Proceedings of the 32nd European Safety and Reliability Conference (ESREL 2022)
Edited by Maria Chiara Leva, Edoardo Patelli, Luca Podofillini, and Simon Wilson
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-18-5183-4_S01-12-623-cd

Long Short-Term Memory ANNs for Fast Assessment of Injection Policies of Water-Flooding in Oil and Gas Reservoirs

Guido Di Federico

Energy Department, Politecnico di Milano, Italy. E-mail: guido.difederico@mail.polimi.it

Ahmed Shokry

Center for Applied Mathematics, Ecole Polytechnique, France. E-mail: ahmed.shokry@polytechnique.edu

Enrico Zio

Energy Department, Politecnico di Milano, Italy & MINES Paris PSL, France. E-mail:enrico.zio@polimi.it; enrico.zio@mines-paristech.fr

Giorgio Fighera

Eni S.p.A., Italy. E-mail: Giorgio.Fighera@eni.com

Emanuele Vignati

Eni S.p.A., Italy. E-mail: emanuele.vignati@eni.com

Models of the reservoirs are used in almost all phases of oil field management. The development of a physics-based model of a reservoir is an effort-intensive and time-consuming task. Even when available, the use of such a model can be computationally expensive. Thus, an emerging research direction is focusing on the use of Machine Learning (ML) to develop problem-specific reservoir models using real field data or datasets generated by model-based simulation. Long Short-Term Memory (LSTM) networks have been recently used due to their capabilities to capture temporal and spatial patterns from multivariate time series. However, they have been applied mainly to small oil fields and/or using simulation datasets with unrealistic characteristics. This paper investigates the efficient use of LSTMs networks to develop dynamic data-driven reservoir models. A procedure for simulating the complex physics-based reservoir model in order to generate input-output datasets of realistic characteristics is presented. The data are, then, used to train an ensemble of LSTM models, each one able to predict the future value of the oil or the water production rate at one production well, as a function of all the water injection rates at all the injection wells. These models are useful for the fast assessment of waterflooding policies with respect to their economic impact. The proposed framework is applied to the Olympus field benchmark case study showing good prediction accuracy and significant reduction in computational time with respect to the reservoir physics-based model.

Keywords: oil and gas, waterflooding, reservoir models, machine learning, Long Short-Term Memory networks

1. Introduction

Many of the existing large oil fields have reached their mature stage of production (BP, 2020), where the natural pressure of the reservoir is no longer sufficient to displace the oil. In such situations, waterflooding technologies are usually applied, in which the water is pumped at high pressure into the subsurface through injection wells strategically positioned across the field, to displace oil and improve its recovery (Novelli et al., 2005). In turn, production wells produce a multiphase mixture at the well-head, whose main components are usually oil and water. Oil displacement through waterflooding is a slow process: its timescale is in the order of months to

years. Hence, also for operational reasons, variations in production strategies cannot be applied frequently, but are modified only every so often (Novelli et al., 2005).

Selecting the water injection profiles (i.e., policy) to be applied in the field through the injection wells over a specific production horizon is critical. A poor water injection policy can lead to a high risk of violating the minimum economic oil production rate or the maximum water production rate allowed by the facilities, ultimately leading to well flooding and shutdown (Novelli et al., 2005). On the contrary, an optimal water injection plan has the potential to maximize oil recovery from the reservoir of interest, decrease production cost, minimize waste of resources, and mitigate environmental hazards such as oil leaks and water contamination (Koroteev and Tekic, 2021).

2. Machine Learning for Waterflooding Management

Traditionally, waterflooding management is done through surveillance methods, in which policies are designed relying on the expertise of field engineers and operators for the analysis of the field real-time production data. Although these methods are relatively easy and cheap to implement, they provide reactive policies that consider only the recent production conditions over very short time periods, with limited ability to foresee the future evolving conditions of the reservoir. This limitation stems from the fact that these methods lack any knowledge about the dynamics of the reservoir. Hence, the availability of a reliable model for the reservoir able to accurately capture the relationships among the different control (e.g., water injections, and bottom hole pressures (BHPs)) and output variables (e.g., production rates), and to predict their dynamic evolution over long periods of time, can assist the design of efficient and sustainable water injection policies.

