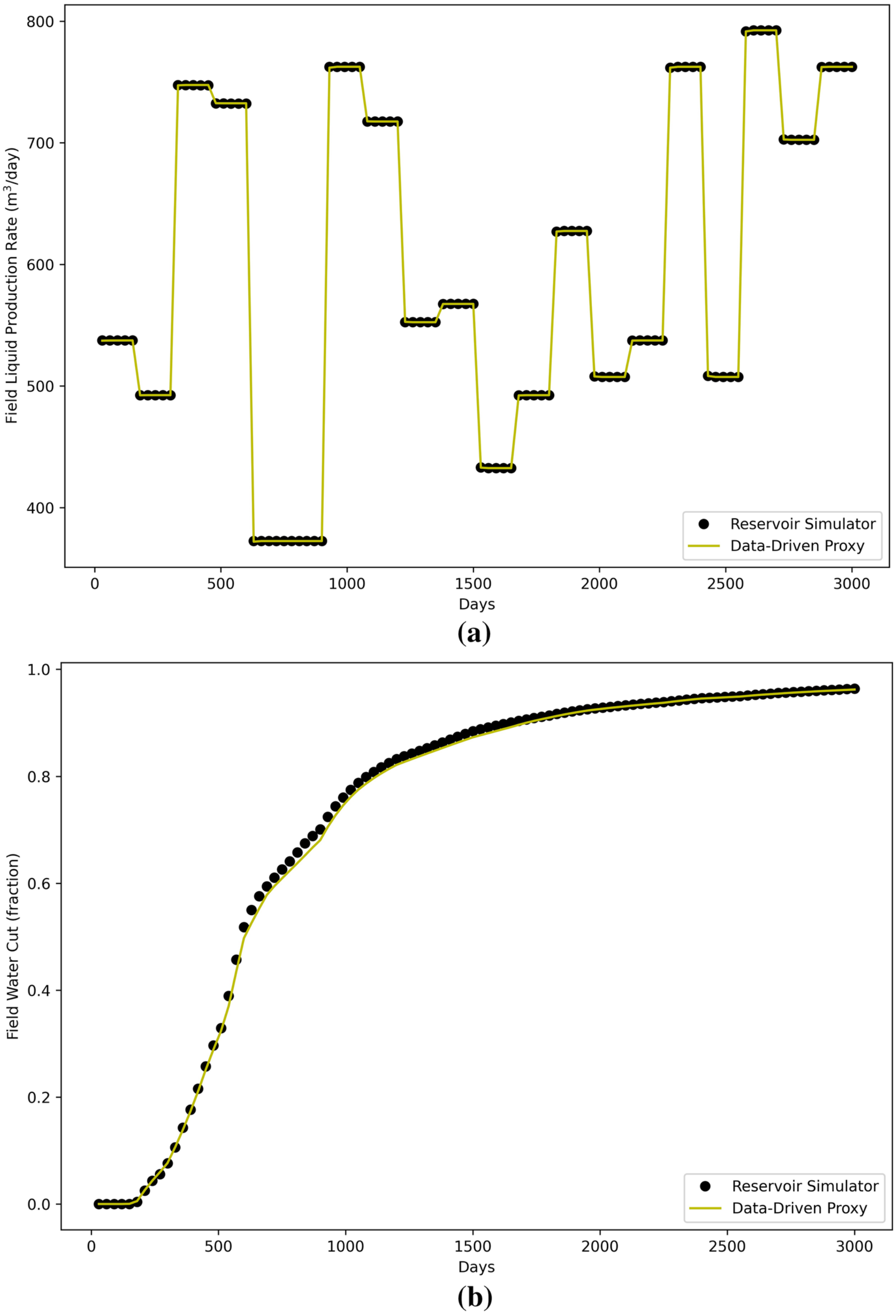


Fig. 16 Results of blind validation of data-driven proxies of SSS sample set 32 (out of 80). **a** Field liquid production rate. **b** Field water cut

control will not switch from injection rates to BHP during the optimization process. The similar condition also applies to the producers (but BHP is the control for producers). Moreover, for the case study of Egg Model, the well rate is determined by equally dividing the field rate by the number of wells. This implies that the optimization problem presented here is slightly simplified for illustration purpose.

The aspect of computational cost is the catalyst for the rapid development of the proxy models. As discussed earlier, NRS can induce high computational footprints especially when the reservoir model is geologically very sophisticated. Therefore, applying proxy models for further analysis is undeniably time saving. To further demonstrate this advantage, we compare the computation time required by

Fig. 17 Results of blind validation of data-driven proxies of HSS sample set 69 (out of 80). **a** Field liquid production rate. **b** Field water cut

