

Production optimization under waterflooding with Long Short-Term Memory and metaheuristic algorithm

Cuthbert Shang Wui Ng ^{a,*}, Ashkan Jahanbani Ghahfarokhi ^a, Menad Nait Amar ^b

^a Department of Geoscience and Petroleum, Norwegian University of Science and Technology, Trondheim, Norway

^b Département Etudes Thermodynamiques, Division Laboratoires, Sonatrach, Boumerdes, Algeria

ARTICLE INFO

Article history:

Received 26 August 2021

Received in revised form

26 November 2021

Accepted 31 December 2021

Keywords:

Production optimization

Numerical reservoir simulation

Machine learning

Long short-term memory (LSTM)

Dynamic proxies

Particle swarm optimization (PSO)

ABSTRACT

In petroleum domain, optimizing hydrocarbon production is essential because it does not only ensure the economic prospects of the petroleum companies, but also fulfills the increasing global demand of energy. However, applying numerical reservoir simulation (NRS) to optimize production can induce high computational footprint. Proxy models are suggested to alleviate this challenge because they are computationally less demanding and able to yield reasonably accurate results. In this paper, we demonstrated how a machine learning technique, namely Long Short-Term Memory, was applied to develop proxies of a 3D reservoir model. Sampling techniques were employed to create numerous simulation cases which served as the training database to establish the proxies. Upon blind validating the trained proxies, we coupled these proxies with particle swarm optimization to conduct production optimization. Both training and blind validation results illustrated that the proxies had been excellently developed with coefficient of determination, R^2 of 0.99. We also compared the optimization results produced by NRS and the proxies. The comparison recorded a good level of accuracy that was within 3% error. The proxies were also computationally 3 times faster than NRS. Hence, the proxies have served their practical purposes in this study.

© 2022 Southwest Petroleum University. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In petroleum industry, reservoir management (RM) is one of the domains that has been emphasized by many oil and gas companies. According to Wiggins and Startzman [1], RM is termed as the employment of available technology, financial and labor resources to optimize the economic performance and recovery of a reservoir. They [1] further expounded that RM could be fathomed as a sequence of operations from its initial discovery of a reservoir to its final abandonment. In this case, production optimization is one of the pivotal parts in RM. Oil and gas companies attempt to optimize hydrocarbon production not only to fulfill the increasing demand

for energy, but also to ensure their higher economic returns. One of the approaches of increased production is to perform waterflooding or water injection. Waterflooding is generally implemented to produce additional volume of hydrocarbon after primary recovery which relies upon natural mechanisms such as gas cap drive and gravitational drainage [2]. Additionally, careful planning and implementation of waterflooding are important to avoid any unnecessary expenditure during the implementation phase. Hence, waterflooding optimization has been emphasized in the research field [3–7] for years to help the oil and gas companies to improve their application of this technique.

To be more precise, waterflooding optimization is considered as one of the engineering problems that requires some mathematical algorithms to come up with some design parameters, which either maximize or minimize any predefined objective function [2,8]. Regarding this, these design parameters include well production rates, well injection rates, bottomhole pressure of well, initiation time of waterflooding, and so forth. More intriguingly, waterflooding problem can also be formulated into a multi-objective problem in which more than one objective function is optimized

* Corresponding author.

E-mail address: cuthbert.s.w.ng@ntnu.no (C.S.W. Ng).

Peer review under responsibility of Southwest Petroleum University.



Production and Hosting by Elsevier on behalf of KeAi

[9–11]. This formulation provides more useful insights to the chemical or petroleum engineers as it has closer proximity to the real-life problem. Additionally, numerical reservoir simulation (NRS) is one of the most widely applied tools of reservoir modeling during the field development stage. NRS can be conveniently employed along with other algorithms to solve any problem related to production optimization. However, one of its drawbacks is that more computational effort is required if it is used to model a geologically sophisticated reservoir [12,13]. This is because NRS uses mathematical equations and physics-based approach to model the flow of fluid in the subsurface. Thus, the computational time of the fluid flow modeling undeniably increases as the complexity of the reservoir modeled increases. Mitigating this computational challenge has been one of the most prevalent research topics.

Thanks to data-driven technology, the computational challenge can be alleviated. Data-driven technology is a framework that applies any input and output data provided to establish a relationship among them [13]. A model that is yielded from this technology is known as “data-driven model”. In this aspect, the main building block of this technology is data. More importantly, machine learning (ML) is one of the techniques used for data-driven modeling. Examples of ML generally include artificial neural network, support vector machine, random forest, extreme gradient boosting, and so on. In addition, data-driven model has displayed its ability to be used as a proxy or surrogate model of NRS. Regarding this, a proxy or surrogate model in general acts as a substitute of NRS and is computationally faster and able to replicate the results of NRS within satisfied level of accuracy. In this context, Dr. Shahab Mohaghegh is one of the pioneers in the petroleum industry to have coined the term of smart proxy model (SPM). SPM is a proxy model that comprises an ensemble of numerous inter-linked neuro-fuzzy systems, which are trained to understand the fluid flow behaviors from NRS [13,14]. SPM has been demonstrated to be successful in different fields of application, including uncertainty analysis [15,16], CO₂ sequestration and utilization [17,18], history matching [19,20], waterflooding [21], and unconventional resources [22,23]. Apart from these, there are other captivating literatures [24–33] discussing the use of ML-based models in the petroleum domains. These literatures in general also elaborated on the high applicability of ML techniques to be employed as a substitute of NRS. Nevertheless, one of the limitations of ML-based proxy modeling is the sufficiency of data. This is because the established ML-based model might not be able to “learn properly” without being supplied with sufficient data. However, when it is provided with too much data, this might undermine the significance of proxy modeling as a lot of simulation runs have to be performed.