In more details, for waterflooding management purposes, a reservoir model should describe the dynamic relation among the injection rates (control variables) applied through the injection wells at a certain time instance, and the output variables represented by the water and oil production rates obtained from the production wells. Such model usually includes equations of state, multiphase flow, mass conservation, and deliverability equations (Dake, 1978). The model is usually solved by means of finite difference techniques, in which the field is discretized as a 3D grid with given geological properties and, then, the aforementioned equations are applied to each grid/cell. The resulting system of non-linear PDEs is then solved numerically to obtain pressure, velocity and saturation distributions. In this way, a geological model paired to a simulator allows to map the dynamic relationships among the production rates, injection rates and state variables.

Due to the complex and non-linear nature of the relationships embedded in a 3D reservoir model, and the demanding numerical techniques required for its solution, its usage can be computationally heavy, especially in applications requiring online and/or lots of simulation runs, such as the one addressed in this work, i.e., water injection policies assessment. To address this problem, ML techniques are being considered to model reservoir behavior (Koroteev and Tekic 2021), due to their potentially low computational cost and independence of any geological or physical knowledge. The ML models can be constructed either using data generated from the simulation of a reservoir model or real data collected from the field.

Among many ML models, LSTM artificial neural networks (ANNs), which are a type of recurrent ANNs (R-ANNs), are of particular interest because of their capabilities to capture temporal and spatial relationship patterns among multivariate time-series of data (Géron, 2019). Kim and Durlofsky (2021) used LSTM ANNs for oil production forecasting using data generated from 2-D and 3-D synthetic reservoir models including two injection and five production wells. The network is used to predict the future values of the oil and water production rates at each well as a function of injection rates and BHPs. Similarly, Deng and Pan (2020) applied echo state networks (ESNs) to approximate two case studies, including a 2-D synthetic reservoir model with one injection and four production wells, and a 3-D synthetic model with eight injectors and four producers. The ESNs were trained to receive water injection rates and BHPs as input, and provide water and oil production rates as output, by means of a fractional flow model to compute

single-phase production rates from the total production rate. Tang et al. (2021) applied LSTM ANNs to approximate a simple 3-D synthetic field model with two injectors and three producers, where injection rates and oil saturation distributions were the inputs, whereas the oil and water production rates were the outputs. Ng et al. (2022) employed LSTM ANNs to approximate the egg-model benchmark that includes eight injectors and four producers, where the LSTM ANN is used to forecast the total liquid production rate as a function of the injection rates combined with permeability measurements.

To train the LSTM ANNs, in these works several profiles of the production times-series data along the field history have been generated by simulating a reservoir model under different control and operating scenarios. But in real situations, a unique time profile of the production data is available, which corresponds to the "actual" operating history of the field. Only the work of Rodriguez and Salazar (2022) has considered this realistic constraint in the reservoir model simulation for training and testing data generation. This was done for a simple case involving a highly homogeneous synthetic field with only one injector and four producers.

The novelty of the framework proposed in this paper lies in:

i) considering realistic bounds on the data availability during the reservoir model simulation, thus using only one profile of production data representing the actual operating history of the field. For this purpose, a procedure for data generation is designed, which imposes tight, practical and realistic constraints on the reservoir simulations to generate the training and testing data.
ii) using only production data for training the ML models, while no geological information or fractional flow model is considered.
iii) analyzing the prediction uncertainty of the LSTM ANNs-based reservoir models using a bagging-based procedure.

3. Proposed Framework

3.1. Data collection/generation

We consider a generic field made up of N_{in} injection wells and N_p production wells. The time span of the whole production history is discretized into N_t equal time steps. The vector of injection rates can be represented as $\boldsymbol{u_{(k)}} = [q_{(k)}^{in,1}, \dots, q_{(k)}^{in,N_{in}}]^{\boldsymbol{T}} \in \mathbb{R}^{N_{in}}$ with allowable upper and lower bounds $\boldsymbol{u}^{max}, \boldsymbol{u}^{min}$, and the vectors of water and oil production rates as $\boldsymbol{q}_{(k)}^{w} = [q_{(k)}^{w,j}, \dots, q_{(k)}^{w,N_p}]^T$ and $\boldsymbol{q}_{(k)}^{o} = [q_{(k)}^{o,j}, \dots, q_{(k)}^{o,N_p}]^T \in \mathbb{R}^{N_p}$, where $k = 1, \dots, N_t$ is the generic time step.