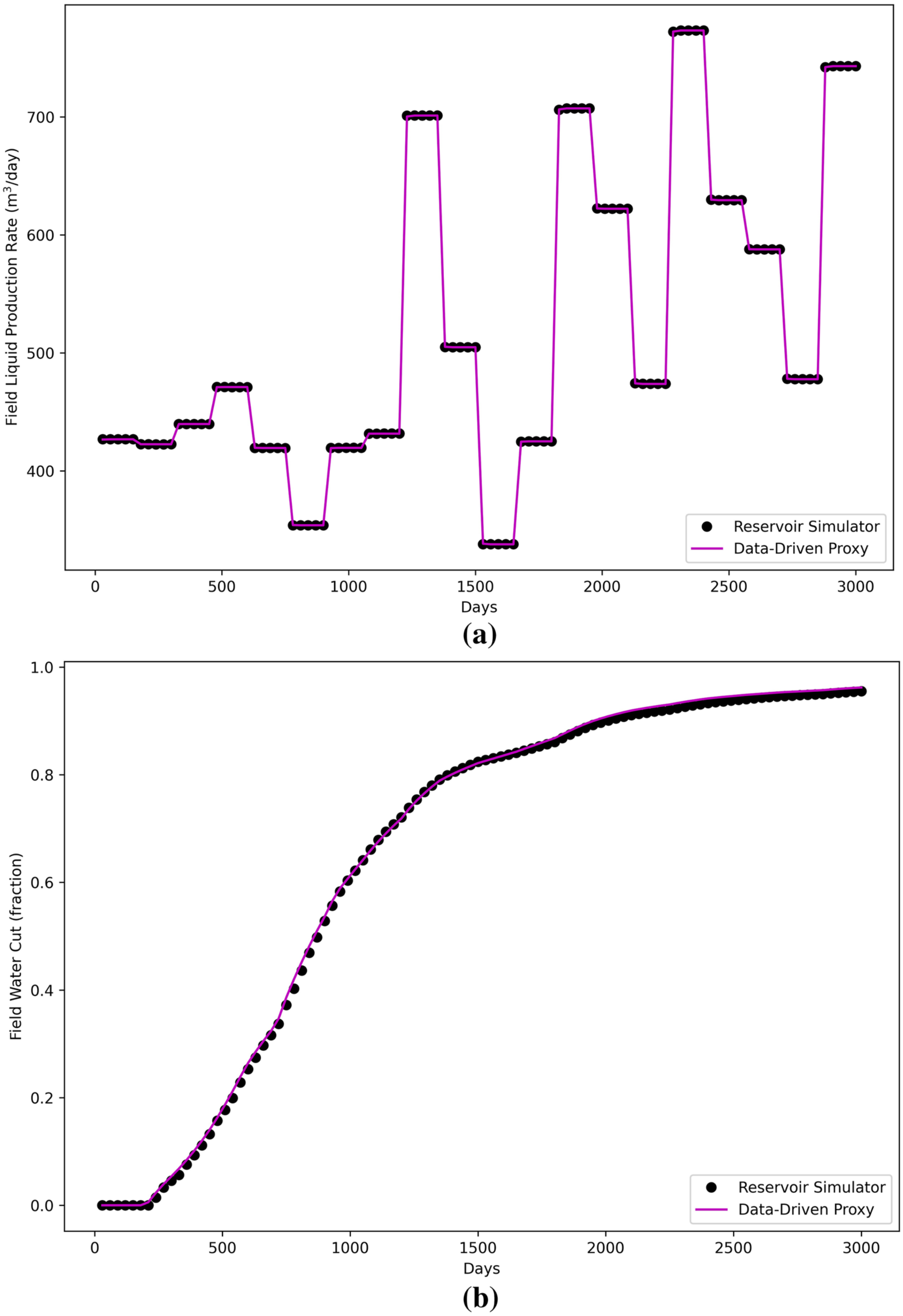


Table 10 Economics parameters used for NPV calculation

Parameters	Values	Units
Oil price, P_o	440.30	USD/m^3
Cost of produced water, C_w	12.58	USD/m^3
Cost of injected water, C_{wi}	12.58	USD/m^3
Discount rate	0.10	Per year

performing the optimization on both reservoir and proxy models. It was conducted on a PC which has the specification of Intel® Core™ i9-9900 CPU @3.10 GHz with 64.0 GB RAM. In this context, the time used by both PSO and GWO are very close. For the 2D reservoir model, the optimization took about 3 h whereas its respective proxies utilized about 1 h and 40 min to finish the optimization. Therefore, it is seen that the proxy models were able to save

Table 11 Optimization results of three scenarios for PSO and GWO

Optimization algorithm	Scenario 1		Scenario 2		Scenario 3	
	GWO	PSO	GWO	PSO	GWO	PSO
$\text{NPV}_{\text{optimal}}$ (million USD)	157.14	155.78	154.29	154.73	161.42	159.57

Fig. 18 Optimized control rates by using simulator and data-driven proxies with the implementation of PSO