Other than being used as proxy models, ML techniques have portrayed their value in the development of predictive models. In this case, Talebkeikhah et al. [34] successfully implemented seven ML methods, based on 1000 experimental points from some Iranian crude samples, to develop the predictive models of viscosity at reservoir conditions. These methods include radial basis function neural network, multilayer perceptron, support vector regression, adaptive neuro-fuzzy inference system, decision trees, and random forest. Besides that, Nait Amar et al. [35] illustrated how the best two out of various developed ML-based models were chosen and combined under the paradigm of Committee Machine Intelligent System (CMIS) to develop a model that could forecast thermal conductivity of carbon dioxide. They further showed the use of weight average approach and group method of data handling (GDMH) to establish the CMIS models. A similar approach was employed and discussed by Mehrjoo et al. [36] to create a predictive model of interfacial tension of methane-brine systems at high pressure and salinity conditions. Also, based on 1985 experimental

points, Nait Amar et al. [37] successfully applied gene expression programming to perform the modeling of density of binary and tertiary mixtures of ionic liquids and molecular solvents.

In this work, we have used an advanced ML technique that is Long Short-Term Memory (LSTM) to build two proxy models, which are correspondingly applied to predict field liquid production rate (FLPR) and field water cut (FWCT). It is essential to point out that the proxy models built here are considered as “dynamic proxies”, which are time-dependent. As Nait Amar et al. [24] stated, time-dependent proxies offer higher flexibility in terms of application under time-dependent constraints. As the two abovementioned dynamic proxies were developed, they were coupled with particle swarm optimization (PSO) to conduct the waterflooding optimization. The details would be presented in the next few sections. After this introduction, the paper is followed by the theoretical framework that generally briefs the techniques involved and the general methodology used in this work. Thereafter, results and discussion about the main findings of this work are presented. The paper then ends with some conclusive remarks derived from this work.

2. Theoretical framework

2.1. Long Short-Term Memory (LSTM)

LSTM is a more advanced version of recurrent neural network (RNN) that is developed to process sequential data, such as texts, sentences, and so on [38]. A simple RNN is generally designed to preserve and deliver information from the current step to the next one [38]. However, a simple RNN suffers the problem of vanishing gradient in which a long-term information cannot be fully utilized [39]. Thus, large amount of previous information is unable to be stored to perform forecast within higher level of accuracy. To elude the problem of vanishing gradient, Hochreiter and Schmidhuber [39] built the LSTM in 1997. The fundamental topology of the LSTM used in this study is demonstrated in Fig. 1. The mathematical formulation of LSTM is shown below:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$

$$\tilde{c}_t = \gamma(W_c x_t + U_c h_{t-1} + b_c) \quad (3)$$

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \quad (4)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t \times \gamma(c_t) \quad (6)$$

The mechanism of LSTM revolves around a cell state c_t . Around the cell state, information is either added or removed via three gates, for instance forget gate f_t , input gate i_t , and output gate o_t . These gates evaluate if the sequential input data should be retained to save pertinent information to the latter stages. Thereafter, according to Equation (1), the forget gate decides on the addition or omission of information. Regarding this, the information in terms of input and hidden state will be saved (removed) if f_t is close to one (zero). Besides that, the input gate is calculated to update the cell state. Via this update, the evaluation of the importance of the input delivered to the next cell is done. Furthermore, the output gate computes the output for the hidden states based on Equation (6). It can be noticed that the activation function and the recurrent

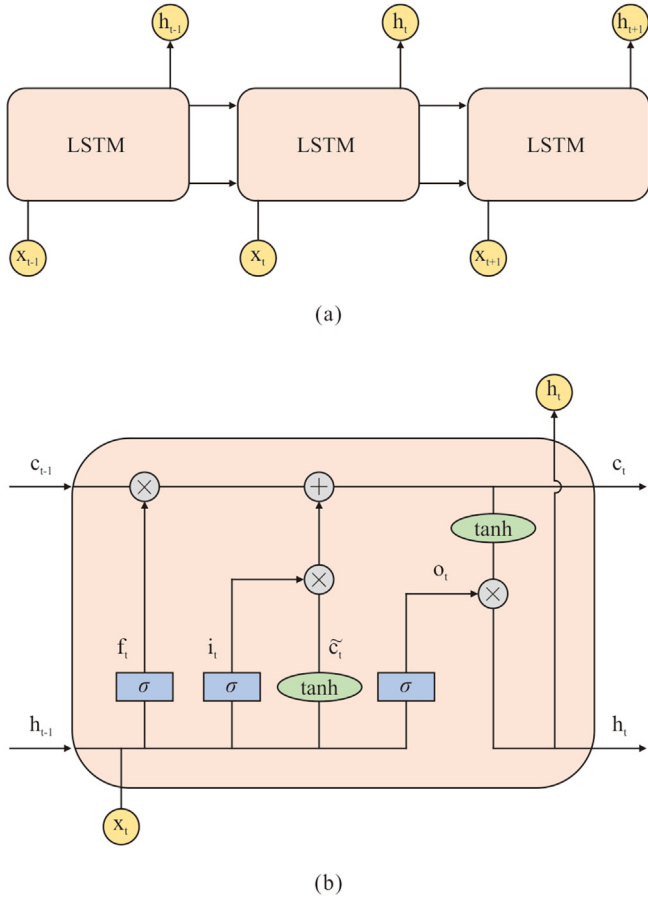


Fig. 1. Architecture of Long Short-Term Memory (LSTM): (a) General topology of LSTM. (b) Detailed structure of LSTM.

activation function used in LSTM are respectively hyperbolic tangent function (indicated as tanh) and sigmoid function (denoted as σ).