As previously mentioned, since real field monitoring data are not available, we follow a specific simulation procedure to ensure that the dataset generated from a physics-based reservoir model possesses minimum realistic characteristics, such that:

- each well has a single injection or production profile over time. In other words, only one scenario (of injection and production rates) can be generated to represent the available data collected along the actual operation history of the field;
- injection profiles are not randomly generated, but time-correlated.

The first point represents the fact that the available data is intrinsically limited to the historical production strategy of the field itself. As such, historical data may not cover the complete range of acceptable values for controls and parameters. The second point ensures that historical data changes smoothly over time.

Injection profile generation is based on the formulation provided by Chilès (2012). The idea is to generate smoother injection profiles with respect to a random initialization, by imposing a correlation, i.e., the closer two points are in the variable space, the closer their values are. Time-correlated injection profiles express the need to gradually change injection rates for the same well at successive time steps. Space-correlated injection profiles account for the need to avoid injecting from neighbor wells disproportionately at the same time step. For simplicity, only time correlation is considered.

Mathematically, correlation can be obtained from a random profile multiplied by the Cholesky lower triangular of the correlation matrix $\boldsymbol{C}$, centering it around an average. $\boldsymbol{C}$ is a function of a variogram form γ (cubic in this case), variance σ and bound values for all wells $\boldsymbol{u}^{max}, \boldsymbol{u}^{min}$, such that $\boldsymbol{C} = \boldsymbol{C}(\gamma, \sigma, \boldsymbol{u}^{max}, \boldsymbol{u}^{min})$.

Then, once input signals are generated, they are simulated by the reservoir simulation software

F^{3D} (representing in this case the real field, or "ground truth") to obtain the output signal $\boldsymbol{q}^{w}_{(k)}$, $\boldsymbol{q}^{o}_{(k)}$ as a function of control variables $\boldsymbol{u}_{(k)}$ and state variables $\boldsymbol{x}_{(k)}$ at the previous and present time steps, i.e. $\boldsymbol{q}^{w}_{(k)}$, $\boldsymbol{q}^{o}_{(k)} = F^{3D}(\boldsymbol{x}_{(k-1)}, \boldsymbol{x}_{(k)}, \boldsymbol{u}_{(k)})$.

3.2. Long Short-Term Memory ANNs

LSTM ANN is a type of R-ANN, that has been proven to successfully tackle the long-term gradient vanishing problem of traditional R-ANNs, showing better overall performance such as faster training and better ability to detect long-term temporal and spatial patterns in time-series data (Hochreiter and Schmidhuber, 1997).

A scheme of a LSTM cell is reported in Fig. 1. At a generic time-step k, a LSTM cell predicts the approximated output $\hat{\boldsymbol{y}}_{(k)}$ as a function of the input $\boldsymbol{u}_{(k)}$ and previous states that the network temporarily stores, namely the hidden or the short-term state $\boldsymbol{h}_{(k-1)}$, and the cell or the long-term state $\boldsymbol{c}_{(k-1)}$.

A LSTM cell is composed by three processing units, which include input gate, forget gate, and output gate. The forget gate processes the input $\boldsymbol{u}_{(k)}$ and the previous hidden state $\boldsymbol{h}_{(k-1)}$ to provide the forget gate's activation vector $\boldsymbol{f}_{(k)} \in [0,1]$ that includes values controlling which element in previous cell state $\boldsymbol{c}_{(k-1)}$ will be forgotten, as in Eq. (1), where $\boldsymbol{W}_f$, $\boldsymbol{R}_f$ and $\boldsymbol{b}_f$ are trainable weight matrixes and biases.

$$\boldsymbol{f}_{(k)} = \boldsymbol{\sigma}_g(\boldsymbol{W}_f\, \boldsymbol{u}_{(k)} + \boldsymbol{R}_f \boldsymbol{h}_{(k-1)} + \boldsymbol{b}_f) \quad (1)$$

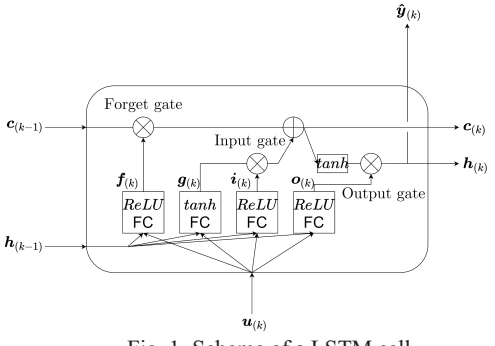

Fig. 1. Scheme of a LSTM cell.