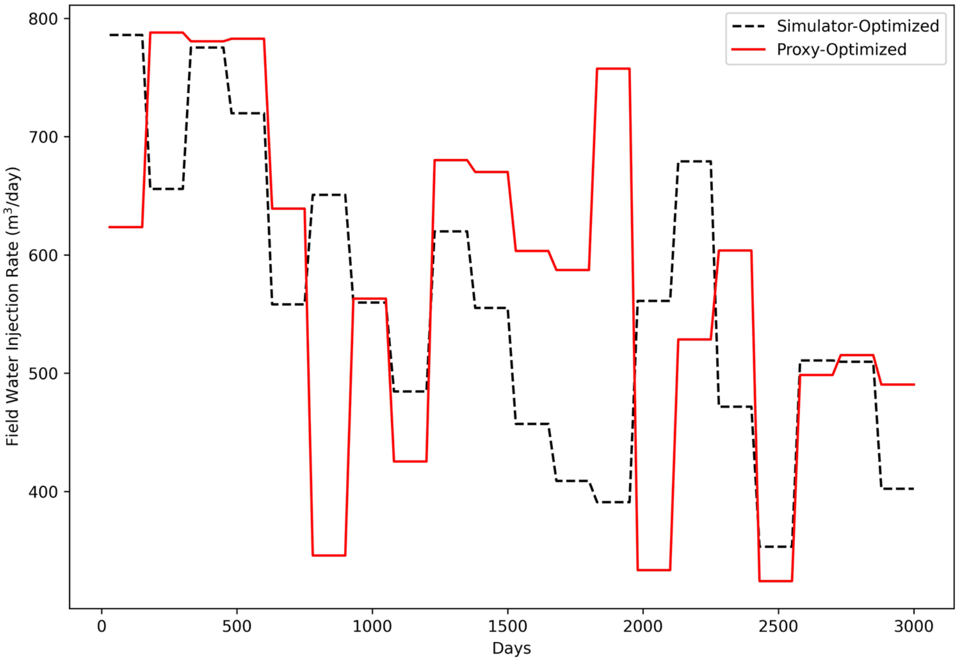
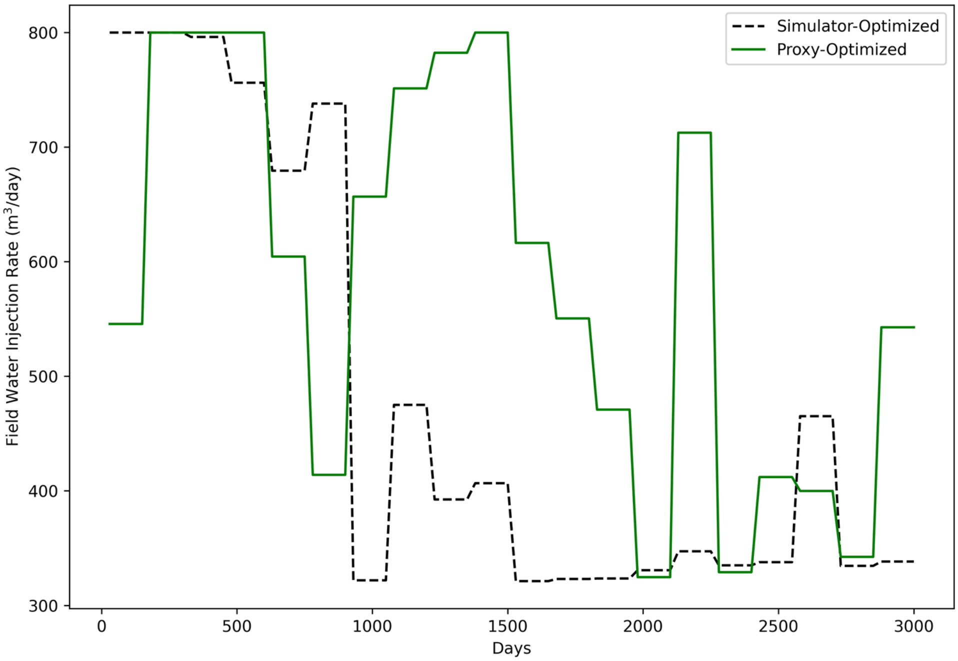


Fig. 19 Optimized control rates by using simulator and data-driven proxies with the implementation of GWO



about 50% of the whole computational time. More intriguingly, this advantage is more obvious for the case study of Egg Model. When applying the optimization algorithm on the Egg Model, it took about 13 h to run the optimization. However, the corresponding proxies only needed 2 h to do so. This illustrates that the data-driven proxies were 6 times

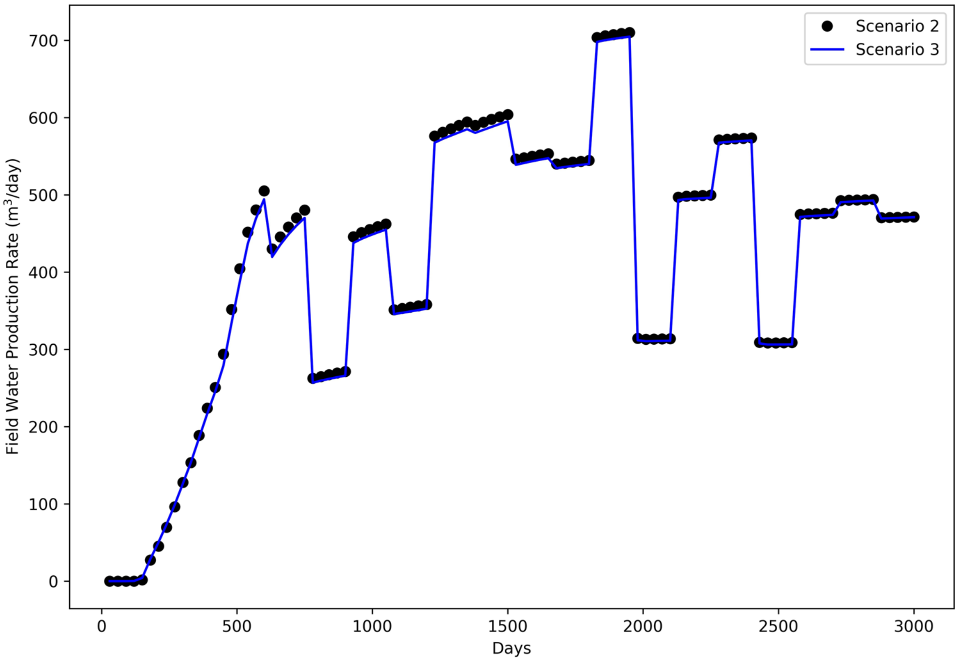


Fig. 20 Plot of PSO-optimized field water production rates, comparison between Scenarios 2 and 3