2.2. Particle swarm optimization (PSO)

In 1995, Kennedy and Eberhart [40] established an optimization algorithm which was known as PSO. In this case, PSO is considered as an example of nature-inspired algorithms because it is formulated by simulating the behavior of flying stock of birds. Mathematically speaking, a swarm of particles indicates several possible solutions to an optimization problem. The status of each particle is computed according to its position and velocity. In this context, the dimension of both position and velocity is the same as the number of optimization parameters. In general, the algorithm commences through the random initialization of the position and velocity of each particle. A cost function, like mean squared error (MSE), is then employed to determine the fitness of each particle. After that, $pbest$ and $gbest$ are computed and saved to update the velocity at current iteration based on Equation (7). In this context, $pbest$ and $gbest$ are found out for every iteration. $pbest$ is the best position of a particle in the dimensional space and $gbest$ is the overall best position of a particle hitherto in the whole swarm. Upon determining the velocity at next iteration, the position of a particle for the next iteration is updated as captured by Equation (8). After a predefined number of iterations, each particle updates its position by minimizing the fitness value until the convergence of the optimal position occurs.

$$v_{jk, t+1} = \omega v_{jk, t} + c_1 r_1 (pbest_{jk, t} - x_{jk, t}) + c_2 r_2 (gbest_{k, t} - x_{jk, t}) \quad (7)$$

$$x_{jk, t+1} = x_{jk, t} + v_{jk, t+1} \quad (8)$$

In Equation (7), $v_{jk, t}$ corresponds to the velocity of the j th particle at step t in k th dimension. $x_{jk, t}$ is its respective position. c_1 and c_2 correspondingly represent the cognitive and social learning factors that regulate the local and global search of the optimal solution. These parameters are selected by trial-and-error approach. r_1 and r_2 are random numbers extracted from uniform distribution of (0, 1). ω is inertial weight that was suggested by Shi and Eberhart [41] to better handle the convergence issue.

Apart from PSO, we would like to reiterate that there are several other metaheuristic algorithms that can be employed to perform modeling and optimization tasks. Examples of these algorithms [42] include, but are not limited to, genetic algorithm, differential evolution, simulated annealing, and ant colony optimization. In this aspect, PSO has been selected due to its computational efficiency and perceivable concept as being briefed in the literature [43]. Also, it has exhibited good results in some of our previous works [29,31,44].

2.3. Formulation of optimization problem and dynamic proxy

One of the most important perceptions about developing a proxy model is that it is an objective-oriented task. This implies that the background of the optimization problem must be clearly understood to provide better insights of proxy modeling. By perceiving the optimization problem, the modelers would know what variables or design parameters should be involved in creating the relevant proxies. Hence, formulation of optimization problem is indeed necessary in the development of proxies. In this work, the selected objective function is net present value (NPV), and it is mathematically shown in Equation (9).

$$NPV(\mathbf{u}) = \sum_{i=1}^{n_{total}} \frac{(Q_o^i(\mathbf{u})P_o - Q_w^i(\mathbf{u})P_w - Q_{wi}^i(\mathbf{u})P_{wi}) \times \Delta t_i}{(1 + \text{interest rate})^{t_i/D}} \quad (9)$$

where \mathbf{u} is the vector of optimization parameters, Q^i is the field production (injection) rate at timestep i and P represents price or cost. The subscripts of o , w , and wi respectively indicate oil, water, and water injected. In this work, P_o is 70 USD/bbl whereas both P_w and P_{wi} are 2 USD/bbl. Also, the optimization parameter used here is the field injection rate. Therefore, the optimization problem pertains to the adjustment of field water injection rate per 150 days for the period of 3000 days. Moreover, Δt_i is the difference of time between current and previous timestep. Besides that, t_i is the elapsed time from beginning until step i and D is the reference time for discounting. D is 365 days as interest rate has a unit of fraction per year and discounting of cash flow is done daily. The interest rate used here is 0.1 per year.

It is noticeable that the dynamic proxies developed here need to yield two parameters, which are field oil and water production rates (FOPR and FWPR). Therefore, by implementing LSTM method, we built two different dynamic proxy models, which respectively predict FLPR and FWCT at a specific timestep. Moreover, the input parameters are the number of days at every timestep i , t_i ; the harmonic mean of grid absolute permeability for every layer of formation, $k_{harmonic}$; the standard deviation of grid absolute permeability for each formation layer, $k_{Std Dev}$; the permeabilities of completed grid blocks (injectors and producers), $k_{\{inj,prod\}}$; the field

water injection rate, \mathbf{u} ; the output value at previous timestep, y_{i-1} . The mathematical formulation of the proxies¹ is illustrated in Equation (10). The harmonic mean of permeability for every formation layer is given by Equation (11).

$$y_i = f(t_i, k_{\text{harmonic}}, k_{\text{Std Dev}}, k_{\{\text{inj. prod}\}}, \mathbf{u}, y_{i-1}) \quad (10)$$

$$k_{\text{harmonic}} = \frac{\sum_{j=1}^m L_j}{\sum_{j=1}^m \frac{L_j}{k_j}} \quad (11)$$

where L_j represents the depth at the top of grid block j , k_j refers to the grid absolute permeability, and m denotes the number of grid blocks. Regarding the inputs of the permeabilities of completed grid blocks (injectors and producers), the reservoir model studied here is the ‘‘Egg Model’’ that was developed by Jansen et al. [45]. There are 7 layers in the reservoir model with 8 injectors and 4 producers. To avoid the curse of dimensionality, the arithmetic mean of the permeability of the completed grid blocks for every well is calculated and this will yield 12 permeability variables. There are also 14 variables of k_{harmonic} and $k_{\text{Std Dev}}$ given Egg Model has 7 layers. In total, there are 29 input variables used to train the dynamic proxies.

About the geological properties of Egg Model, its permeability is heterogeneous whereas its porosity is homogeneous with a value of 0.2. The initial water saturation for each grid block is 0.1. The dimension of each block is $8 \text{ m} \times 8 \text{ m} \times 4 \text{ m}$ with a total number of $60 \times 60 \times 7$ (only 18533 grid blocks are active). The horizontal permeability distribution of Egg Model is illustrated in Fig. 2. Refer to Jansen et al. [45] for the remaining details of the geological properties of this model. To be able to conduct the studies here, the control of both injectors and producers has been altered. In this aspect, the eight injectors are identical, and the rate is within the range of $40 \text{ m}^3/\text{day}$ and $100 \text{ m}^3/\text{day}$. Hence, the optimization problem considering the constraint is summarized as shown below:

$$\text{subject to } \begin{cases} \max \text{NPV (FWIR)} \\ 320 \text{ Sm}^3/\text{day} \leq \text{Field Water Injection Rate} \leq 800 \text{ Sm}^3/\text{day} \\ \text{Bottomhole Pressure of Each Producer} \geq 395 \text{ bar} \end{cases} \quad (12)$$

2.4. Data preparation, neural network training, and blind validation procedure

After completing the formulation of optimization problem and dynamic proxy modeling, we have a clearer idea of input and output variable types. Thereafter, we employ the methodology discussed and used in Ref. [31] to conduct the proxy modeling. With respect to this, a database needs to be generated and formatted that can be used to train the dynamic proxies. To create this database, we generate 60 different injection schedules by employing three sampling techniques, such as Latin Hypercube Sampling [46], Hammersley Sequence Sampling [47], and Sobol Sequence Sampling [48]. Each technique constitutes 20 schedules. Thereafter, each of the schedules is fed into the reservoir simulator to provide the necessary information to build the database.

