Computational Geosciences (2022) 26:1065–1100
https://doi.org/10.1007/s10596-022-10146-6

ORIGINAL PAPER

Physics-constrained deep learning forecasting: an application with capacitance resistive model

Abderrahmane Yewgat[1,2] · **Daniel Busby**[1] · **Max Chevalier**[2] · **Corentin Lapeyre**[3] · **Olivier Teste**[2]

Received: 20 January 2021 / Accepted: 29 March 2022 / Published online: 10 May 2022

Abstract

It is well known that the construction of traditional reservoir simulation models can be very time and resources consuming. Particularly in the case of mature fields with long history and large number of wells where such models can be extremely difficult and long to history match. In this case data driven models can represent a cost-effective alternative, or they can provide complementary analysis to classical reservoir modelling. Due to data scarcity full machine learning approaches are also usually doomed to fail. In this work we develop a new Physics-Constrained Deep Learning approach that combined neural networks with a reduced physics approach: Capacitance Resistive Model (CRM). CRM are data-driven methods that are based on a simple material balance approximation, that can provide very useful reservoir insight. CRM can be used to analyze the underlying connections between producer wells and injector wells that can then be used to better allocate water injection. Such analysis can usually require very long tracer tests or very expensive 4D seismic acquisition and interpretation. CRM can provide directly these wells connection information using only available production and pressure data. The problem with CRM approaches, based on classical optimizers, is that they often detect spurious correlations and can be not very robust and reliable. Our physics-constrained deep learning approach called Deep-CRM performs production data regularization via the neural network approximation that helps to provide a better CRM parameter identification also with the use of robust gradient descent optimization methods developed and widely used by the large deep learning community. We show first on a synthetic and then in real reservoir case that Deep-CRM was able to identify most of the injector-producer connections with higher accuracy with respect to traditional CRM. Deep-CRM produced also better liquid production forecasts on the performed blind tests.

Keywords Deep learning · Subsurface physics

1 Introduction

Capacitance Resistive Models (CRM) [1] are material balance models. They are based on a set of coupled ordinary differential equations (ODE) describing the material balance. CRM aim to predict liquid rate in a reservoir using only dynamic data of production liquid rates, water injections and Bottom Hole Pressure (BHP). In addition, CRM can explain the underlying connectivity between several injectors and producers that could be a valuable information for dynamic synthesis and for better understanding of fluid flows in the reservoir. Current work on CRM is done in three steps. The first step consists in solving analytically the CRM ODE, which is easily done using the variations of constants method. The previous step gives a closed form solution depending only on CRM parameters. The second step is to solve the nonlinear inverse problem on CRM parameters using a nonlinear multivariate regression based on the closed form solution and available observations data. This step is generally done using a classical optimizer such as SLSQP https://docs.scipy.org/doc/scipy/reference/optimize.minimize-slsqp.html and L-BFGS-B https://docs.scipy.org/doc/scipy/reference/optimize.minimize-lbfgsb.html. However these optimizers highly depend on a proper choice of the initialization to avoid local minima (Holanda et al [16]). Moreover, these optimizers compute approximate numerical gradients which may suffer from stability problems. The last step consists in forecasting liquid rate production using optimized CRM' parameters and CRM ODE' analytical solution.

✉ Abderrahmane Yewgat
Abderrahmane.yewgat@totalenergies.com

Extended author information available on the last page of the article.

To address the limitations of previous works on CRM, in this work we take a different approach to optimize CRM parameters. First, instead of relying on the analytical solution of CRM ODE, we learn a general solution based solely on the data. For that we use Deep Learning more specifically Artificial Neural Networks (ANNs) [17]. ANNs give a highly differentiable cost function, and an exact gradient computation approach, thanks to ANNs' automatic differentiation property [18] which is more stable than numerical approximation methods [18].

Second, we constrain the previous solution on the CRM ODE using the new theory of Physics-Informed Neural Networks (PINNs) [10, 11]. PINNs' framework gives us the ability to integrate physical prior knowledge such as ODE or PDE in our ANNs' model. Finally, during the optimization process our approach optimizes at the same time the deep learning model's parameters (weights) to match the data plus the CRM parameters to satisfy the CRM model.

The aim of this work is to define a complete and new approach to optimize CRM parameters and forecast total production rates. In a real field, producers' BHP are not always available due to sensors problems. In this work we show that our approach can simultaneously forecast producers' BHP and producer's liquid rate.

The paper is organized as follows. First, we introduce the related work on CRM. Second, we detail the theory of CRM and PINNs. Our approach called Deep-CRM is presented in the third section. We focus on the mathematical description of Deep-CRM and show experiments in order to compare our approach to the nonlinear multivariate regression on the closed form solution. These experiments are based on two datasets: a synthetic dataset generated using ECLIPSE® and SISMAGE®, and a real field dataset provided by one of our affiliates.

2 Related work

In this section we introduce a few techniques that are related to our approach Deep-CRM. A full description will be given in Section 3.3. Mathematical details and formulation are provided in the Appendix.

Capacitance resistive models: CRM are material balance models that can be used to study and predict the impact of water injection on producers. In Yousef et al. [2], and in his PhD dissertation [3], Yousef gave a widespread introduction to CRM from theoretical backgrounds to real field applications. In Sayarpour et al [4], authors firstly solve the CRM ODE analytically, in a closed form, using superposition in time and space, and then use nonlinear multivariate regression to optimize the physical parameters. In this work we use the exact solution developed in [4] as a comparison to our proposed Deep-CRM approach.

Artificial neural networks: Artificial Neural Networks (ANNs) presents a new paradigm for learning by mimicking the function of the human brain. Thanks to the universal approximation theorem [5], ANNs can approximate any continuous function, which makes them a powerful tool to model complex phenomenon with strong non-linearities. Earlier work from [6] has studied the impact of injectors on producers using ANNs. However, the connectivity obtained with such approach are qualitative, and cannot be used for example if one wanted to establish an allocation scenario. Another drawback of such approach is the non-integration of available physical information such as ODE or PDE. Such information can enhance the quality and the interpretability of the model forecasting. Lastly [7] combined Long short term memory LSTM [8] and EFAST [9] global sensitivity method to study the effect of injectors on producers. Same comments as for [6], the connectivity obtained with this approach are mainly qualitative and cannot be used to quantify how much a given injector can impact a given producer. As [6] this latter approach does not integrate physical constraints. In [7] authors did not study the total production rates forecasting. In this work, and based on CRM physical formulation, we develop a hybrid ANNs/CRM model to quantify the interaction between injectors-producers and to forecast total production rates.

Physics informed neural networks: A new approach called Physics Informed Neural Networks was recently introduced in [10, 11] using the power of deep learning to model and solve complex PDEs. Promising results were proven for several PDEs from various domains, e.g. Navier-Stokes [14], Darcy flow problem [13], and 1D & 2D Coupled Burgers' Equation [12]. In these papers it is shown that PINNs could provide better or similar quality results than classical solvers in these various application domains with generally less computational time.

3 Our approach: Deep-CRM

Before introducing how PINNs and CRM can be coupled, we detail separately the mathematics of each of these models.

3.1 The mathematics of capacitance resistive models

The name CRM came from the similarity between the CRM Ordinary Differential Equation (ODE) and the governing equations of electrical Capacitor Resistors Models. CRM

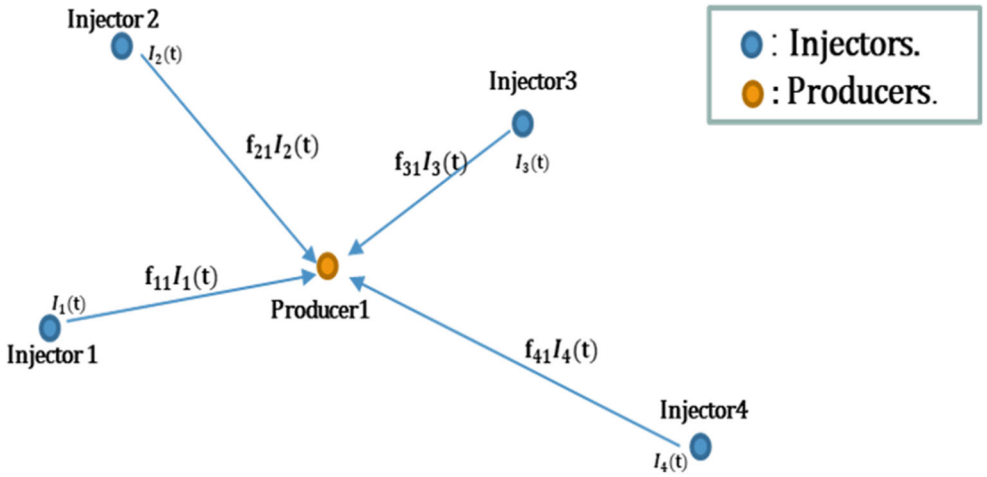

Fig. 1 Producer 1 is connected to Injector 1, Injector 2, Injector 3 and Injector 4. The connectivity with each Injector is measured using f_{ij}. The closest f_{ij} to 1 the strongest the connection between Producer j and Injector i

consists in modelling the variation of the total production rates over time $\mathbf{q(t)}$ for each producer by taking into account the injections $\mathbf{I(t)}$ of all injectors and the producer bottom hole pressure variations $\mathbf{p_{wf}(t)}$. CRM is derived from the following material balance (1) and the deliverability equation (2):

$$C_t V_p \frac{d\bar{p}}{dt} = I(t) - q(t) \tag{1}$$

$$q(t) = J(\bar{p}(t) - p_{wf}(t)) \tag{2}$$

Where C_t is the total compressibility, V_p is the control volume, $\bar{p}$ is the volume averaged pressure, $I(t)$ is the total injection at time $\boldsymbol{t}$, and $\mathbf{q(t)}$ is the liquid rate at time $\boldsymbol{t}$. p_{wf} is the bottom hole pressure, and $\boldsymbol{J}$ is the productivity index of the given producer. Replacing (2) in Eq. 1 yields the following CRM ODE equation:

$$\tau \frac{dq(t)}{dt} + q(t) = I(t) - \tau \times J \times \frac{dp_{wf}(t)}{dt} \tag{3}$$

Where τ is the time constant defined as:

$$\tau = \frac{C_t V_p}{J} \tag{4}$$

Let us consider $\boldsymbol{N}$ injectors and $\boldsymbol{M}$ producers, then Eq. 3 can be written as:

$$\forall j \in [1..M], \tau_j \frac{dq_j(t)}{dt} + q_j(t) = \sum_{i=1}^{N} f_{ij} I_i(t) - \tau_j \times J_j \times \frac{dp_{wf,j}(t)}{dt} \tag{5}$$

Example Figure 1 represents a case with $\boldsymbol{N} = 4$ injectors and $\boldsymbol{M} = 1$ producer, where the edges are weighted by the amount of injection. $I_i(t)$ represents the total injection of injector i at time t. f_{ij} represents the connectivity between injector i and producer j. $I_i(t) f_{ij}$ is the quantity of water injection received by producer j from injector i at time t.