Then, the input gate performs the update of the previous cell state $\boldsymbol{c}_{(k-1)}$ into the current cell state $\boldsymbol{c}_{(k)}$ through three steps:

i) selecting the new information to be added to the cell state by processing the input $\boldsymbol{u}_{(k)}$ and the previous hidden state $\boldsymbol{h}_{(k-1)}$ (as in Eq. (2)) outputting input/update gate's activation vector $\boldsymbol{i}_{(k)} \in [0,1]$:

$$\boldsymbol{i}_{(k)} = \boldsymbol{\sigma}_i(\boldsymbol{W}_i\, \boldsymbol{u}_{(k)} + \boldsymbol{R}_i \boldsymbol{h}_{(k-1)} + \boldsymbol{b}_i) \quad (2)$$

ii) computing the potential vector of cell state $\boldsymbol{g}_{(k)} \in [-1,1]$, as follows:

$$\boldsymbol{g}_{(k)} = tanh(\boldsymbol{W}_g\, \boldsymbol{u}_{(k)} + \boldsymbol{R}_g \boldsymbol{h}_{(k-1)} + \boldsymbol{b}_g) \quad (3)$$

iii) updating the previous cell state $\boldsymbol{c}_{(k-1)}$ into the current cell state $\boldsymbol{c}_{(k)}$, as in Eq. (4), where $\otimes$ is an element-wise multiplication operator:

$$\boldsymbol{c}_{(k)} = \boldsymbol{f}_{(k)} \otimes \boldsymbol{c}_{(k-1)} + \boldsymbol{i}_{(k)} \otimes \boldsymbol{g}_{(k)} \quad (4)$$

Finally, the output gate computes the final output through two steps:

i) the relevant portion of the cell state to be utilized is selected as follows:

$$\boldsymbol{o}_{(k)} = \boldsymbol{\sigma}_o(\boldsymbol{W}_o\, \boldsymbol{u}_{(k)} + \boldsymbol{R}_o \boldsymbol{h}_{(k-1)} + \boldsymbol{b}_o) \quad (5)$$

ii) The output is calculated as follows:

$$\boldsymbol{h}_{(k)} = \boldsymbol{o}_{(k)} \otimes tanh(\boldsymbol{c}_{(k)}) \quad (6)$$

Cells are packed into layers, which make up the overall network. A final dense layer processes the cell states and produces the approximated output of interest for the network, $\hat{\boldsymbol{q}}^{w}_{(k)}$ or $\hat{\boldsymbol{q}}^{o}_{(k)}$, from $\hat{\boldsymbol{y}}_{(k)}$. A LSTM ANN can predict the output over more than one step ahead in the future, considering the input values over a certain number of past time steps (known as input sliding window), which can range from single to several time steps in the past. In this study, the output is predicted over one step ahead in the future, whereas the size of the input sliding window is determined during the network hyperparameter selection process.

4. Case study: Olympus field

The Olympus benchmark is a reservoir physics-based model inspired by an oilfield in the North Sea (ISAPP website, 2017). The field consists of 118×181×16 grid blocks of dimension 50×50×3 m and is bounded on one side by a boundary fault. Six minor geological faults are also present in the field model. Fig. *2* shows the permeability distribution for the Olympus field along with the locations of seven injection points, I1:I7, and eleven production wells, P1:P11. Table 1 reports the geological property ranges in different facies (rock types).

Notice that the top zone contains high permeability channels embedded in a low

permeability matrix, whereas the bottom zone has overall lower permeability. The two zones are separated by an impermeable shale layer. Overall, there are four different types of facies.

Table 1. Geological properties of the Olympus field.

Facies	Zone	Porosity	Perm. [mD]
Channel sand	Top	0.2-0.35	400-1000
Shale	Top, barrier	0.03	1
Coarse sand	Bottom	0.2-0.3	150-400
Fine sand	Bottom	0.1-0.2	75-150
Fine sand	Bottom	0.05-0.1	10-50

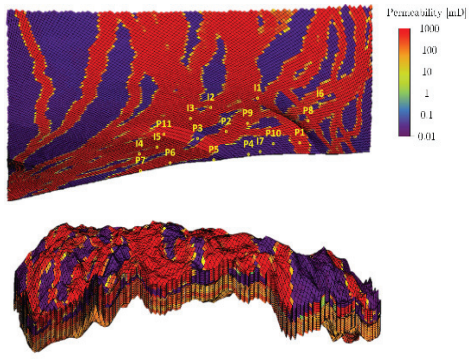

Fig. 2. Geological model of the Olympus field with permeability distribution and wells location.