¹ The permeability refers only to the horizontal permeability, here. Also, the permeability in both x- and y-directions are the same.

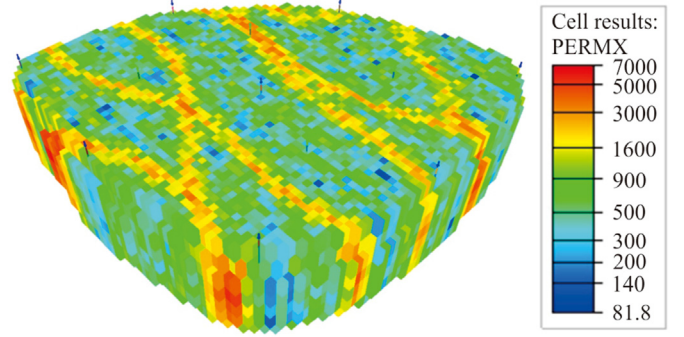


Fig. 2. Permeability distribution of egg model.

For illustrative purposes, the summary of the database is presented in Table 1. It is essential to highlight that the statistical parameters provided in Table 1 are determined ‘‘categorically’’. For instance, for the variable of k_{harmonic} , the maximum and minimum values are determined by finding the highest and lowest values of all the 7 variables of k_{harmonic} (knowing that there are 7 layers). By following this logic, the pertinent mean and standard deviation are computed.

Then, when the database is ready, it is normalized between 0 and 1 ‘‘categorically’’ using the following formula:

$$X_{\text{normalized}} = \frac{X_n - X_{\min}}{X_{\max} - X_{\min}} \quad (13)$$

where $X_{\text{normalized}}$ implies the normalized value of X_n whereas X_{\max} and X_{\min} correspondingly represent the maximum and minimum values of X . Then, the database was divided into training set (80% of the points), validation (10% of the data), and testing sets (the remaining 10%). Validation set is employed to prevent any overfitting issue during training whereas testing set is used to evaluate

the predictability of the model prior to proceeding to blind validation phase. If excellent performance is illustrated during training, validation, and testing stages, then we would proceed to generate the database of blind validation. In this case, we reapply each of the three abovementioned sampling methods to respectively create additional 80 injection scenarios. Thereafter, we evaluate if the prediction performance of the dynamic proxies is within satisfied level of accuracy. Upon finishing the blind validation phase, the proxies are prepared for application. In this paper, we have utilized two statistical metrics to evaluate the training and prediction performance of the models, namely coefficient of determination and root mean squared error. The formula of each metrics is correspondingly displayed as Equations (14) and (15).

$$R^2 = 1 - \frac{\sum_{i=1}^n (Y_i^{\text{proxy}} - Y_i^{\text{sim}})^2}{\sum_{i=1}^n (Y_i^{\text{proxy}} - \bar{Y})^2} \quad (14)$$

Table 1
Summary of database.

Types of Data	Number of Data Points	Maximum Value	Minimum Value	Mean Value	Standard Deviation
Static Data					
t_j	1×6000	3000	30	1515	865.98
k_{harmonic}	7×6000	632.21	593.84	616.18	15.29
$k_{\text{Std Dev}}$	7×6000	1458.26	660.57	1010.98	262.06
k_{inj}	8×6000	1890.14	333.03	783.62	471.59
k_{prod}	4×6000	3759.54	361.41	1332.09	1404.51
Dynamic Data					
u	1×6000	800	320	559.76	138.51
y_{i-1} and y_i (FLPR)	2×6000	800.04	0	556.82	143.66
y_{i-1} and y_i (FWCT)	2×6000	1	0	0.710	0.319

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (Y_i^{\text{proxy}} - Y_i^{\text{sim}})^2}{n}} \quad (15)$$

where Y_i indicates the output value, the superscripts proxy and sim represent the proxy model and reservoir simulator model, respectively, \bar{Y} is the mean value of the output, and n is the number of data points.

3. Results and discussion

Before proceeding to the results of our dynamic proxy models, it is essential to briefly explain that the trial-and-error approach has been implemented to determine the topology of our proxies. In this case, the dynamic proxy of FLPR has been built with one input layer, one hidden layer, and one output layer. There are 50 nodes used in the hidden layer. Besides that, the dynamic proxy of FWCT has the similar architecture as that of FLPR but with an additional hidden layer. Both hidden layers consist of 50 nodes. Besides that, one of the backpropagation algorithms, namely Adaptive Moment Estimation (Adam), has been applied to train both proxies. Peruse King and Ba [49] for details. Pertaining to the parameters considered for Adam, the number of training iterations is 2000, the learning rate is 0.001, exponential decay rate for the 1st moment estimates is 0.9, that for the 2nd moment estimates is 0.999, and numerical stability is 10^{-7} .

Fig. 3 illustrates the cross plot between the actual values and the predicted values for both proxies of FLPR and FWCT. Based on this plot, it is deducible that albeit the proxy of FLPR slightly outperforms that of FWCT, both proxies have undergone an excellent training phase. This is further supported by the results of training, validation, and testing performance displayed in Table 2. With respect to this, it can be confirmed that the overfitting issue has been prevented as the validation performances of both proxies are

Table 2
Training, validation, and testing performances of the dynamic proxies.