Using the definition of f_{ij}, and for each injector i we can obtain the following constraint on f_{ij}:

$$\forall i \in [1..N], \sum_{j=1}^{M} f_{ij} I_i(t) \leq I_i(t) \Rightarrow \sum_{j=1}^{M} f_{ij} \leq 1 \tag{6}$$

The constraint (6) reflects that the injection of injector i is distributed on producers $j = 1..M$ based on their connectivity f_{ij}. The constraint in Eq. 6 is an inequality because some loss in water injection may occur due to the installation issues.

Using Eq. 6 we can rewrite (5) as:

$$\forall j \in [1..M], \tau_j \frac{dq_j(t)}{dt} + q_j(t) = \sum_{i=1}^{N} f_{ij} I_i(t) - \tau_j \times J_j \times \frac{dp_{wf,j}(t)}{dt}$$
$$\sum_{j=1}^{M} f_{ij} \leq 1 \tag{7}$$

$\forall i \in [1..N], \forall j \in [1..M], f_{ij} \geq 0, \tau_j \geq 0, J_j \geq 0$

In [4] the authors show that in the case of constant or linear variation in injection or in BHP, Eq. 5 can be solved analytically:

$$q_j(t_n) = q_j(t_0) e^{\frac{-(t_n - t_0)}{\tau_j}} + \sum_{k=1}^{n} \left\{ e^{\frac{-(t_n - t_k)}{\tau_j}} (1 - e^{\frac{-\Delta t_k}{\tau_j}}) \left[\sum_{i=1}^{N} \left[f_{ij} I_i^{(k)} \right] - \tau_j \times J_j \times \frac{\Delta p_{wf,j}^{(k)}}{\Delta t_k} \right] \right\} \tag{8}$$

where $q_j(t_n)$ is the j^{th} producer liquid rate at time t_n. $I_i^{(k)}$ is the rate of the injector i for the k time interval.

Knowing the closed form solution, and having available observations, the parameters of the CRM ODE can be obtained using a nonlinear multivariate regression:

$$(f_{ij}, \tau_j, J_j)_{j=1..M, i=1..N} = argmin \left(\sum_{j=1}^{M} \sum_{n=1}^{N_{step}} (q_j(t_n) - \overbrace{q_j(t_n)})^2 \right) \quad (9)$$

where $\overbrace{q_j(t_n)}$ is the observed rate of the j producer at time t_n.

In the next following sections we detail PINNs, ANNs and our approach named Deep-CRM.

3.2 The mathematics of physics informed neural networks

In this Section we present the Physics Informed Neural Networks PINNs approach. Let u be our quantity of interest (QOI) satisfying the following ODE:

$$N_t(u(t)) = f(t), t \in D \quad (10)$$

Where N_t is a differential operator, D is the time domain and $f(t)$ is a known function. The aim of PINNs is to solve (10) by approximating the QOI using ANNs.

ANNs are composed of many neurons connected to each other. Each neuron carries out a part of the total computation. Thanks to such connections ANNs can learn complex nonlinear functions [5].

Figure 2 shows an example of ANNs with one input t (representing time in this case) and one output $q_j(t)$ (representing the total production rates of producer j at time t). The ANNs has $L = 6$ hidden layers and 5 neurons per hidden layer.

For ANNs architecture we denote $W : E \rightarrow R$ as the weight function of the corresponding architecture. We define $h_W(t)$ as the prediction function of the architecture.

We define $\Delta(h_W(t), y(t))$ the loss for predicting $h_W(t)$ when the true (or target) value is $y(t)$. Δ is a distance function, e.g. $l^1, l^2 \ldots$ for a given H examples, the total loss of the networks is:

$$L_H(W) = \frac{1}{|H|} \sum_{y(t) \in H} \Delta(h_W(t), y(t)) \quad (11)$$

$|H|$ is the number of data observations. In the rest of this paper L_H will be called data loss. The output of the ANNs, approximating the QOI, should also satisfy the ODE (10) resulting in the next physics loss function:

$$L_D(W) = N_t(h_W(t)) - f(t), \forall t \in D \quad (12)$$

Thus, the final loss to minimize to get the optimal ANNs weight function W is:

$$L(W) = L_H(W) + \alpha * L_D(W) \quad (13)$$

The hyper-parameter α measures the balance between the physics loss (12) and the data loss (11). As any hyper-parameter, α need to be estimated on a validation dataset, different than the train dataset and the test dataset.

In next section we show how PINNs and CRM can be coupled.

3.3 CRM + PINNs = Deep-CRM

In this section we explain how CRMs and ANNs can be coupled in a PINNs approach. For such purpose we introduce Fig. 3. The main aim of Fig. 3 is to illustrate how the different quantities of interest: total production rates, bottom hole pressure, injections rates can be related or connected thanks to CRMs ordinary differential equation. Figure 3 is composed of two Figures, box 1 and box2. The left side of the box 2 contains three parts. The above ANNs represents the approximation of producer j bottom hole pressure (BHP) using producer j well head pressure (WHP). In the middle we have the ANNs that approximates producer j total production rates. The bellow part corresponds to the different injectors rates $(I_i(t); i = 1...N)$. The right side

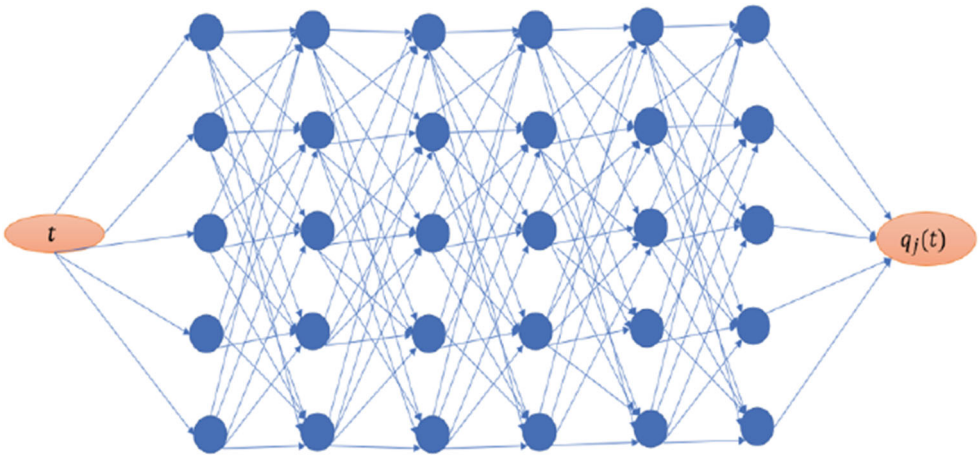

Fig. 2 Example of Artificial Neural Network with $L = 6$ hidden layers and 5 neurons per layer

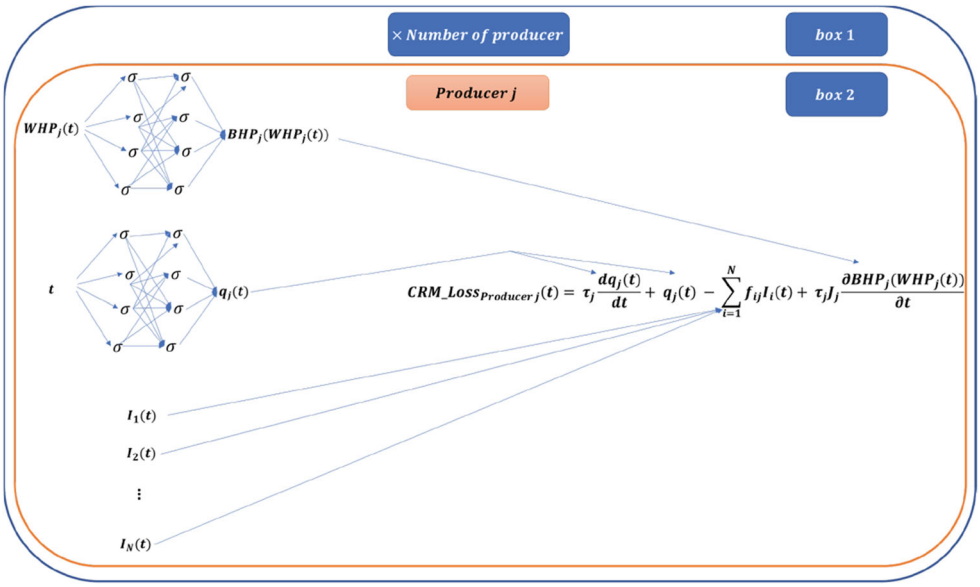

Fig. 3 Deep-CRM Architecture

of the box contains producer j CRMs loss or CRMs constraint. The different arrows indicates that the quantities on the left side are mapped to CRMs loss. This loss function is defined for each time step t. The global producer j CRMs loss will be the sum of this loss for the different time steps. Thanks to the minimization of this global CRMs loss we will ensure that the CRMs loss for producer j is respected during training and inference part. The box 1 indicates that the computation done in the box 2 will be performed for the all the different producers. The different producers global CRMs loss will be sum up giving the general CRMs loss defined in Eq. 14:

$$L_D(W, W_p) = \sum_{j=1}^{M} \sum_{t \in D_j} \left[\tau_j \frac{dq_j(t)}{dt} + q_j(t) - \sum_{i=1}^{N} f_{ij} I_i(t) + \tau_j * J_j * \frac{dp_{wf,j}(t)}{dt} \right] \tag{14}$$

Where $W = (w_1, w_2, \ldots, w_M)$ and $W_p = (w_{1p}, w_{2p}, \ldots, w_{Mp})$ with w_j and w_{jp} are respectively the weight function of the j^{th} producer total production rates ANNs and its corresponding BHP ANNs for $j = 1, .., M$. D_j represents the set of points where the j^{th} producer CRM ODE should be satisfied. The set of points D_j is referred to as collocation points and will usually include training and testing domains.