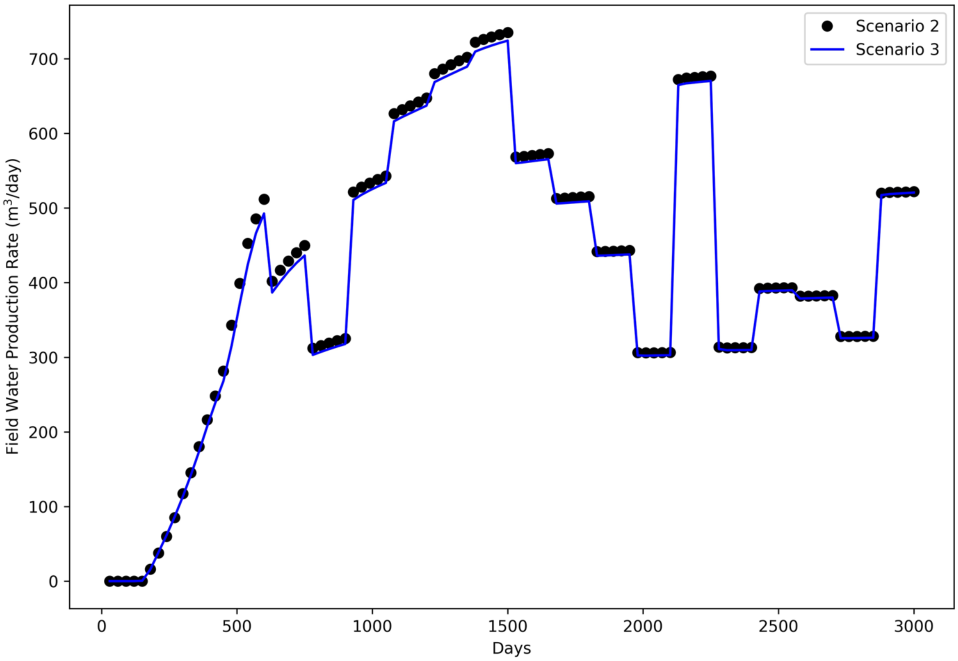


Fig. 21 Plot of GWO-optimized field water production rates, comparison between Scenarios 2 and 3

faster than the initial Egg Model in terms of optimization time.

There is also an important concern about the number of reservoir simulations required for building the proxy models. Some literatures (Mohaghegh 2011; He et al. 2016; Vida et al. 2019; Shahkarami and Mohaghegh 2020) suggested a rule of thumb that 10 to 15 simulations could be sufficient for the development of robust proxy models. Nonetheless, Nait Amar et al. (2018) had run 75 simulations to generate the necessary database to develop the proxy models. Moreover, Golzari et al. (2015) even performed 200 simulations to build the data-driven models. Therefore, there is no strict rule of how many simulations are exactly needed to establish the spatiotemporal database. It is widely dependent upon the purpose of application of the data-driven models. Also, the bigger the database, the more accurate the data-driven model can be. Despite this, we need to understand that there is always a trade-off

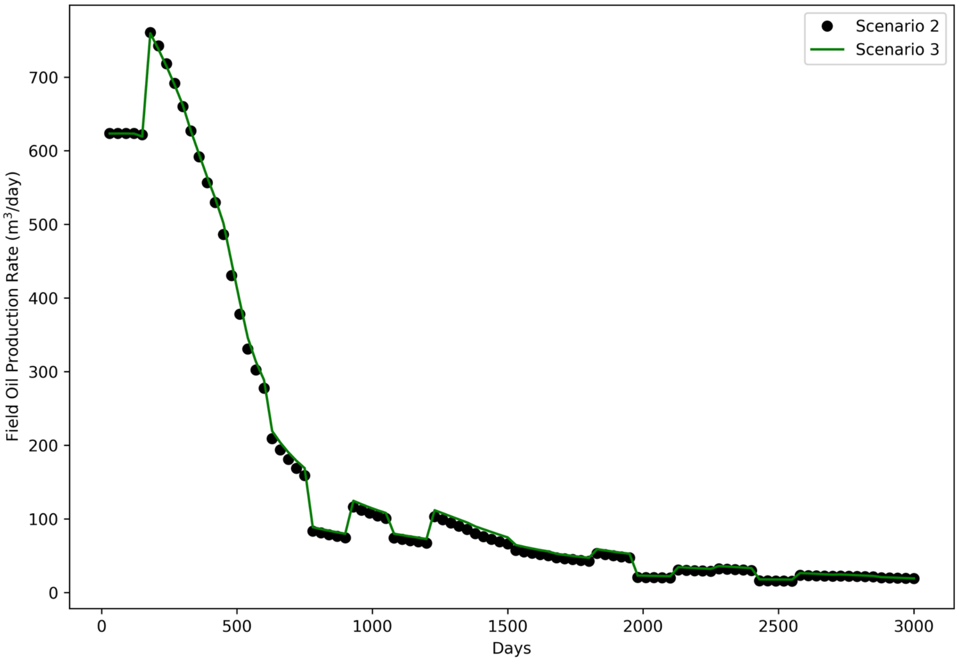


Fig. 22 Plot of PSO-optimized field oil production rates, comparison between Scenarios 2 and 3

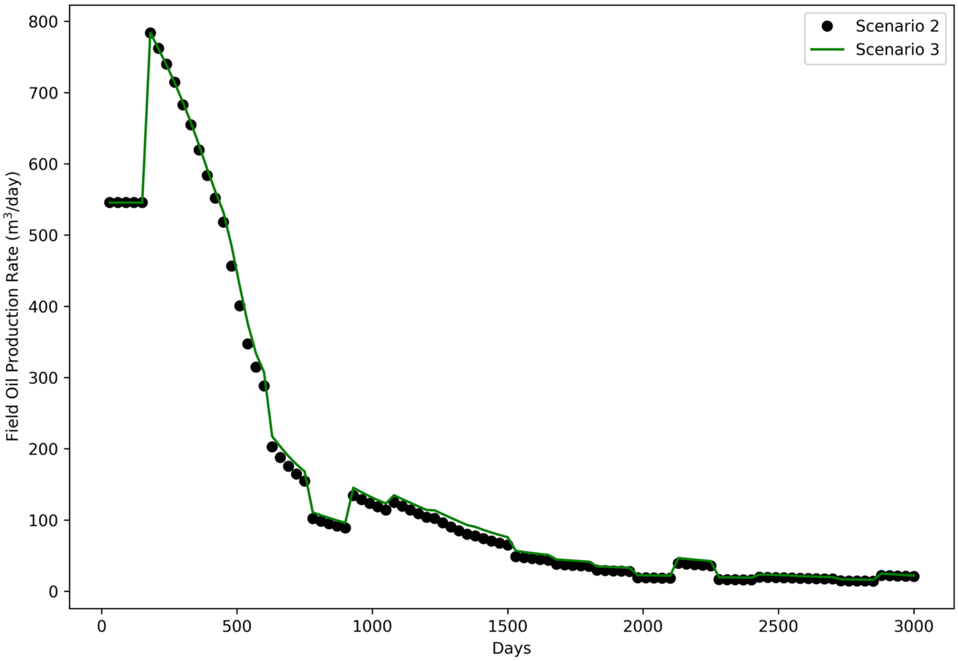


Fig. 23 Plot of GWO-optimized field oil production rates, comparison between Scenarios 2 and 3