		LSTM-FLPR	LSTM-FWCT
Training	R^2	0.9999	0.9999
	RMSE	0.2447	0.0021
Validation	R^2	0.9999	0.9999
	RMSE	0.2565	0.0020
Testing	R^2	0.9999	0.9999
	RMSE	0.2361	0.0016

as good as those of training. This also proves that both proxies have gone through a healthy trend of training. It is often important to ensure that the proxies have been trained “healthily”. Otherwise, the developed proxies will have a very weak predictability by only “memorizing” and being able to predict the data from the training set within satisfied level of accuracy. In addition, it is demonstrated that both proxies have good prediction ability as they have shown splendid testing results. Nevertheless, both proxies still must proceed to blind validation stage to further evaluate their predictability before being practically applied to perform optimization in this work.

To conduct the blind validation, three different sampling methods have been used to correspondingly create 80 additional injection schedules as mentioned earlier. Hence, each of these schedules will yield a set of performance metrics for each proxy. To provide a better evaluation of blind validation performance, the mean of the metrics for each sampling technique is shown instead in Table 3. Based upon the results, it can be inferred that both proxies have been successfully blind validated and are prepared to be used for optimization. However, for illustrative purpose, the blind validation results of one of the samples retrieved by using Latin Hypercube method are displayed in Fig. 4. Although the blind validation dataset has not been used to develop the models, the models can still predict the outputs reasonably well. This further

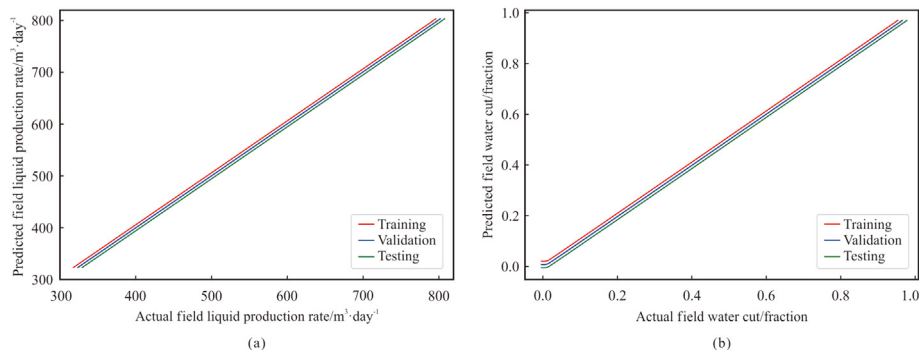


Fig. 3. Cross plot between actual and predicted values considering training, validation, and testing sets: (a) FLPR. (b) FWCT.

Table 3
Blind validation performances of the dynamic proxies considering three sampling techniques.

		LSTM-FLPR	LSTM-FWCT
Latin Hypercube	Mean R ²	0.9999	0.9992
	Mean RMSE	0.2513	0.0078
Sobol Sequence	Mean R ²	0.9999	0.9989
	Mean RMSE	0.2109	0.0093
Hammersley Sequence	Mean R ²	0.9999	0.9989
	Mean RMSE	0.2040	0.0092

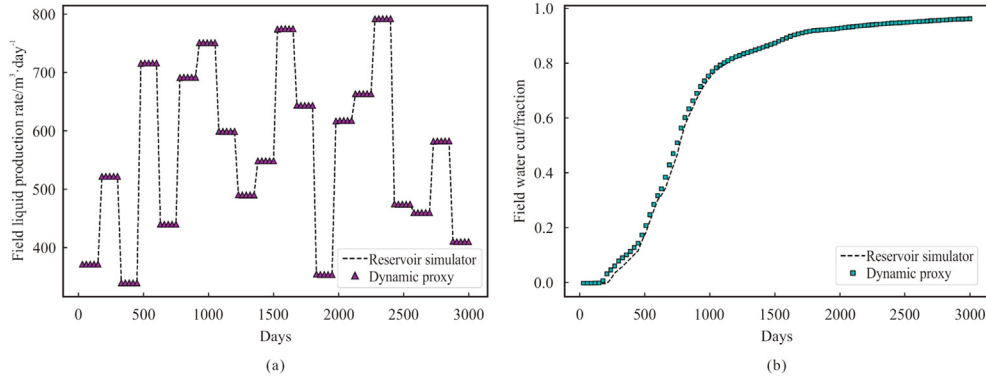


Fig. 4. Blind validation of Latin Hypercube sample set 32 (out of 80): (a) FLPR. (b) FWCT.

provides higher confidence regarding the integrity of the proxies built in this paper.

As it has been explained, both proxies of FLPR and FWCT have been coupled with PSO to conduct the waterflooding optimization. In this aspect, the FWIR would be periodically tuned to maximize the NPV for a certain period of production. Regarding the parameters of PSO, the inertial weight is 0.8 whereas both the social and cognitive learning factors are 1.05. Also, the number of iterations is initialized to be 100 in tandem with 15 particle swarms per iteration. The case in which the optimization is done by applying both proxies, is termed as “Dynamic Proxies”. Thereafter, to assess the proximity of results of optimization, the optimized FWIRs resulted from the case of “Dynamic Proxies” are fed into the simulator to compute its respective NPV. Such case of optimization is known as “Simulator-Dynamic Proxies” in this paper. To have a more comprehensive comparison, the reservoir simulator has also been coupled with PSO to conduct the optimization. This case is labeled as “Simulator”.

Upon completing these three cases, the optimal NPV obtained from each case is recorded in Table 4. In general, it is noticeable that the proxies have illustrated practically accurate results. When comparing the NPVs of “Simulator-Dynamic Proxies” and “Dynamic Proxies”, the error is calculated to be 2.6%. Furthermore, the error between “Simulator” and “Dynamic Proxies” is determined to be 1.6%. For illustrative purpose, the optimized FWIRs derived from “Simulator” and “Dynamic Proxies” are plotted in Fig. 5. More interestingly, regarding the strength of the models, the computational time for “Dynamic Proxies” is about 4 h whereas that of “Simulator” is about 12 h. Hence, the dynamic proxies are 3 times faster than the simulator for optimization in this study. This highlights the significance of the application of dynamic proxies. To

Table 4
Optimal NPV considering three cases.