The Eq. 14 is the main part in Deep-CRM. Thanks to such loss the different quantities: producer total production rates $q(t)$, producer bottom hole pressure BHP $p_{wf}(t)$ and injection rate $I(t)$ are connected. That means that during the training process these quantities will simultaneously respect the historical data: on producer rate (16) and on BHP (17) plus the CRM constraint in Eq. (14). This results in model that respect historical data plus the given physics. This type of training is different than classical training where models are only trained to respect historical data. In our case the different models respect the historical data and satisfies the given physics on all the domain (Train and Test).

The loss in Eq. 14 constrains the model to respect the physics on a given domain. In our case we have chosen to apply it on all the domain including train domain and test domain. This implies that the different models will respect the physics in the past (training), and in the future (test). Thus the prediction of the different models is not based only on the historical part, but is also constrained to respect the given physics.

There are two reasons for why we have decided to estimate BHP using WHP. First, when dealing with different data-sets presented after, we have noticed that the BHP is not fully defined for the different producers. This problem is due to gauges/sensors problems. In the data-sets this effect corresponds to missing values in BHP (NaN values). On the other side, the WHP if fully defined for the different producers. In [20] authors have proposed to estimate BHP from WHP using ANNs. The mapping between BHP and WHP can be established using classical methods based on correlation analysis, however these approaches may be time and resources consuming. On the other hand Data-Driven approach like ANNs, and based only on historical data, can establish a full mapping between BHP and WHP. In this work we adopted similar approach as in [20] to estimate missing BHP using WHP. The main difference between our approach and approach in [20] is the introduction of CRMs ODE (loss in Eq. 14) in the training process.

Thanks to CRMs ODE the training of total production rates ANNs and BHP ANNs is done simultaneously. Which means that the BHP estimation is not based only

on WHP data, historical BHP data, but also on total production rates historical data.

Second, another benefit of estimating *BHP* using *WHP* is the automatic differentiation property. In fact, in CRMs ODE (loss in Eq. 14) and in the second part we need to compute the derivative of BHP with respect to time. Classically one can use finite difference approximation to compute such derivative. However, such approximation can suffer from stability problem and it introduces error in gradient computation. Thanks to ANNs' automatic differentiation property, the BHP gradient can be computed in an exact form following the next equation (15):

$$\frac{dBHP_j(t)}{dt} = \frac{dBHP_j(t)}{dWHP(t)} * \frac{dWHP_j(t)}{dt} \tag{15}$$

The derivative $\frac{dBHP_j(t)}{dWHP(t)}$ is computed using automatic differentiation thanks to BHP' ANNs. The $\frac{dWHP_j(t)}{dt}$ can be computed knowing that WHP is fully known.

The model total production rates data loss is defined in Eq. 16.

$$L_H(W) = \sum_{j=1}^{M} \sum_{y_j(t)\in H_j} \Delta(h_{qj}(t), y_j(t)) \tag{16}$$

Where $h_{qj}(t)$, $j = 1, \ldots, M$ is the j^{th} producer total production rates at time $\boldsymbol{t}$. H_j contains the j^{th} producer observations for $j = 1, \ldots, M$.

The BHP data loss is defined in Eq. 17:

$$L_{H_p}(W_p) = \sum_{j=1}^{M} \sum_{\bar{p}_j(t)\in H_{p_j}} [\Delta(h_{pj}(t), \bar{p}_j(t))] \tag{17}$$

where $h_{pj}(t)$, $j = 1, .., M$ represents the prediction of the j^{th} producer BHP at time $\boldsymbol{t}$, H_{p_j} its corresponding set of observations.

At the same time, the producers should also respect the different constraints on the physical parameters (Table 1). This results in the global loss:

$$\begin{aligned} &Global\_loss(W, W_p, f, \tau, J) \\ &= L_H(W) + L_{H_p}(W_p) + L_D(W, W_p) + L_{Injector}(f) \\ &\quad + L_{Connectivity}(f) + L_{Timeconstant}(\tau) \\ &\quad + L_{Indexofproductivity}(J) \end{aligned} \tag{18}$$

With $f = f_{(i,j)\in 1..N\times 1..M}$, $\tau = (\tau_j)_{j=1..M}$, $J = (J_j)_{j=1..M}$. and

In next section we introduce the different experiments on the two datasets.

4 Experiments

In this section we describe two applications of Deep-CRM on two datasets: first on a synthetic dataset and then on a real dataset. Before applying the Deep-CRM method we have to perform a data rescaling and a data filtering. Here we provide some details of these two operations.

Data rescaling Before applying Deep-CRM we rescale the injections and production total rates by the maximum rate production computed over all the producers; the pressure is rescaled with respect to the maximum pressure. Rescaling the data aims at making the learning faster and prevents the network from stacking in local minima[18].

Data processing CRM ODEs explain the interactions between injectors and producers based on their stabilized signals. In Fig. 4 left plot, we can notice the presence of many shutdowns and spikes in the producers total production rates. Those shutdowns and spikes are due to human intervention and cannot be explained by injections signals. Thus, training the model on data containing such shutdowns and spikes can lead to consider spurious correlations between producers and injectors rates. In order to remove shutdowns and spikes we apply a low pass filter on the data. An example of this operation is in Fig. 4, where the figure on the left represents the Producer 5 liquid rate before low pass filter processing and the figure on the right represents the modified liquid rate of Producer 5 after the low pass filter application.

4.1 Objectives and protocols

Deep-CRM has dual objectives, firstly discovering the underlying connectivity between injectors and producers, and secondly performing total production rates forecasting giving the amount of injected water and the bottom hole pressure BHP of each producer. In these experiments we evaluate Deep-CRM and we compare them to classic CRM on both objectives. For neural networks, there is no universal configuration (optimizer, number of neurons, number of layers, activation function) that works for every case. Each

Table 1 Constraints and its corresponding losses used in our approach

Constraint	Loss
$\forall i \in [1..N], \forall j \in [1..M], \sum_{j=1}^{M} f_{ij} \leq 1$	$L_{Injector} = \sum_{i=1}^{N} max(0, \sum_{j=1}^{M} f_{ij} - 1)$
$\forall i \in [1..N], \forall j \in [1..M], f_{ij} \geq 0$	$L_{Connectivity} = \sum_{i=1}^{N} \sum_{j=1}^{M} max(0, -f_{ij})$
$\forall j \in [1..M], \tau_j \geq 0$	$L_{Timeconstant} = \sum_{j=1}^{M} max(0, -\tau_j)$
$\forall j \in [1..M], J_j \geq 0$	$L_{Indexofproductivity} = \sum_{j=1}^{M} max(0, -J_j)$

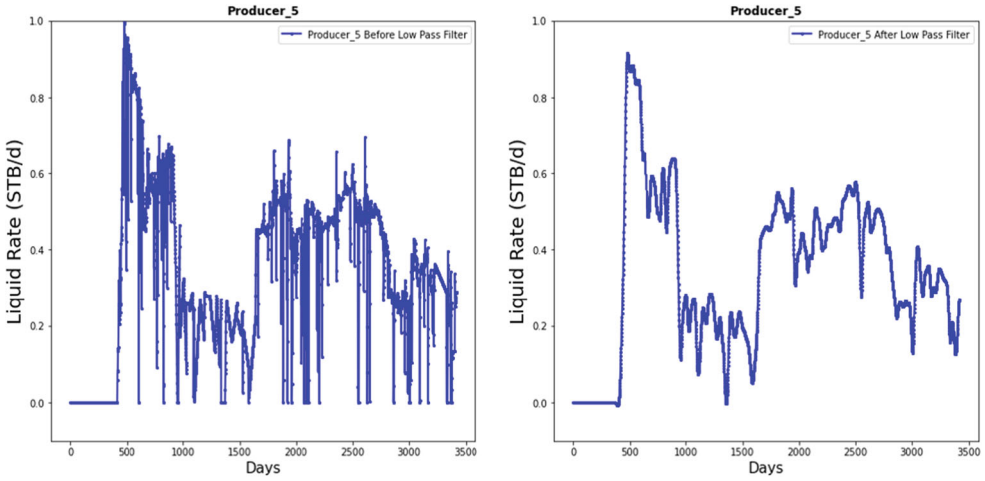

Fig. 4 Producer 5 total production rates before low pass filter processing (right) and after low pass filter processing (left)

case study is different, though we need to test several configurations and validate them on a validation set. To find the optimal combination of these hyperparameters we use a simple grid search approach [19]. This process was adopted for the synthetic dataset. For the real dataset we have kept the same configuration as for the synthetic dataset.

For the synthetic dataset, training phase is performed on 50% of the data, 10% of the data is used as validation for selecting the best model and 40% for testing. For the real dataset 70% of the dataset is used for training and 30% for testing. The following metrics are used for model selection and comparison with the classical optimizer SLSQP https://docs.scipy.org/doc/scipy/reference/optimize.minimize-slsqp.html:

$$MSE(y, \hat{y}) = \mathbb{E}[(y - \hat{y})^2] = \frac{1}{N}\sum_{i=1}^{N}(y_i - y_i)^2 \tag{19}$$

$$MAE(y, \hat{y}) = \mathbb{E}[|y - \hat{y}|] = \frac{1}{N}\sum_{i=1}^{N}|y_i - y_i| \tag{20}$$

$$NMSE(y, \hat{y}) = \frac{MSE(y, \hat{y})}{Max(y)} \tag{21}$$

$$NMAE(y, \hat{y}) = \frac{MAE(y, \hat{y})}{Max(y)} \tag{22}$$

4.2 Synthetic dataset

The synthetic dataset called Sondous was simulated using SIMAGE® and ECLIPSE® (see Figs. 5 and 6). It contains 97 observations irregularly distributed between November 1, 1988 and November 16, 1998. Figure 5 represents a 3D view of the field. We can observe the presence of a fault in the middle of the field (black line). The scale of colour corresponds to the net-to-gross (NTG). Figure 6 illustrates the same field in a 2D view, where the 3 producers (P1, P2 and P3) and 2 injectors (I1 and I2) are positioned. We can notice the presence of the fault separating the producer one (P1) from the rest of producers (P2 and P3) and injectors (I1 and I2). The sealing fault was introduced on purpose in the synthetic dataset, to test the ability of Deep-CRM to