between the size of the database and the computational time. When the database is humongous, it means that the neural network training might take longer time to complete. This challenge is termed as the curse of dimensionality. In this paper, we empirically selected to run 60 simulations as explained to establish our spatiotemporal database. Upon building the proxies, we performed blind validation with 80 new data samples as discussed. Based on the results shown, it can be deduced that this database was deemed to be practically sufficient to yield useful proxies. As presented in other literatures (Nait Amar et al. 2018; Amini and Mohaghegh 2019; Shahkarami and Mohaghegh 2020), the number of blind validation cases usually is about 10 or even less. In our work, we presented 80 blind validation cases to further demonstrate the higher feasibility of practical application of our data-driven proxies.

Fig. 24 Plot of PSO-optimized field water production rates (Scenario 2) and those of unoptimized case

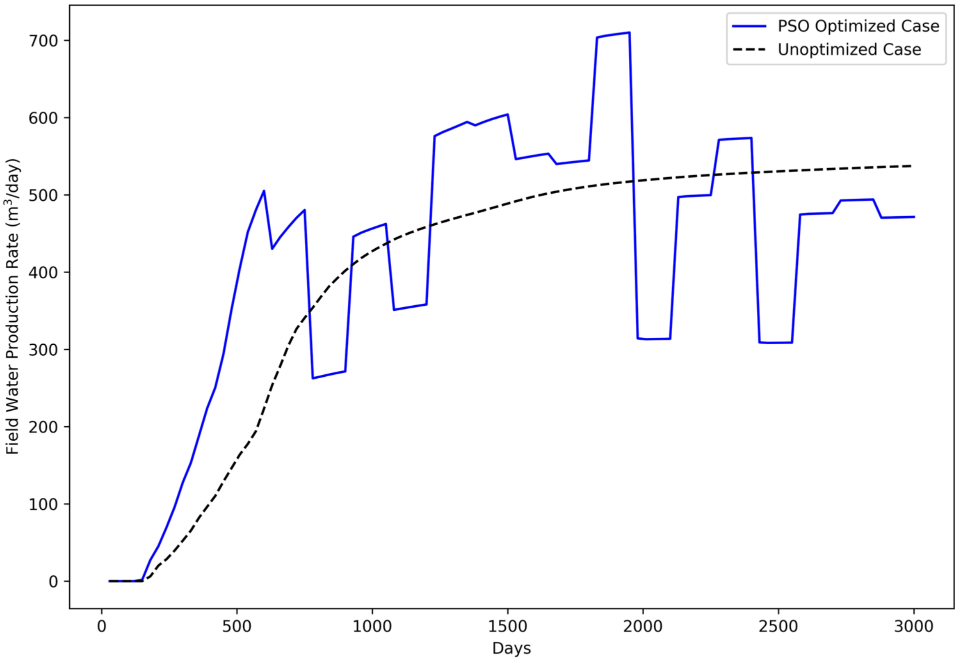
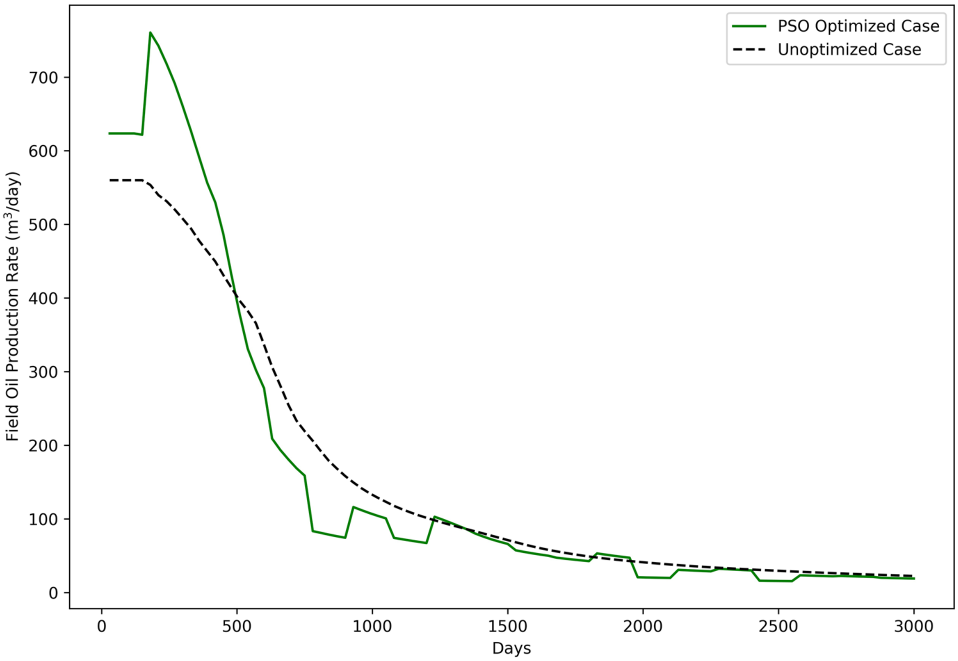