Models	Simulator	Simulator-Dynamic Proxies	Dynamic Proxies
NPV _{optimal} (million USD)	155.89	154.39	158.34

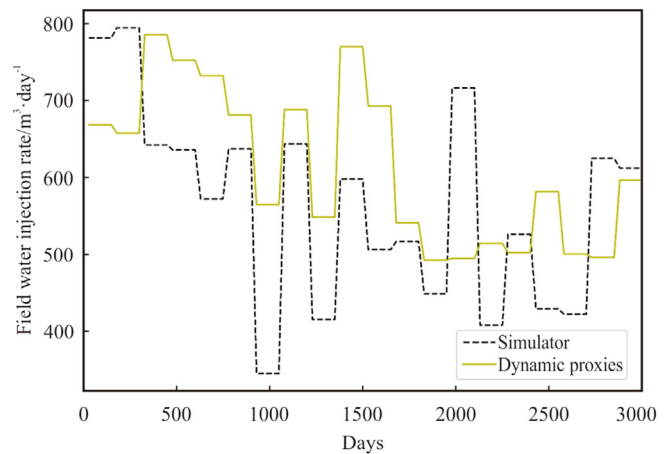


Fig. 5. Optimized FWIRs derived from Simulator and Dynamic Proxies.

further check the integrity of these proxies, the plot of optimized field water (oil) production rates between “Simulator-Dynamic Proxies” and “Dynamic Proxies” is illustrated in Fig. 6 (Fig. 7). The respective statistical evaluation is also tabulated in Table 5. Based on these results, both proxies have practically served their purposes of application by reaching satisfied level of accuracy with less demanding computational effort.

Nonetheless, there are a few limitations about the models developed in this work. As mentioned earlier, one of the limitations includes the application of the models. In this aspect, proxy modeling is an objective-driven task. Therefore, the established models can only be aptly employed to solve the optimization problem outlined. Besides that, there is a concern about the behavior of the training database as noise, which is an important issue to flow rate signal, is not considered in the data used. Hence, the models might not demonstrate high applicability when noisy data is introduced for optimization purpose. This is indeed part of the future works that is worth investigating.

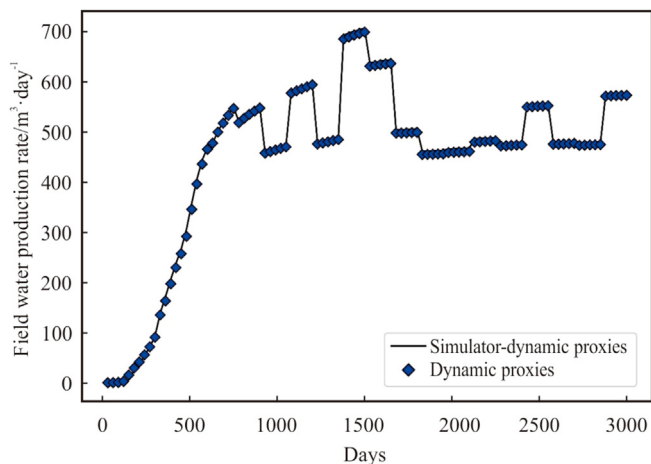


Fig. 6. Optimized FWPR derived from Simulator-Dynamic Proxies and Dynamic Proxies.

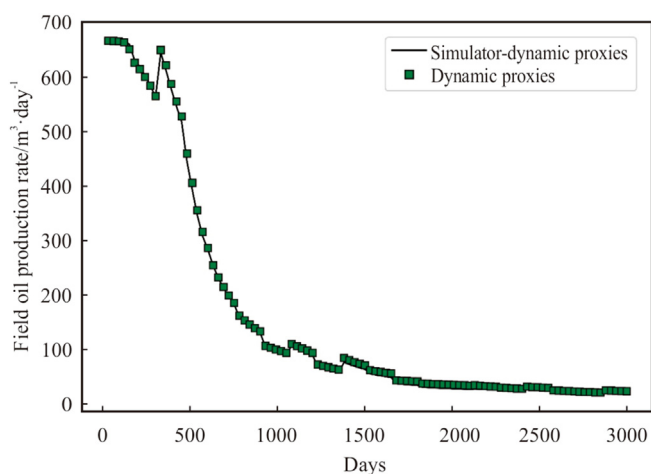


Fig. 7. Optimized FWOR derived from Simulator-Dynamic Proxies and Dynamic Proxies.

Table 5
Statistical evaluation of optimized FWPR and FOPR.

		Optimized FWPR	Optimized FOPR
Optimization	R ²	0.9990	0.9993
	RMSE	5.531	5.604

4. Conclusions

In this study, we applied the LSTM approach to develop two dynamic proxies, which correspondingly could predict FLPR and FWCT based upon a 3D reservoir model known as the “Egg Model”. One of the main objectives of this investigation was to study the applicability of LSTM to be employed as proxy models for production optimization. According to the training and blind validation results, it could be deduced that these two proxies could accurately emulate the outputs yielded by the reservoir simulator. Moreover, we coupled these dynamic proxies with PSO to conduct the optimization. From the results of optimization and comparative analysis, the dynamic proxies were able to yield optimal results close to simulator only within 3% error, but 3 times faster. This finding further highlights the significance of dynamic proxies in terms of

application. Although these proxies are case-dependent, they have excellently served their purpose of use in this study. Besides that, these summarized findings also confirm the cogency of the methodology used to establish these dynamic proxies. Finally, we also believe that there is still room for improvement of the methodology discussed in this paper. One of them includes the consideration of noise-handling ability as highlighted earlier. Besides that, the introduction of decision variables with higher dimensionality and the application of multi-objective optimization are parts of possible future studies. As the methodology achieves a satisfactory level of maturity, its potential use can later be extended to optimization of CO₂ storage and/or EOR.

Authors' contributions

Cuthbert Shang Wui Ng: Methodology, Problem Formulation, Data Analysis and Investigation, Modeling, Programming, Writing, Reviewing, Editing, Revising.

Ashkan Jahanbani Ghahfarokhi: Supervision, Methodology, Writing, Reviewing, Revising, and Editing.

Menad Nait Amar: Methodology, Writing, Reviewing, Editing and Revising.

Declaration of interest statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

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).