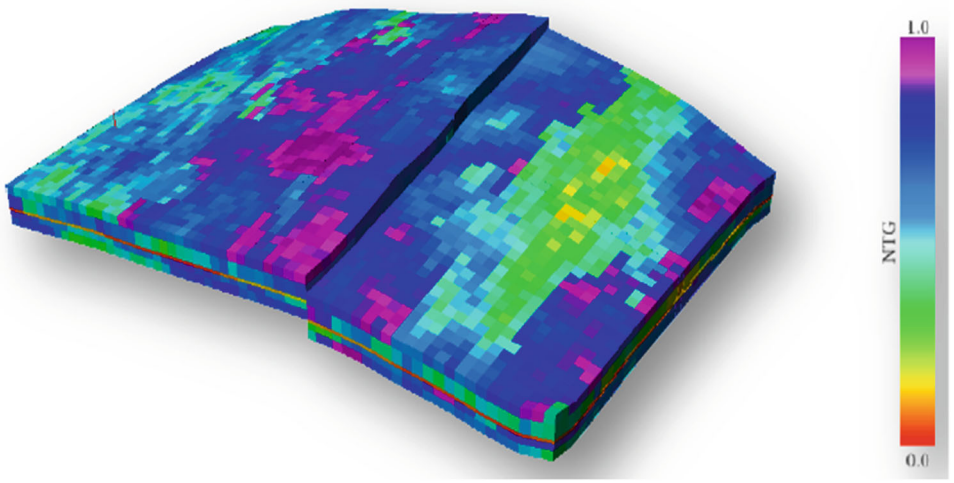

Fig. 5 Sondous field in 3D image using NTG scale. The fracture in the middle represents the sealing fault

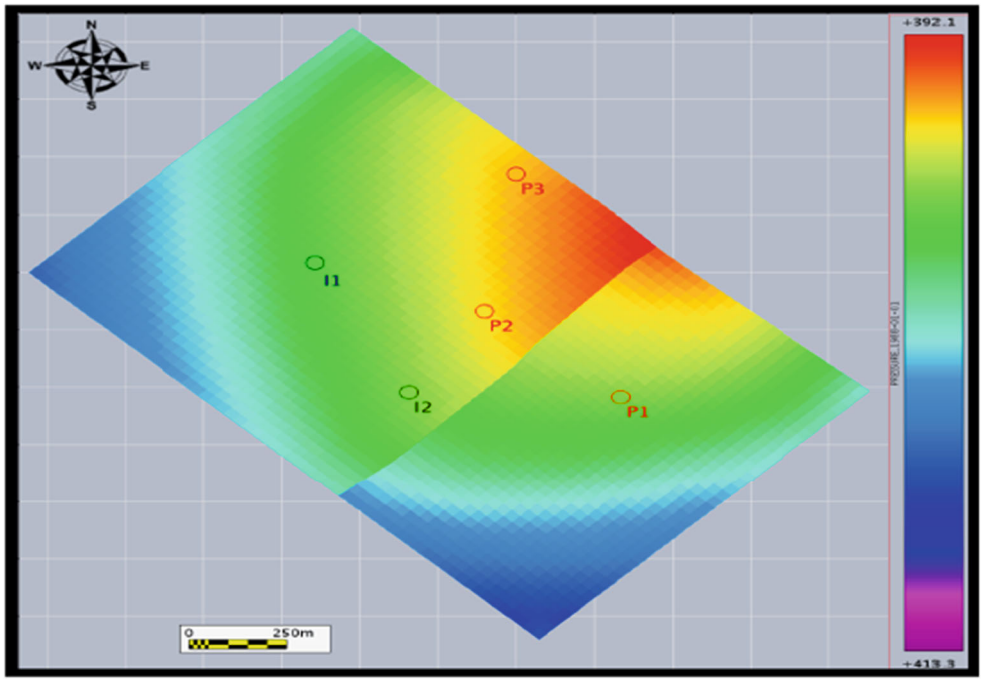

Fig. 6 Sondous field in 2D image using pressure scale. The middle line represents the sealing fault separating Producer 1 (P1) from Injectors I1, I2, Producer 2 (P2) and Producer 3 (P3)

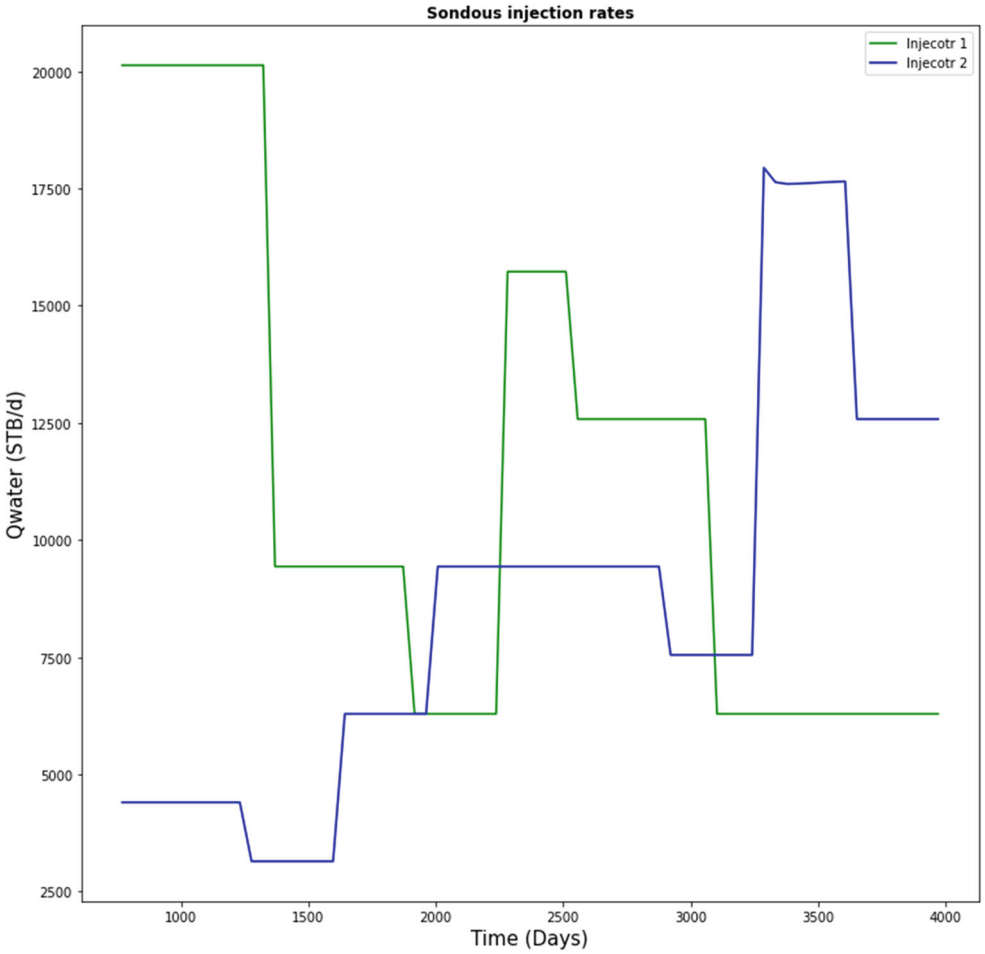

Fig. 7 Synthetic dataset injection liquid rate for Injector 1 and Injector 2

discover the interactions between injectors and producers. In Fig. 7 the colour scale represents the reservoir pressure. The BHP is constant for the different producers. $BHP_{P1} = 396\ bar$, $BHP_{P2} = 395\ bar$ and $BHP_{P3} = 395.5\ bar$.

Figure 8 shows the total production rates for the three producers, while Fig. 7 shows the injection rates for the two injectors. In Fig. 7 we can observe the stepwise variation of the different injectors liquid rates. In Fig. 8 we can observe that the curve of P1 is significantly lower than those of P2 and P3.

We first analyze the forecasting ability of Deep-CRM. Figure 9 represents the evolution of Deep-CRM main loss functions over the number of optimization iterations. Training loss is the global loss to minimize (blue curve), which includes the Physics loss (green curve) and the Model loss (red curve), plus the ODE constraints losses. We can observe that the RMSprop optimizer manage to minimize all the different losses over the iterations. Moreover, we can see that the different losses reach a plateau around 20.000 iterations. As a result we stop the optimizer after 30.000 iterations. The validation loss is particularly important since it shows the generalization of the model. A very known problem in machine learning is the model overfitting. It means that the model cannot generalize outside the learning dataset in such case the validation loss will increase although the model general loss is decreasing. In our case the validation loss is decreasing which indicates that our model is not overfitting and therefore we will be able to use it outside the learning dataset.

The hyper-parameter α introduced in Eq. 13 has been estimated on the same validation dataset (10% of the total dataset). In Fig. 10 we show the evolution of the NMSE for different values of α between 0 and 1 with a step of 0.1. The optimal value is $\alpha = 0.7$. α was estimated with the best model from the first grid search.

In Table 2 we compare different architectures with different numbers of layers ([5,6]) and numbers of neurons per layer ([200,300]). We can see that the configuration 6 hidden layers and 300 neurons per hidden layer has the lowest NMSE and thus will be retained as the best configuration.