4.1. Applications and Results

The field operating period of 20 years is discretized into constant and equal timesteps, each of 30 days. Then, the proposed data generation procedure is used to design seven water injection profiles (over the operating period of 20 years) for the seven injection wells, where the injection rates are not allowed to change over less than three consecutive time steps. The designed control policy/scenario is, then, simulated using the reservoir physics-based model to obtain the corresponding 22 output signals (water and oil production rates at each of the eleven production wells). The time required to obtain the dataset is about 60 minutes using Eclipse® commercial simulator.

The dataset is split as follows: the first 14 years (170 data points) are used as a training subset, the next four years (50 data points) as validation subset and the remaining two years (24 data points) as a testing subset that represents the future period to be forecasted. This is also equivalent to splitting ratios of 70%, 20% and 10% for training, validation and testing, respectively. Fig. 3 shows the seven water injection rates (a), eleven water production rates (b), and the eleven oil production rates (c). The solid, dashed, and dotted lines distinguish the training, validation and testing data subsets, respectively. When the injected water first comes out at production wells, the so-called breakthrough has occurred. From this point on, the water fraction usually rises, as the reservoir is being gradually flooded and oil swept (Dake, 1978): this can be noticed in Figure 3 (b-c).

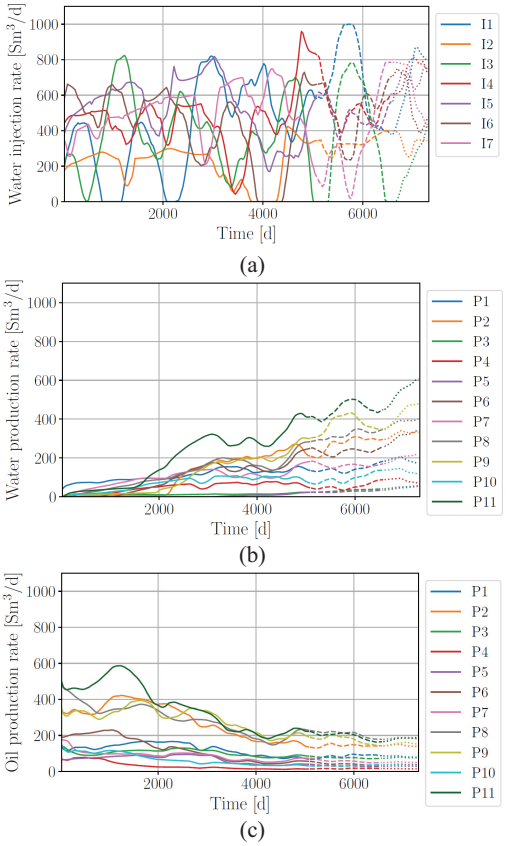

Fig. 3. Profiles of the water injection rates (a), water production rates (b), and oil production rates (c) in the generated dataset.

For the j-th production well, two independent LSTM networks are trained: one network $F^{LSTM,w,j}$ predicts the water production rate whereas the other $F^{LSTM,o,j}$ predicts the oil production rate as a function of the injection schedule at all the seven injection wells, i.e., $\hat{q}^{o,j}_{(k)} = F^{LSTM,o,j}(\theta, \boldsymbol{u}_{(k)}, \ldots, \boldsymbol{u}_{(k-l)})$ and $\hat{q}^{w,j}_{(k)} = F^{LSTM,wj}(\theta, \boldsymbol{u}_{(k)}, \ldots, \boldsymbol{u}_{(k-l)})$, where l is the number of previous steps used for the prediction and θ is the set of trainable parameters of the networks, i.e. weights and biases.

The Adam optimizer is used as the training algorithm (Géron, 2019). The hyperparameters for the networks are chosen based on a sensitivity analysis carried out using all combinations of the following values: 1 or 2 hidden layers, 10 or 20 neurons per layer, 3 or 5 past time steps, l, learning rate equal to 10^{-4}, 10^{-3} or 10^{-2}, L1 regularization set to 0, 10^{-2} or 10^{-1}, and L2 regularization to 10^{-3}, 10^{-2} or 10^{-1}. The training of each combination is performed in parallel on Eni's High Performance Computing (HPC) system, using 12 CPU units and 1 GPU unit for each combination. The combinations are tested in parallel, and the best model for each well is selected. The average training time for a single hyperparameter combination (i.e., the training of one LSTM ANN) is about 10 minutes per well for oil and 10 minutes per well for water.