References

- [1] M.L. Wiggins, R.A. Startzman, An approach to reservoir management, SPE Repr. Ser. (1998), <https://doi.org/10.2118/20747-ms>.
- [2] L.W. Lake, R. Johns, B. Rossen, G.A. Pope, *Fundamentals of Enhanced Oil Recovery*, others, 2014.
- [3] B. Guyaguler, R.N. Horne, L. Rogers, J.J. Rosenzweig, Optimization of well placement in a gulf of Mexico waterflooding project, SPE Reservoir Eval. Eng. (2002), <https://doi.org/10.2118/78266-PA>.
- [4] A. Mamghaderi, A. Bastami, P. Pourafshary, Optimization of waterflooding performance in a layered reservoir using a combination of capacitance-resistive model and genetic algorithm method, J. Energy Resour. Technol. (2013), <https://doi.org/10.1115/1.4007767>.
- [5] J.L. Mogollón, T.M. Lokhandwala, E. Tillerio, New trends in waterflooding project optimization, SPE Lat. Am. Caribb. Pet. Eng. Conf. Proc. (2017), <https://doi.org/10.2118/185472-ms>.
- [6] A.J. Hong, R.B. Bratvold, G. Nævdal, Robust production optimization with capacitance-resistance model as precursor, Comput. Geosci. (2017), <https://doi.org/10.1007/s10596-017-9666-8>.
- [7] P. Ogbeiw, Y. Aladeitan, D. Udebhulu, An approach to waterflood optimization: case study of the reservoir X, J. Pet. Explor. Prod. Technol. (2018), <https://doi.org/10.1007/s13202-017-0368-5>.
- [8] S.S. Rao, *Engineering optimization: theory and practice*. <https://doi.org/10.1002/9781119454816>, 2019.
- [9] M.C. Bellout, D. Echeverría Ciaurri, L.J. Durlofsky, B. Foss, J. Kleppe, Joint optimization of oil well placement and controls, Comput. Geosci. (2012), <https://doi.org/10.1007/s10596-012-9303-5>.
- [10] X. Liu, A.C. Reynolds, Gradient-based multi-objective optimization with applications to waterflooding optimization, Comput. Geosci. 20 (2016), <https://doi.org/10.1007/s10596-015-9523-6>.
- [11] M. Al-Aghbari, M. Al-Wadhahi, A.M. Gujarathi, Multi-objective optimization of Brugge field for short-term and long-term waterflood management, Arabian J. Sci. Eng. (2021), <https://doi.org/10.1007/s13369-021-05614-7>.
- [12] S.D. Mohaghegh, Reservoir simulation and modeling based on artificial intelligence and data mining (AI&DM), J. Nat. Gas Sci. Eng. (2011), <https://doi.org/10.1016/j.jngse.2011.08.003>.
- [13] S.D. Mohaghegh, *Data-Driven Reservoir Modeling*, 2017.
- [14] S.D. Mohaghegh, S. Amini, V. Gholami, R. Gaskari, G. Bromhal, Grid-Based