In Figs. 11, 12 and 13 we present the prediction of Deep-CRM with the best configuration (red curve), the classic CRM with SLSQP https://docs.scipy.org/doc/scipy/

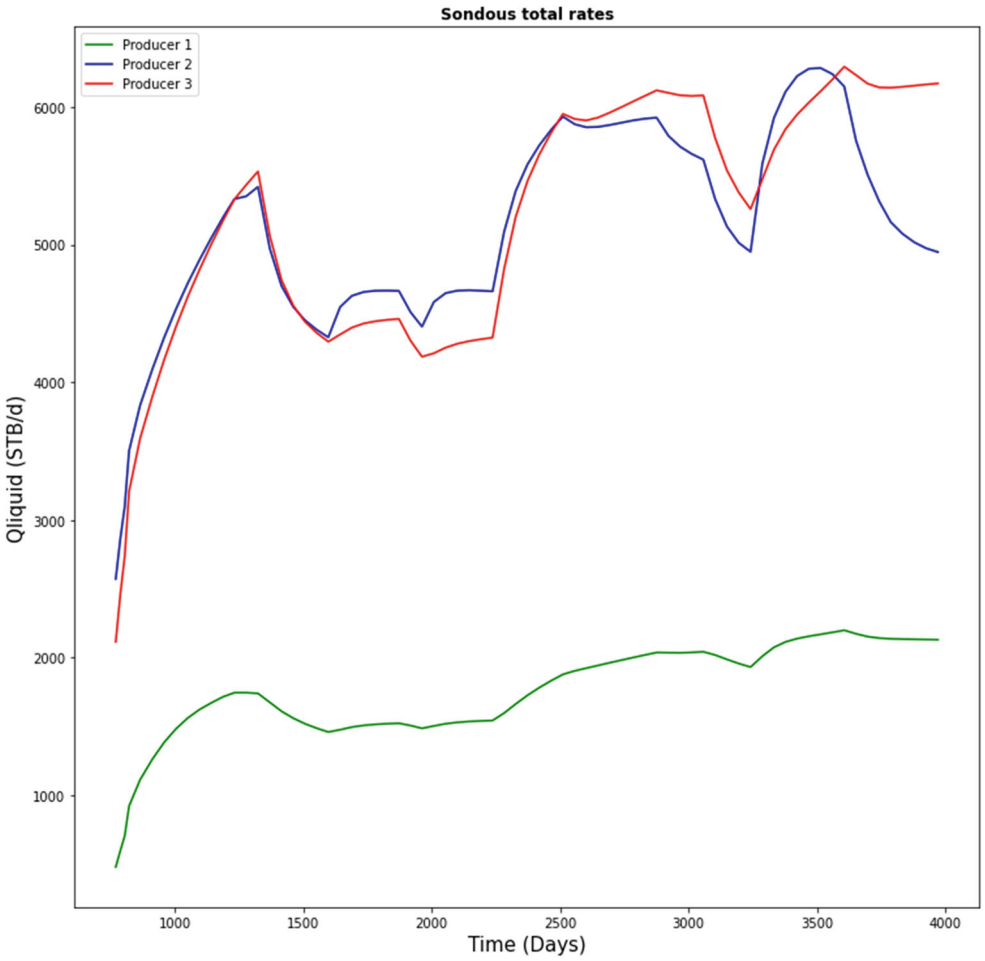

Fig. 8 Sondous Producer1, Producer 2 and Producer 3 liquid rates

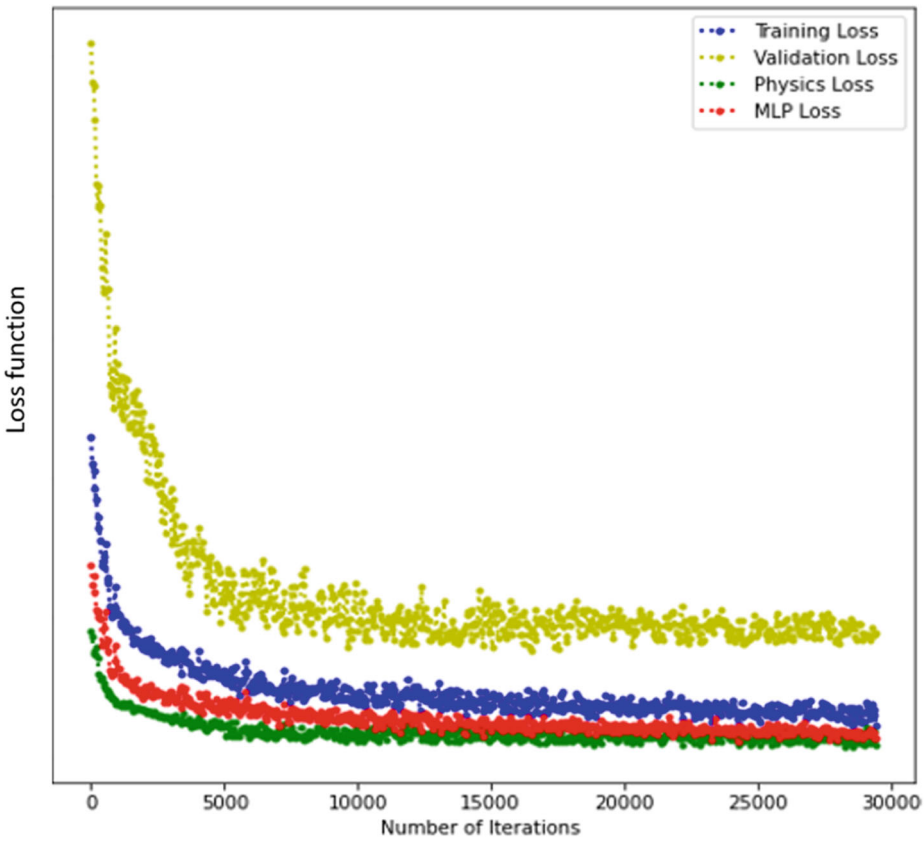

Fig. 9 Training loss, Validation loss, Physics loss and MLP loss, for Deep-CRM on Sondous case

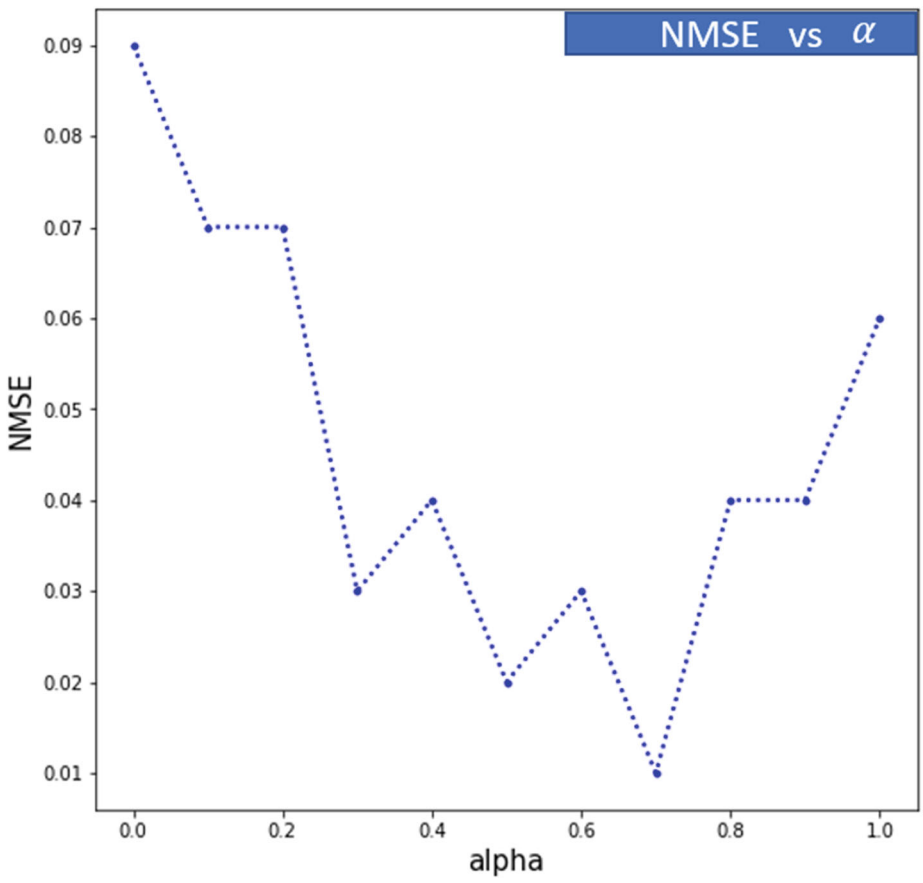

Fig. 10 Variation of the NMSE for α between 0 and 1, with a step of 0.1

Table 2 Grid search for number of layers and number of neurons per layer

	Producer 1	Producer 2	Producer 3	Total NMSE
Model 1 [5 layers, 200 neurons per layer]	1.72	10.22	1.22	13.16
Model 2 [5 layers, 300 neurons per layer]	1.46	21.3	0.79	23.55
Model 3 [6 layers, 200 neurons per layer]	1.15	26.76	1.04	28.95
Model 4 [6 layers, 300 neurons per layer]	0.3	12.17	0.48	12.95

reference/optimize.minimize-slsqp.html (dark curve), compared to the real data (blue curve). The comparison between Deep-CRM prediction, classic CRM with SLSQP https://docs.scipy.org/doc/scipy/reference/optimize.minimize-slsqp.html prediction, is done on the test part. The latter correspond to 40% of the time series. The test part starts after the dark olive line till the end of the time series.

The first line in Table 3 contains the normalized mean squared error (NMSE) between Deep-CRM prediction and the real data on test part. The second line contains the NMSE between classic CRM with SLSQP https://docs.scipy.org/doc/scipy/reference/optimize.minimize-slsqp.html and real data on the test part.

Based on Table 3 we can conclude that Deep-CRM has lower mean of MAE (Mean of MAE = 0.05) compared to classic CRM with SLSQP (Mean of MAE = 0.06). Table 3 proves that Deep-CRM is performing better than classic CRM with SLSQP in terms of MAE.

In Table 4 we show the obtained values for the physical parameters: wells connectivity and time Constants, Using Deep-CRM architecture.