Fig. 25 Plot of PSO-optimized field oil production rates (Scenario 2) and those of unoptimized case



Conclusions

In this work, we implemented ML technique to build dynamic proxy models and conduct the optimization of well control rates on two waterflooding case studies, i.e., a 2D synthetic reservoir model and the 3D Egg Model. The main objective was to achieve the maximization of NPV by determining the optimal control rate with the help of two metaheuristic algorithms, which include PSO and GWO. In order to do that, for each case study, we maneuvered the modeling of ANN to build two proxy models in which one could predict the field liquid production rates at a certain time, and another could forecast the field water cut. Thereafter, we successfully coupled these models with the optimization algorithms to perform the waterflooding optimization. Based upon our investigation, GWO generally outperformed

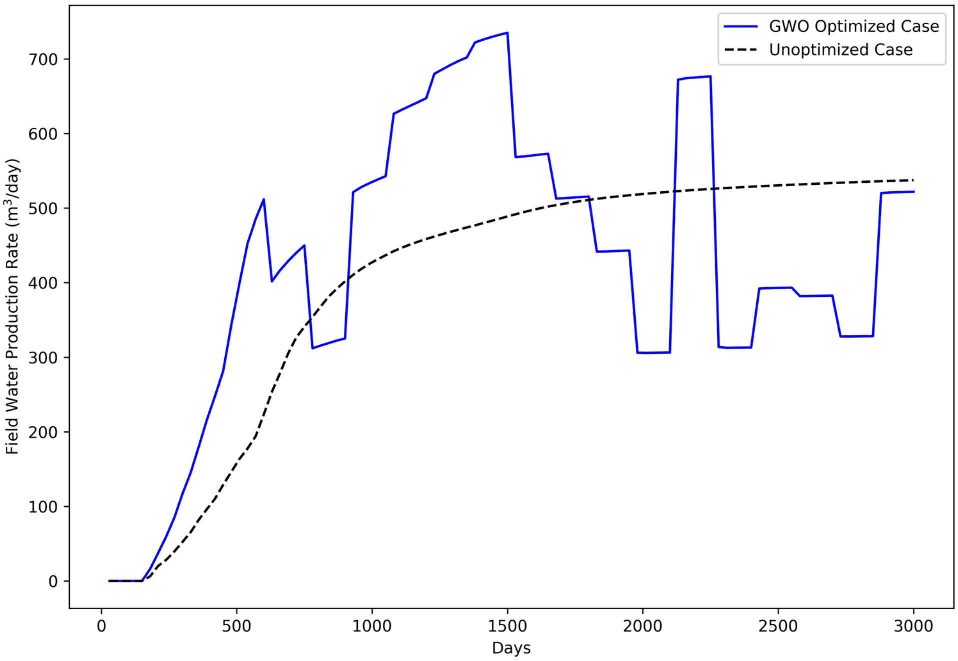


Fig. 26 Plot of GWO-optimized field water production rates (Scenario 2) and those of unoptimized case

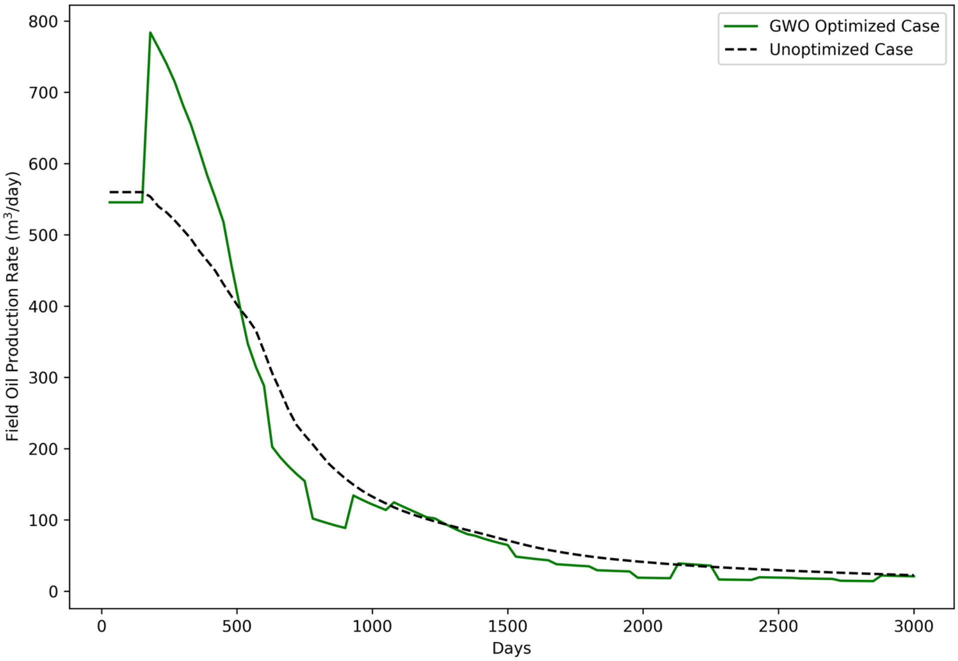


Fig. 27 Plot of GWO-optimized field oil production rates (Scenario 2) and those of unoptimized case

PSO in the context of optimization. However, the accuracy of results (prediction of optimized field liquid production rates and field water cut) was slightly higher when the proxies were coupled with PSO. This could be due to the sampling strategy applied in this study. Nonetheless, we conclude that the data-driven proxies have successfully served their purpose of application. Also, the results derived from this study verify the validity of the methodology presented in data-driven proxy modeling.

Acknowledgements This research is a part of BRU21—NTNU Research and Innovation Program on Digital Automation Solutions for the Oil and Gas Industry (www.ntnu.edu/bru21).