Fig. 4 and Fig. 5 show the prediction NRMSE at each production well, for water and oil, respectively. The NRMSE is given for the training, validation and testing subsets, and calculated by dividing the RMSE by an overestimated variability range of the whole training and validation subsets.

Considering the challenging characteristics of the generated datasets (see Section 3.1), the 22 LSTMs ANNs achieved satisfactory performance. Also importantly, the ANNs achieved almost 100 times reduction in the computational cost required for prediction at one single time-step (about 1.2s) compared to the physics-based reservoir model embedded in the Eclipse® commercial simulator. Fig. *6* shows, for example, the predicted oil and water production rates over the validation and testing subsets, at three different wells, and compares them to the exact values (black dotted lines).

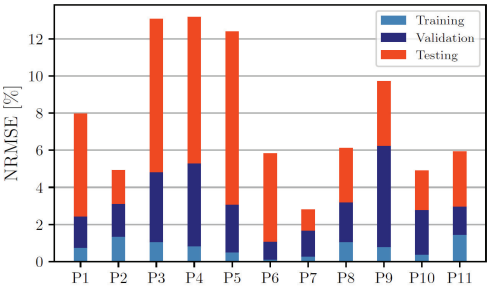

Fig. 4. NRMSE [%] of the water-forecasting LSTMs.

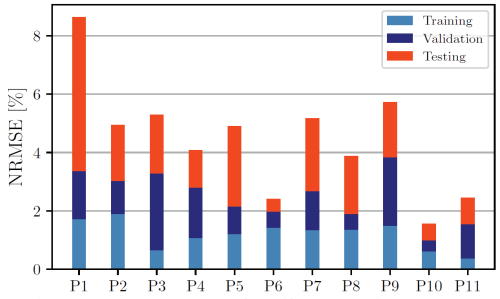

Fig. 5. NRMSE [%] of the oil-forecasting LSTMs.

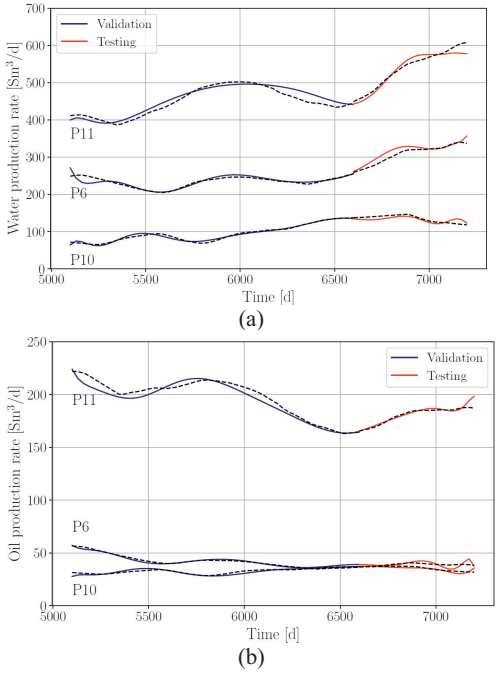

Fig. 6. Water (a) and oil (b) production rates predicted by the LSTMs over the validation and testing subsets at three wells.

4.1.1. Uncertainty Quantification of the LSTM ANNs Prediction

A straightforward way to quantify the prediction uncertainty of a ML model is bagging. The main idea is to obtain an ensemble of ML models, by fitting the model multiple times using the same dataset but changing the splitting ratios of the training and validation subsets, while maintaining the testing subset constant. In this study, 90% of data (union of the training and validation subsets, which are also equivalent to the first 18 years of the field operation) is randomly split 10 times between 65-80% for training and 25-10% for validation. Notice that, due to the recurrent nature of the LSTM networks, the different 10 splits must respect the time sequence of the data.

We show the application results of this procedure for one LSTM ANN, $F^{LSTM,o,11}$, which predicts the oil production rate at well P11: the highest oil-producing well over the field history.