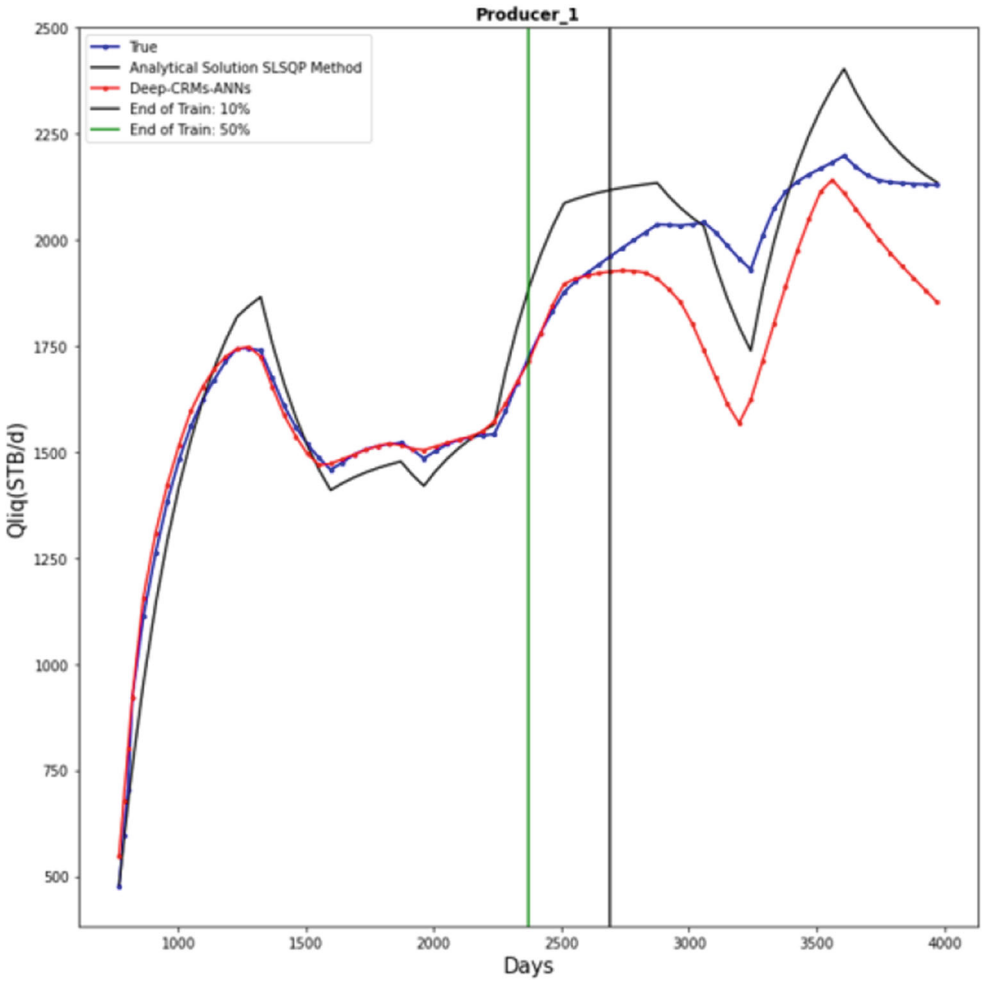

Fig. 11 Producer 1 Deep-CRM and SLSQP forecasting (Sondous data)

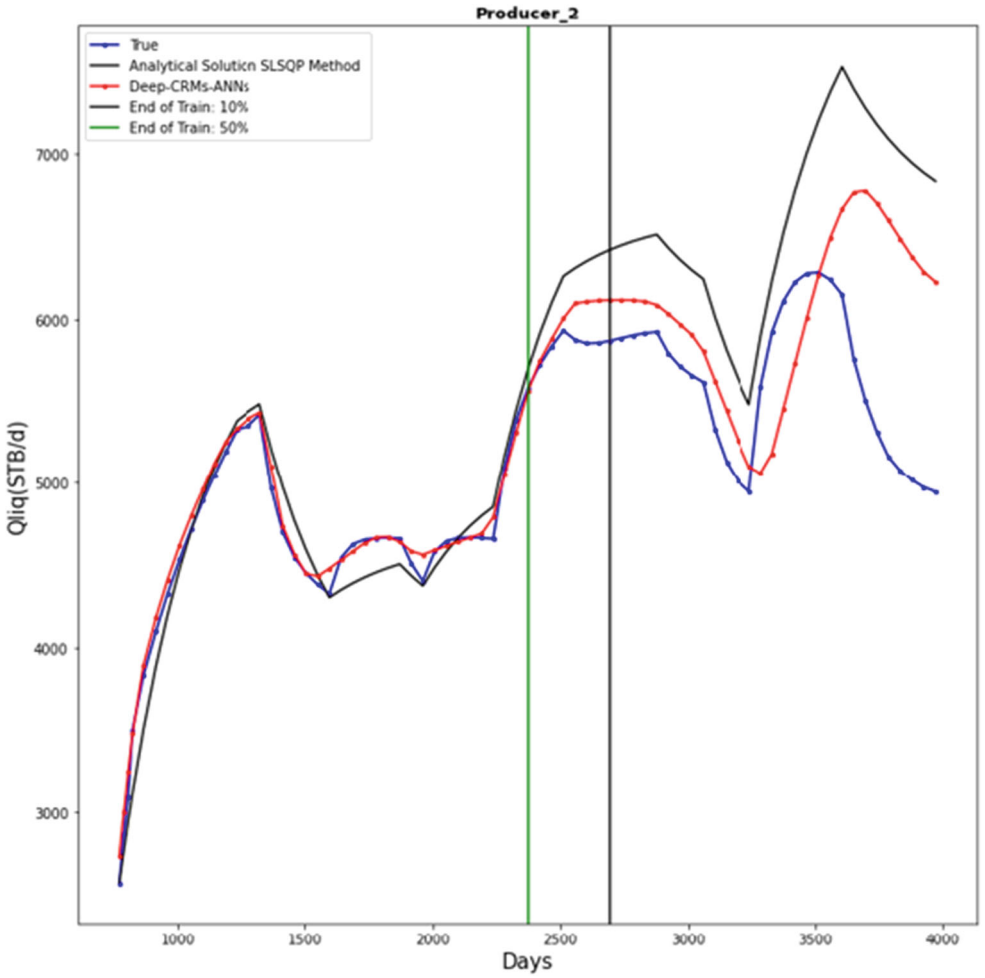

Fig. 12 Producer 2 Deep-CRM and SLSQP forecasting (Sondous data)

Table 4 show that, connectivity between Injector 1 and Producer 1 is nearly equal to zero, and the connectivity between Injector 2 and Producer 1 is smaller than the connectivity of Injector 2 with other producers. These results confirm the presence of a partially sealing fault separating Producer 1 from other Injectors.

Table 5 presents the connectivity obtained with classic CRM using SLSQP optimizer. We can see that the approach can detect the sealing fault separating Producer 1 from Injector 1 and Injector 2.

On the synthetic dataset: Sondous, the two methods give the same results in term of connectivity. In term of forecasting Deep-CRM gives better result than classic CRM with SLSQP optimizer.

4.3 Real dataset

The real dataset provided by one of our affiliates, represents an offshore oil field. It contains 6 producers and 5 injectors. CRM are applicable only at reservoir bottom conditions, for that reason we have transformed the total production rates from surface condition to the bottom of the reservoir condition:

liquid rate (bottom) = water + Bo×Oil with Bo=2.3

In Figs. 14, 15, 16, 17, 18 and 19 we show the daily production rates of the 6 producers over 3417-time daily observations starting from March 3, 2009 to June 7, 2018. The data was normalized between 0 and 1 due confidentiality clauses. This normalization will be applied to different data related to the real case. In comparison to the synthetic dataset (see Fig. 8) the real dataset contains many spikes due to field operations thus before applying our approach we need to filter those spikes since they cannot be explained by injection signal or BHP signal.

In Figs. 20, 21, 22, 23 and 24 we present the injectors water rates. The data are also normalized between 0 and 1 for confidentiality issues. Same comments as for the total production rates, injectors water rates present many shutdowns and spikes.

In Figs. 25, 26, 27, 28, 29 and 30 we plot each producer bottom hole pressure (BHP) and its corresponding well head pressure (WHP). The blue line corresponds to the BHP, and

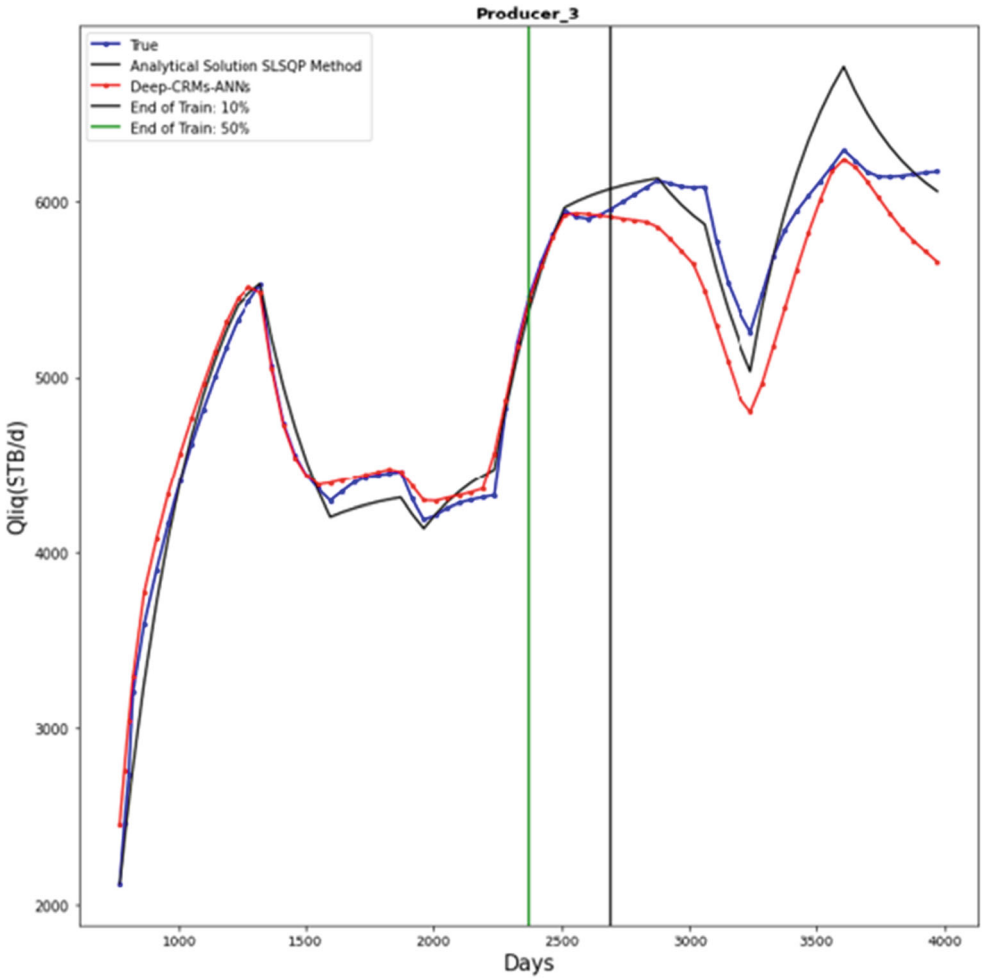

Fig. 13 Producer 3 Deep-CRM and SLSQP forecasting (Sondous data)

Table 3 MAE for Deep-CRM and CRM using SLSQP optimizer

	Producer 1	Producer 2	Producer 3	Mean
Deep-CRM	0.015	0.085	0.045	0.05
CRM (SLSQP)	0.018	0.13	0.024	0.06