Funding This study is supported by Norwegian University of Science and Technology (NTNU).

References

Alakeely A, Horne RN (2020) Simulating the behavior of reservoirs with convolutional and recurrent neural networks. SPE Reserv Eval Eng

Alenezi F, Mohaghegh S (2017) Developing a smart proxy for the SACROC water-flooding numerical reservoir simulation model. In: SPE Western Regional Meeting Proceedings

Amini S, Mohaghegh S (2019) Application of machine learning and artificial intelligence in proxy modeling for fluid flow in porous media. Fluids. https://doi.org/10.3390/fluids4030126

Babaei M, Pan I (2016) Performance comparison of several response surface surrogate models and ensemble methods for water injection optimization under uncertainty. Comput Geosci. https://doi.org/10.1016/j.cageo.2016.02.022

Baumann EJM, Dale SI, Bellout MC (2020) FieldOpt: a powerful and effective programming framework tailored for field development optimization. Comput Geosci. https://doi.org/10.1016/j.cageo.2019.104379

Bellout MC, Echeverría Ciaurri D, Durlofsky LJ et al (2012) Joint optimization of oil well placement and controls. Comput Geosci. https://doi.org/10.1007/s10596-012-9303-5

Bruce WA (1943) An electrical device for analyzing oil-reservoir behavior. Trans AIME. https://doi.org/10.2118/943112-g

Buduma N, Locascio N (2017) Fundamentals of deep learning : Designing Next-Generation Machine Intelligence Algorithms

Dige N, Diwekar U (2018) Efficient sampling algorithm for large-scale optimization under uncertainty problems. Comput Chem Eng 115:431–454. https://doi.org/10.1016/j.compchemeng.2018.05.007

Ertekin T, Sun Q (2019) Artificial intelligence applications in reservoir engineering: a status check. Energies

Forouzanfar F, Reynolds AC (2013) Well-placement optimization using a derivative-free method. J Pet Sci Eng. https://doi.org/10.1016/j.petrol.2013.07.009

Forouzanfar F, Reynolds AC (2014) Joint optimization of number of wells, well locations and controls using a gradient-based algorithm. Chem Eng Res Des. https://doi.org/10.1016/j.cherd.2013.11.006

Golzari A, Haghighat Sefat M, Jamshidi S (2015) Development of an adaptive surrogate model for production optimization. J Pet Sci Eng. https://doi.org/10.1016/j.petrol.2015.07.012

Guo Z, Reynolds AC (2018) Robust life-cycle production optimization with a support-vector-regression proxy. SPE J. https://doi.org/10.2118/191378-PA

Guyaguler B, Horne RN, Rogers L, Rosenzweig JJ (2002) Optimization of well placement in a gulf of Mexico waterflooding project. SPE Reserv Eval Eng. https://doi.org/10.2118/78266-PA

Hammersley JM, Handscomb DC (1964) Monte Carlo methods

He Q, Mohaghegh SD, Liu Z (2016) Reservoir simulation using smart proxy in SACROC unit - Case study. In: SPE Eastern Regional Meeting

Hemmati-Sarapardeh A, Larestani A, Nait Amar M, Hajirezaie S (2020) Introduction. In: Applications of artificial intelligence techniques in the petroleum industry

Hong AJ, Bratvold RB, Nævdal G (2017) Robust production optimization with capacitance-resistance model as precursor. Comput Geosci. https://doi.org/10.1007/s10596-017-9666-8

Hong A, Bratvold RB, Lake LW (2019) Fast analysis of optimal improved-oil-recovery switch time using a two-factor production model and least-squares Monte Carlo algorithm. SPE Reserv Eval Eng. https://doi.org/10.2118/191327-PA

Jansen JD, Douma SD, Brouwer DR, et al (2009) Closed-loop reservoir management. In: SPE Reservoir Simulation Symposium Proceedings

Jansen JD, Fonseca RM, Kahrobaei S et al (2014) The egg model - a geological ensemble for reservoir simulation. Geosci Data J. https://doi.org/10.1002/gdj3.21

Kalantari-Dahaghi A, Mohaghegh SD (2011) A new practical approach in modelling and simulation of shale gas reservoirs: application to New Albany Shale. Int J Oil, Gas Coal Technol. https://doi.org/10.1504/IJOGCT.2011.038925

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: IEEE International Conference on Neural Networks - Conference Proceedings

Kingma DP, Ba JL (2015) Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings

Liang X, Weber DB, Edgar TF, et al (2007) Optimization of oil production based on a capacitance model of production and injection rates. In: SPE Hydrocarbon Economics and Evaluation Symposium

Lu R, Forouzanfar F, Reynolds AC (2017) An efficient adaptive algorithm for robust control optimization using StoSAG. J Pet Sci Eng. https://doi.org/10.1016/j.petrol.2017.09.002

McKay MD, Beckman RJ, Conover WJ (1979) A Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics. https://doi.org/10.2307/1268522

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mohaghegh SD (2006) Quantifying uncertainties associated with reservoir simulation studies using surrogate reservoir models. In: Proceedings - SPE Annual Technical Conference and Exhibition

Mohaghegh SD (2011) Reservoir simulation and modeling based on artificial intelligence and data mining (AI&DM). J Nat Gas Sci Eng. https://doi.org/10.1016/j.jngse.2011.08.003

Mohaghegh SD (2013) Reservoir modeling of shale formations. J. Nat. Gas Sci. Eng.