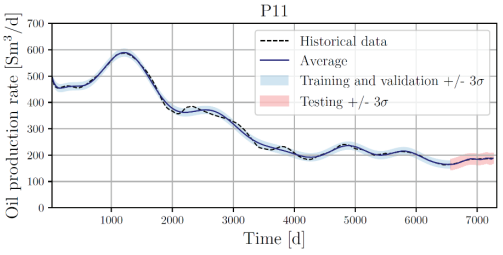

Fig. 7 shows the mean of the prediction $\mp 3\,\sigma$ of the 10 LSTM ANNs, where σ is the standard deviation of the prediction. The average and standard deviation NRMSE of the prediction of the 10 LSTM ANNs in the training/validation and testing are 1.66% and 1.26%, respectively, which correspond to 7 and 9 Sm^3.

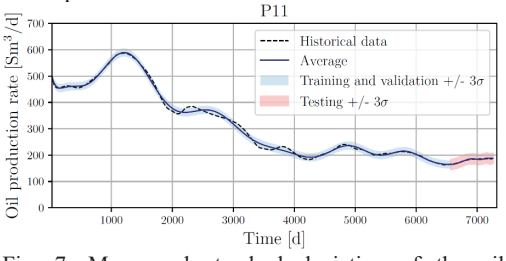

Fig. 7. Mean and standard deviation of the oil production rate prediction at well P11, using the LSTM ANNs ensemble.

5. Conclusions

This work proposes an efficient framework for developing ML-based reservoir models using LSTM ANNs. Such models can be used for fast assessment and design of water injection policies in mature oil fields. A procedure is adopted for simulating complex physics-based reservoir models to generate input (water injection rates) and output (water and oil production rates) signals possessing minimum realistic characteristics. Then, an ensemble of LSTM ANNs is developed, each one able to predict the future value of the water or oil production rates at each production well. A bagging-based procedure is adopted for estimating the prediction uncertainty of the LSTMs ANNs. The framework has been applied to the Olympus reservoir model benchmark, showing effectiveness in terms of: i) good prediction accuracy, even when using datasets with real, challenging characteristics, such as the limited number of training patterns, and the few available information on the dynamic conditions of the field, ii) significant reduction in the

required computational cost compared to a physics-based reservoir model, and iii) good ability to estimate the LSTM ANNs prediction uncertainty.

Acknowledgement

This work was sponsored and supported by Eni S.p.A.

References

BP. (2020). "World Energy Outlook." https://www.bp.com/en/global/corporate/%0Aenergy-economics/energy-outlook.html.

Chilès, J.P. (2012) *Geostatistics: Modeling Spatial Uncertainty*. Wiley.

Dake, L.P. (1978) *Fundamentals of Reservoir Engineering,Volume 8*. Developments in Petroleum Science.

Deng, L. and Pan, Y. (2020) "Machine-Learning-Assisted Closed-Loop Reservoir Management Using Echo State Network for Mature Fields under Waterflood." *SPE Reservoir Evaluation and Engineering* 23 (4): 1298–1313.

Géron, A. (2019) *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

Hochreiter, S. and Schmidhuber J. (1997) "Long Short-Term Memory." *Neural Computation* 9 (8): 1735–80.

ISAPP website (2017). "Olympus Field Model.". https://www.isapp2.com/downloads/olympus-reservoir-model.pdf.

Kim, Y. D., and Durlofsky, L. (2021) "A Recurrent Neural Network-Based Proxy Model for Well-Control Optimization with Nonlinear Output Constraints." *SPE Journal* 26 (4): 1837–57.

Koroteev, D. and Tekic, Z. (2021) "Artificial Intelligence in Oil and Gas Upstream: Trends, Challenges, and Scenarios for the Future." *Energy and AI* 3: 100041.

Ng, C. S. W. et al. (2022) "Production Optimization under Waterflooding with Long Short-Term Memory and Metaheuristic Algorithm." *Petroleum*.

Novelli, L. et al. (2005). *Hydrocarbons: Origin, Exploration and Production*. Eni Corporate University.

Rodriguez, A. X. and Salazar, D. A. (2022) "Methodology for the Prediction of Fluid Production in the Waterflooding Process Based on Multivariate Long–Short Term Memory Neural Networks." *Journal of Petroleum Science and Engineering* 208 (PE): 109715.

Tang, L. et al. (2021) "Well Control Optimization of Waterflooding Oilfield Based on Deep Neural Network." *Geofluids* 2021 (4): 1-15.