Table 4 Deep-CRM Connectivity and Time Constant

Connectivity	Producer 1	Producer 2	Producer 3
Injector 1	0.07	0.21	0.22
Injector 2	0.11	0.36	0.32
Time Constant (days)	204	208	211

Table 5 CRM with SLSQP Connectivity and Time Constant

Connectivity	Producer 1	Producer 2	Producer 3
Injector 1	0.07	0.22	0.23
Injector 2	0.12	0.40	0.33
Time Constant (days)	280	329	297

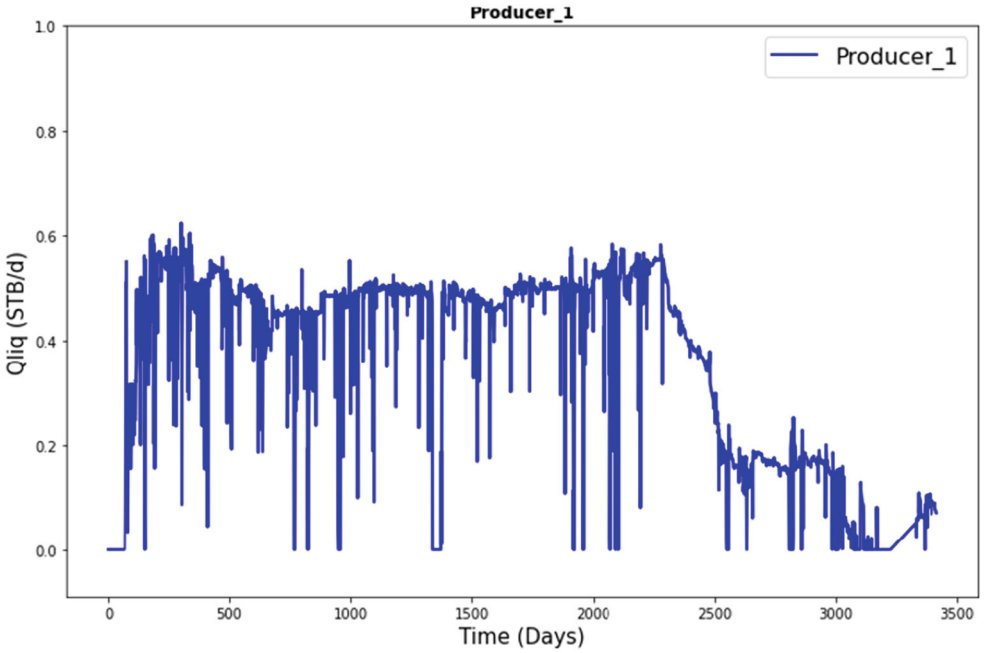

Fig. 14 Producer 1 total production rate

the dark line corresponds to the WHP. As we can see on the different plots, the BHP contains missing values. Two type of missing values can be noticed: 1- Small missing values representing a blank between two known values. This effect can be noticed on the different producers BHP. 2- Large missing values, where a big part of the BHP is missing. This case corresponds to Producer 1, Producer 3, and Producer 5. On the other hand one can notice that the well head pressure is fully defined for each producer. As explained before in section 3.3, we will complete the missing BHP values using available BHP values and WHP values.

Using geological information and different techniques e.g. interference test, pressure response at producer wells when injection is on/off, salinity test, tracer test and 4D seismic images. Based on the results of all these tests we constructed Table 6 where we have summarized this infor-

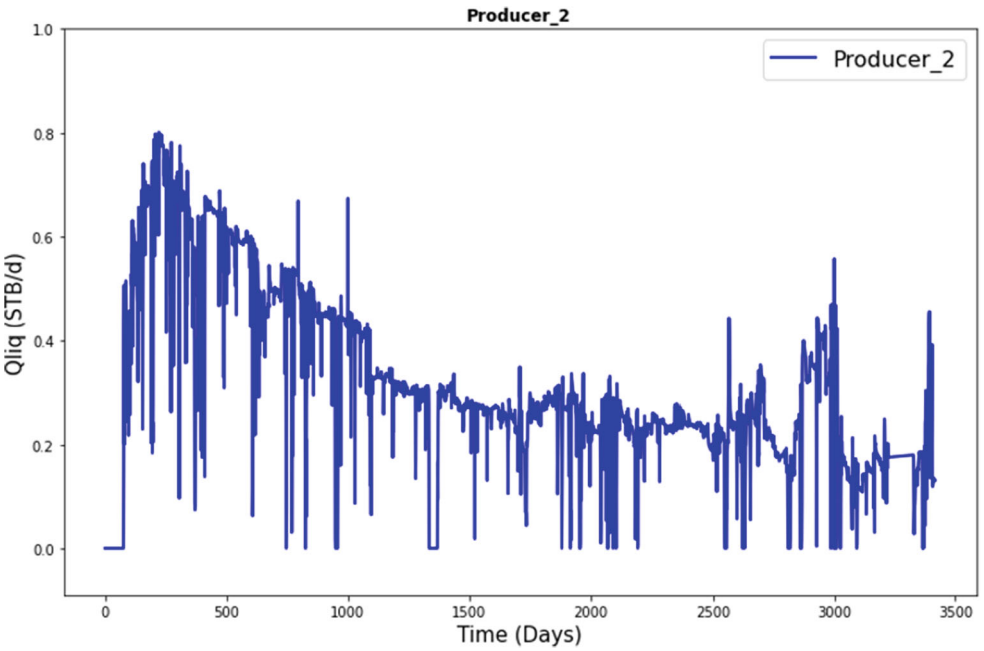

Fig. 15 Producer 2 total production rate

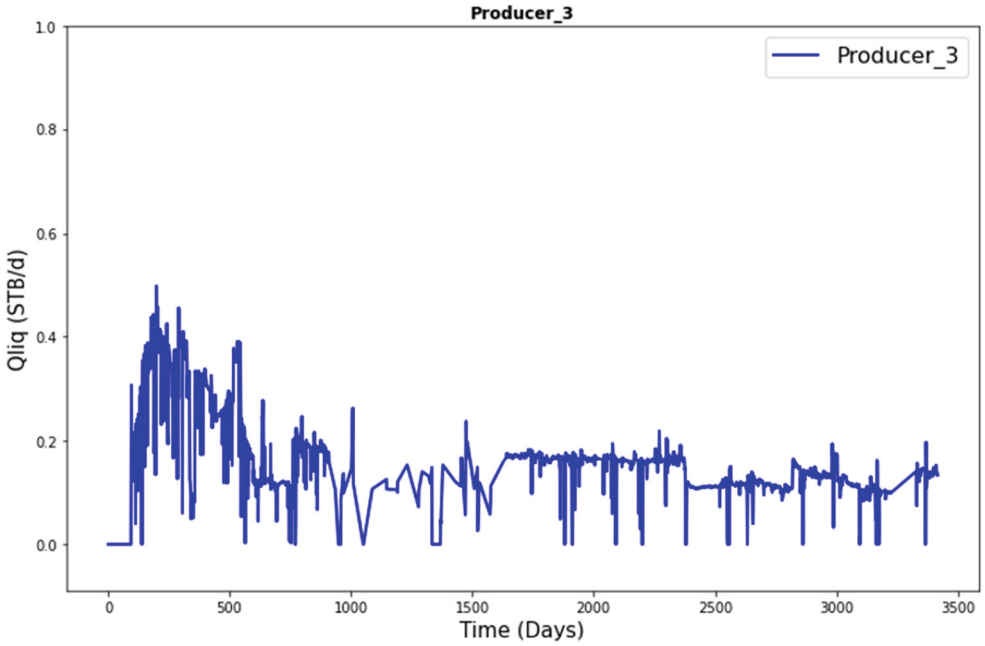

Fig. 16 Producer 3 total production rate

mation. Strong connection means that the connection has been proven by our affiliate. Weak connection corresponds to the case of proven connection but at the same time not a strong connection. No connection is the case of impossible connection between injector and producer. No Information connection means that no test have been performed by our affiliate, but probably because they thought that the wells were not or weakly connected.

To help the optimizer finding the best solution we have decided to set a priori the No connection to zero to limit the space of possible solutions.

4.3.1 Deep-CRM vs Classic CRMs

In this Part we discuss the application of Deep-CRM on the real field dataset. Moreover, in this case study we compare