- Surrogate Reservoir Modeling (SRM) for fast track analysis of numerical reservoir simulation models at the grid block level, Soc. Pet. Eng. West. Reg. Meet. (2012), <https://doi.org/10.2118/153844-ms>, 2012.
- [15] S.D. Mohaghegh, Quantifying uncertainties associated with reservoir simulation studies using surrogate reservoir models, Proc. SPE Annu. Tech. Conf. Exhib. (2006), <https://doi.org/10.2523/102492-ms>.
- [16] S.D. Mohaghegh, H. Hafez, R. Gaskari, M. Haajizadeh, M. Kenawy, Uncertainty analysis of a giant oil field in the middle east using surrogate reservoir model, in: 12th Abu Dhabi Int. Pet. Exhib. Conf. ADIPEC 2006 Meet. Increasing Oil Gas Demand through Innov, 2006, <https://doi.org/10.2523/101474-ms>.
- [17] G. Vida, M.D. Shahab, M. Mohammad, Smart proxy modeling of SACROC CO₂-EOR, Fluids (2019), <https://doi.org/10.3390/fluids4020085>.
- [18] A. Shahkarami, S. Mohaghegh, Applications of smart proxies for subsurface modeling, Petrol. Explor. Dev. (2020), [https://doi.org/10.1016/S1876-3804\(20\)60057-X](https://doi.org/10.1016/S1876-3804(20)60057-X).
- [19] A. Shahkarami, S.D. Mohaghegh, V. Gholami, S.A. Haghghat, Artificial intelligence (AI) assisted history matching, in: Soc. Pet. Eng. SPE West. North Am. Rocky Mt. Jt. Meet, 2014, <https://doi.org/10.2118/169507-ms>.
- [20] Q. He, S.D. Mohaghegh, Z. Liu, Reservoir simulation using smart proxy in SACROC unit - case study, in: SPE East. Reg. Meet, 2016, <https://doi.org/10.2118/184069-MS>.
- [21] F. Alenezi, S. Mohaghegh, Developing a smart proxy for the SACROC waterflood numerical reservoir simulation model, in: SPE West. Reg. Meet. Proc., 2017, <https://doi.org/10.2118/185691-ms>.
- [22] J. Jalali, S.D. Mohaghegh, Reservoir simulation and uncertainty analysis of enhanced CBM production using artificial neural networks, in: SPE East. Reg. Meet, 2009, <https://doi.org/10.2118/125959-ms>.
- [23] A. Kalantari-Dahaghi, S.D. Mohaghegh, A new practical approach in modelling and simulation of shale gas reservoirs: application to New Albany Shale, Int. J. Oil Gas Coal Technol. (2011), <https://doi.org/10.1504/IJOGCT.2011.038925>.
- [24] M. Nait Amar, N. Zeraibi, K. Redouane, Optimization of WAG process using dynamic proxy, genetic algorithm and ant colony optimization, Arabian J. Sci. Eng. (2018), <https://doi.org/10.1007/s13369-018-3173-7>.
- [25] N.A. Menad, Z. Noureddine, An efficient methodology for multi-objective optimization of water alternating CO₂ EOR process, J. Taiwan Inst. Chem. Eng. 99 (2019) 154–165, <https://doi.org/10.1016/j.jtice.2019.03.016>.
- [26] J. Kim, H. Yang, J. Choe, Robust optimization of the locations and types of multiple wells using CNN based proxy models, J. Petrol. Sci. Eng. 193 (2020), 107424, <https://doi.org/10.1016/j.petrol.2020.107424>.
- [27] J. Kim, K. Lee, J. Choe, Efficient and robust optimization for well patterns using a PSO algorithm with a CNN-based proxy model, J. Petrol. Sci. Eng. 207 (2021), 109088, <https://doi.org/10.1016/j.petrol.2021.109088>.
- [28] L. Deng, Y. Pan, Data-driven proxy model for waterflood performance prediction and optimization using Echo State Network with Teacher Forcing in mature fields, J. Petrol. Sci. Eng. 197 (2021), 107981, <https://doi.org/10.1016/j.petrol.2020.107981>.
- [29] C.S.W. Ng, A. Jahanbani Ghahfarokhi, M. Nait Amar, O. Torsæter, Smart proxy modeling of a fractured reservoir model for production optimization: implementation of metaheuristic algorithm and probabilistic application, Nat. Resour. Res. 30 (2021) 2431–2462, <https://doi.org/10.1007/s11053-021-09844-2>.
- [30] M. Nait Amar, A. Jahanbani Ghahfarokhi, C.S.W. Ng, N. Zeraibi, Optimization of WAG in real geological field using rigorous soft computing techniques and nature-inspired algorithms, J. Petrol. Sci. Eng. (2021), 109038, <https://doi.org/10.1016/j.petrol.2021.109038>.
- [31] C.S.W. Ng, A. Jahanbani Ghahfarokhi, M. Nait Amar, Application of nature-inspired algorithms and artificial neural network in waterflooding well control optimization, J. Pet. Explor. Prod. Technol. (2021), <https://doi.org/10.1007/s13202-021-01199-x>.
- [32] S.H. Yousefi, F. Rashidi, M. Sharifi, M. Soroush, A.J. Ghahfarokhi, Interwell connectivity identification in immiscible gas-oil systems using statistical method and modified capacitance-resistance model: a comparative study, J. Petrol. Sci. Eng. 198 (2021), 108175, <https://doi.org/10.1016/j.petrol.2020.108175>.
- [33] M. Nait Amar, N. Zeraibi, A. Jahanbani Ghahfarokhi, Applying hybrid support vector regression and genetic algorithm to water alternating CO₂ gas EOR, Greenh. Gases Sci. Technol. (2020), <https://doi.org/10.1002/ghg.1982>.
- [34] M. Talebkeikhah, M. Nait Amar, A. Naseri, M. Humand, A. Hemmati-Sarapardeh, B. Dabir, M.E.A. Ben Seghier, Experimental measurement and compositional modeling of crude oil viscosity at reservoir conditions, J. Taiwan Inst. Chem. Eng. 109 (2020) 35–50, <https://doi.org/10.1016/j.jtice.2020.03.001>.
- [35] M. Nait Amar, A. Jahanbani Ghahfarokhi, N. Zeraibi, Predicting thermal conductivity of carbon dioxide using group of data-driven models, J. Taiwan Inst. Chem. Eng. 113 (2020) 165–177, <https://doi.org/10.1016/j.jtice.2020.08.001>.
- [36] H. Mehrjoo, M. Riazi, M. Nait Amar, A. Hemmati-Sarapardeh, Modeling interfacial tension of methane-brine systems at high pressure and high salinity conditions, J. Taiwan Inst. Chem. Eng. 114 (2020) 125–141, <https://doi.org/10.1016/j.jtice.2020.09.014>.
- [37] M. Nait Amar, M.A. Ghrga, A. Hemmati-Sarapardeh, Application of gene expression programming for predicting density of binary and ternary mixtures of ionic liquids and molecular solvents, J. Taiwan Inst. Chem. Eng. 117 (2020) 63–74, <https://doi.org/10.1016/j.jtice.2020.11.029>.
- [38] M.Z. Alom, T.M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M.S. Nasrin, M. Hasan, B.C. Van Essen, A.A.S. Awwal, V.K. Asari, A state-of-the-art survey on deep learning theory and architectures, Electron 8 (2019), <https://doi.org/10.3390/electronics8030292>.
- [39] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780. <http://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf>.
- [40] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE Int. Conf. Neural Networks Conf. Proc, 1995, <https://doi.org/10.4018/ijmfmfp.2015010104>.
- [41] Y. Shi, R. Eberhart, Modified particle swarm optimizer, in: Proc. IEEE Conf. Evol. Comput. ICEC, 1998, <https://doi.org/10.1109/icec.1998.699146>.
- [42] K.L. Du, M.N.S. Swamy, Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature, Springer International Publishing, 2016, <https://doi.org/10.1007/978-3-319-41192-7>.
- [43] A.E. Ezugwu, O.J. Adeleke, A.A. Akinyelu, S. Viriri, A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems, Neural Comput. Appl. 32 (2020), <https://doi.org/10.1007/s00521-019-04132-w>.
- [44] C.S.W. Ng, A. Jahanbani Ghahfarokhi, M. Nait Amar, Well production forecast in Volve field: application of rigorous machine learning techniques and metaheuristic algorithm, J. Petrol. Sci. Eng. 208 (2022), 109468, <https://doi.org/10.1016/j.petrol.2021.109468>.
- [45] J.D. Jansen, R.M. Fonseca, S. Kahrobaei, M.M. Siraj, G.M. Van Essen, P.M.J. Van den Hof, The egg model - a geological ensemble for reservoir simulation, Geosci. Data J. (2014), <https://doi.org/10.1002/gdj3.21>.
- [46] M.D. McKay, R.J. Beckman, W.J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics (1979), <https://doi.org/10.2307/1268522>.
- [47] J.M. Hammersley, D.C. Handscomb, Monte Carlo methods. <https://doi.org/10.1007/978-94-009-5819-7>, 1964.
- [48] I.M. Sobol, On the distribution of points in a cube and the approximate evaluation of integrals, USSR Comput. Math. Math. Phys. (1967), [https://doi.org/10.1016/0041-5553\(67\)90144-9](https://doi.org/10.1016/0041-5553(67)90144-9).
- [49] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, in: 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., 2015.