Mohaghegh SD (2017a) Data-driven reservoir modeling

Mohaghegh SD (2017b) Shale analytics

Mohaghegh SD, Hafez H, Gaskari R, et al (2006) Uncertainty analysis of a giant oil field in the middle east using surrogate reservoir model. In: 12th Abu Dhabi international petroleum exhibition and conference, ADIPEC 2006: meeting the increasing oil and gas demand through innovation

Mohaghegh SD, Liu J, Gaskari R, et al (2012) Application of surrogate reservoir model (SRM) to an onshore green field in Saudi Arabia; case study. In: Society of Petroleum Engineers - North Africa Technical Conference and Exhibition 2012, NATC 2012: Managing Hydrocarbon Resources in a Changing Environment

Mohaghegh SD, Gaskari R, Maysami M (2017) Shale analytics: Making production and operational decisions based on facts: A case study in marcellus shale. In: Society of Petroleum Engineers - SPE Hydraulic Fracturing Technology Conference and Exhibition 2017

Nait Amar M, Zeraibi N, Redouane K (2018) Optimization of WAG process using dynamic proxy, genetic algorithm and ant colony optimization. Arab J Sci Eng. https://doi.org/10.1007/s13369-018-3173-7

Nait Amar M, Zeraibi N, Jahanbani Ghahfarokhi A (2020) Applying hybrid support vector regression and genetic algorithm to water alternating CO2 gas EOR. Greenh Gases Sci Technol. https://doi.org/10.1002/ghg.1982

Navrátil J, Kollias G, King AJ, et al (2019) Accelerating physics-based simulations using neural network proxies: an application in oil reservoir modeling. arXiv

Ng CSW, Jahanbani Ghahfarokhi A, Nait Amar M, Torsæter O (2021) Smart proxy modeling of a fractured reservoir model for production optimization: implementation of metaheuristic algorithm and probabilistic application. Nat Res Resour. https://doi.org/10.1007/s11053-021-09844-2

Ogbeiwi P, Aladeitan Y, Udebhulu D (2018) An approach to waterflood optimization: case study of the reservoir X. J Pet Explor Prod Technol. https://doi.org/10.1007/s13202-017-0368-5

Peaceman DW (1977) Fundamentals of numerical reservoir simulation. Elsevier, Amsterdam

Pouladi B, Karkevandi-Talkhooncheh A, Sharifi M et al (2020) Enhancement of SPSA algorithm performance using reservoir quality maps: application to coupled well placement and control optimization problems. J Pet Sci Eng. https://doi.org/10.1016/j.petrol.2020.106984

Sarma P, Chen WH, Durlofsky LJ, Aziz K (2008) Production optimization with adjoint models under nonlinear control-state path inequality constraints. SPE Reserv Eval Eng. https://doi.org/10.2118/99959-pa

Sayarpour M, Zuluaga E, Kabir CS, Lake LW (2007) The use of capacitance-resistive models for rapid estimation of waterflood performance and optimization. In: Proceedings - SPE Annual Technical Conference and Exhibition

Shahkarami A, Mohaghegh S (2020) Applications of smart proxies for subsurface modeling. Pet Explor Dev. https://doi.org/10.1016/S1876-3804(20)60057-X

Shahkarami A, Mohaghegh S, Gholami V et al (2014) Modeling pressure and saturation distribution in a CO2 storage project using a surrogate reservoir model (SRM). Greenh Gases Sci Technol. https://doi.org/10.1002/ghg.1414

Shahkarami A, Mohaghegh SD, Hajizadeh Y (2018) Assisted history matching using pattern recognition technology. Int J Oil Gas Coal Technol. https://doi.org/10.1504/IJOGCT.2018.090966

Shi Y, Eberhart R (1998) Modified particle swarm optimizer. In: Proceedings of the IEEE Conference on Evolutionary Computation, ICEC

Sobol IM (1967) On the distribution of points in a cube and the approximate evaluation of integrals. USSR Comput Math Math Phys. https://doi.org/10.1016/0041-5553(67)90144-9

Teixeira AF, Secchi AR (2019) Machine learning models to support reservoir production optimization. In: IFAC-PapersOnLine

Thakur GC (1996) What is reservoir management? JPT J Pet Technol. https://doi.org/10.2118/26289-JPT

Udy J, Hansen B, Maddux S, et al (2017) Review of field development optimization of waterflooding, EOR, and well placement focusing on history matching and optimization algorithms. Processes

Valladão DM, Torrado RR, Flach B, Embid S (2013) On the stochastic response surface methodology for the determination of the development plan of an oil & gas field. In: Society of Petroleum Engineers - SPE Intelligent Energy International 2013: Realising the Full Asset Value

Van Essen GM, Zandvliet MJ, Van Den Hof PMJ et al (2009) Robust waterflooding optimization of multiple geological scenarios. SPE J. https://doi.org/10.2118/102913-PA

Vida G, Shahab MD, Mohammad M (2019) Smart proxy modeling of SACROC CO2-EOR. Fluids. https://doi.org/10.3390/fluids4020085

Volkov O, Bellout MC (2017) Gradient-based production optimization with simulation-based economic constraints. Comput Geosci. https://doi.org/10.1007/s10596-017-9634-3

Wang L, Li ZP, Adenutsi CD et al (2021) A novel multi-objective optimization method for well control parameters based on PSO-LSSVR proxy model and NSGA-II algorithm. J Pet Sci Eng. https://doi.org/10.1016/j.petrol.2020.107694

Xu C, Nait Amar M, Ghriga MA et al (2020) Evolving support vector regression using Grey Wolf optimization; forecasting the geomechanical properties of rock. Eng Comput. https://doi.org/10.1007/s00366-020-01131-7

Yousef AA, Gentil P, Jensen JL, Lake LW (2006) A capacitance model to infer interwell connectivity from production- and injection-rate fluctuations. SPE Reserv Eval Eng. https://doi.org/10.2118/95322-pa

Zhang K, Zhang LM, Yao J et al (2014) Water flooding optimization with adjoint model under control constraints. J Hydrodyn. https://doi.org/10.1016/S1001-6058(14)60009-3

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.