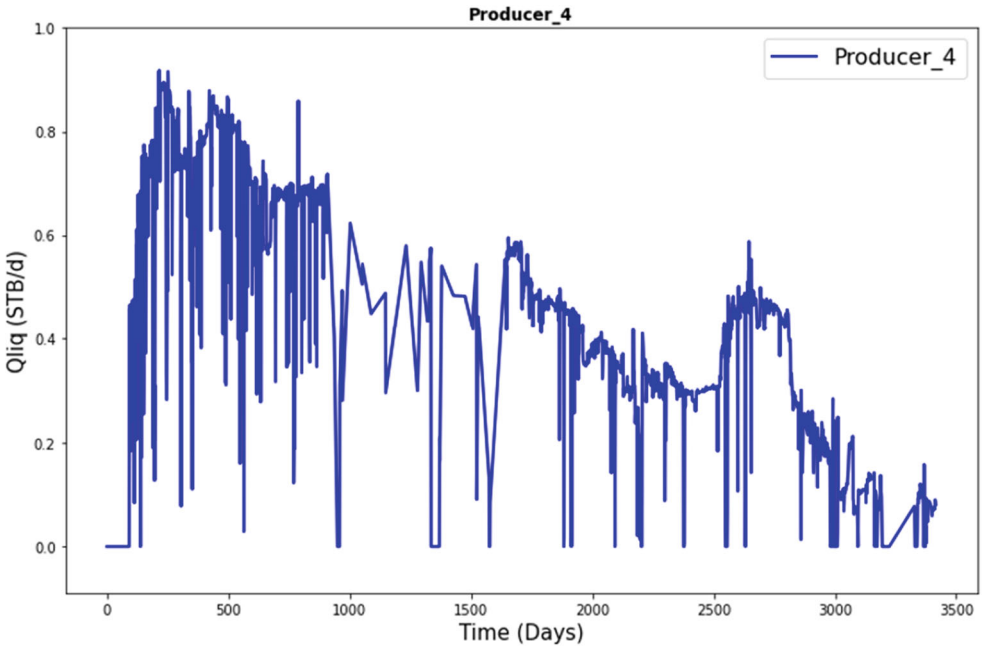

Fig. 17 Producer 4 total production rate

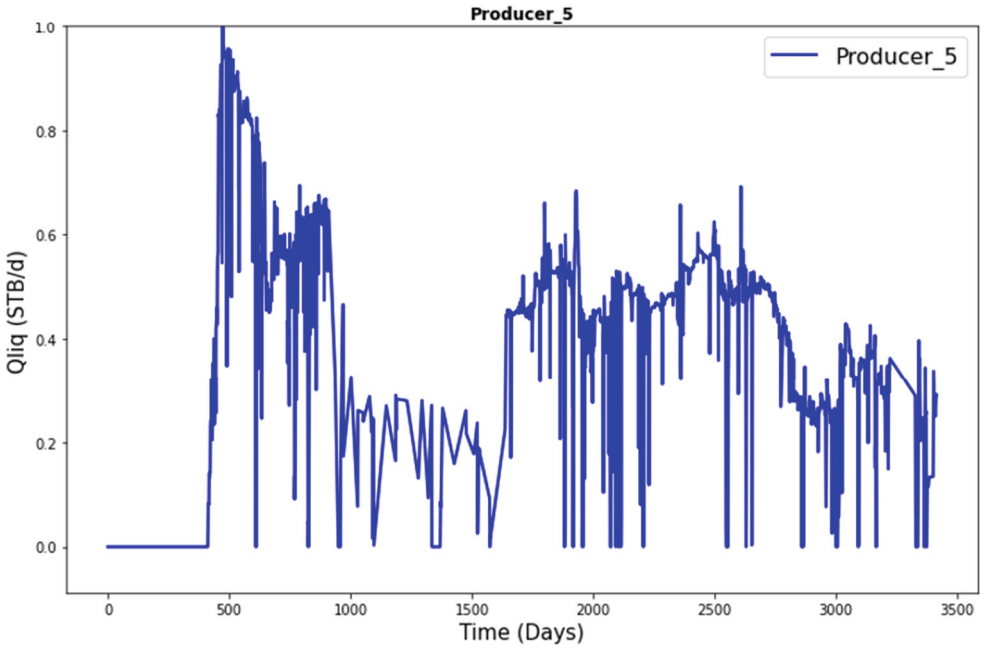

Fig. 18 Producer 5 total production rate

Deep-CRM approach to classical CRM using SLSQP optimizer.

In Figs. 31, 32, 33, 34, 35 and 36 we present the results of BHP missing values interpolation and extrapolation with Deep-CRM. The blue curve represents BHP values. The dark curve corresponds to WHP values. The red curve represents Deep-CRM interpolation and extrapolation of BHP values. In future work we will extend the comparison between BHP after imputation and before imputation. At this stage we can say that Deep-CRM provides good interpolation and extrapolation of BHP based on available BHP data, and Well head pressure data.

Figures 37, 38, 39, 40, 41 and 42 shows the results of Deep-CRM (blue curve, continuous line) and SLSQP (red curve, dotted line) on the real dataset (dark curve, dot line). The vertical green line separates 70% of the data, representing 6 years of production, used for training and 30%, representing 3 years of production, used for testing.

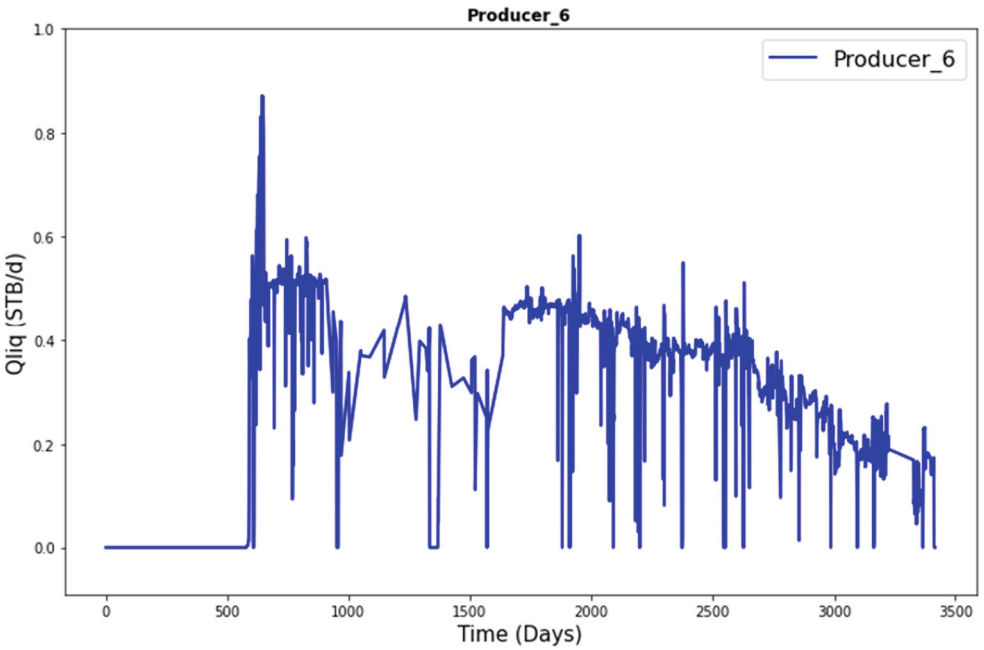

Fig. 19 Producer 6 total production rate

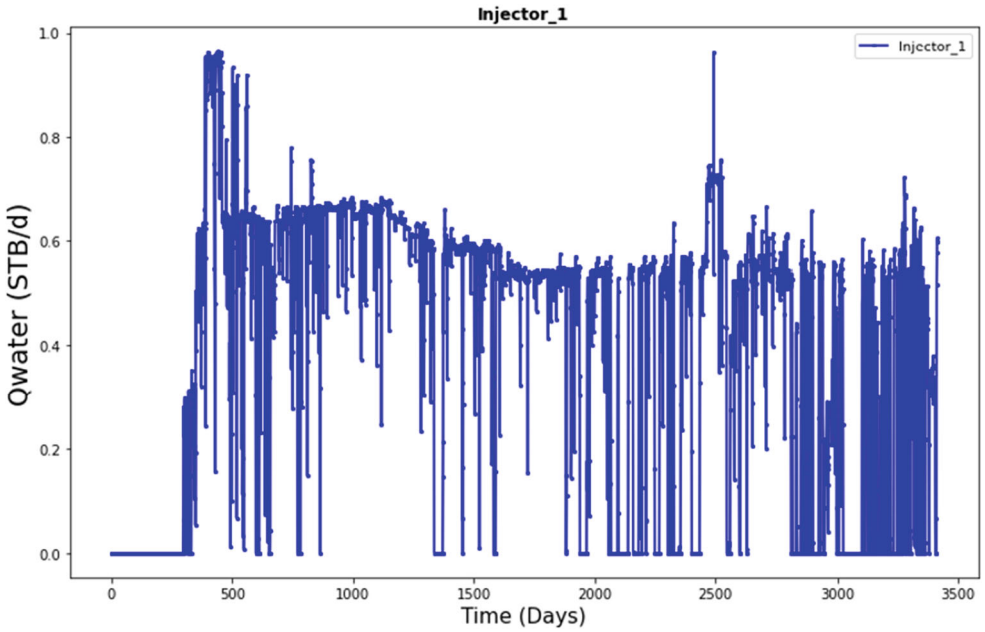

Fig. 20 Injector 1 water rate

In Table 7 we compare Deep-CRM approach to Classic CRM with SLSQP using the mean absolute error. The last column of Table 7 is the mean of the MAE on different producers for Deep-CRM and CRM (SLSQP). According to Table 7 Deep-CRM has the lowest MAE compared to Classic CRM with SLSQP optimizer. This shows that Deep-CRM performs better than CRM with SLSQP also in this test case.

Table 7 presents the physical parameters (i.e. connectivity per producer-injector couple) obtained after Deep-CRM optimization. First observation is that all the connectivity are between 0 and 1 as it should be. The second observation is that the sum of connectivity per row is less than 1 meaning that constraints in (23) has been respected. A third observation is that with Deep-CRM we have less loss in the injection compared to CRM (SLSQP) in Table 9. The

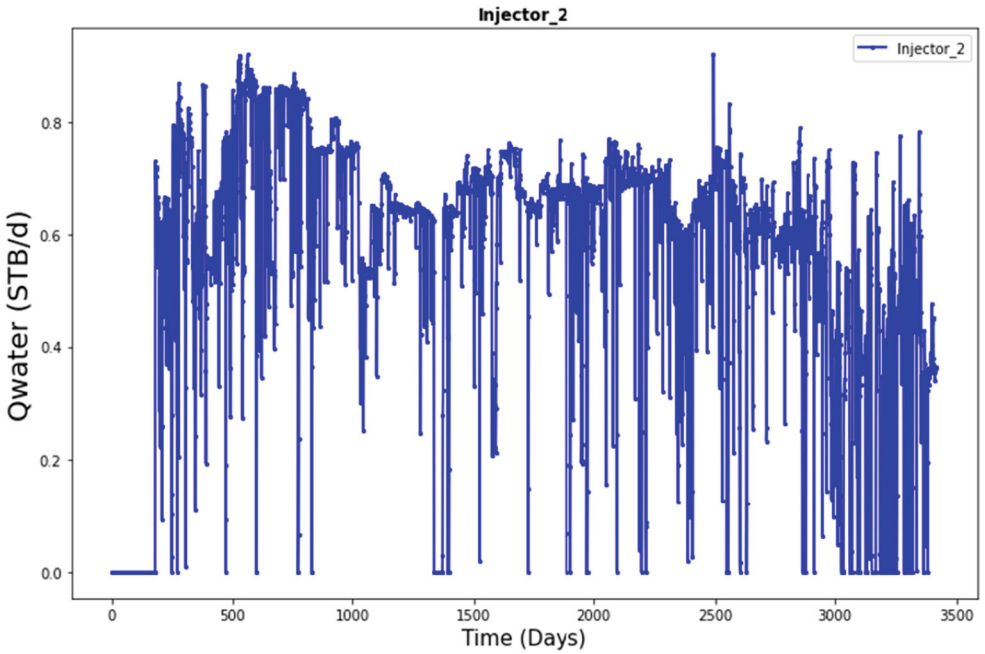

Fig. 21 Injector 2 water rate

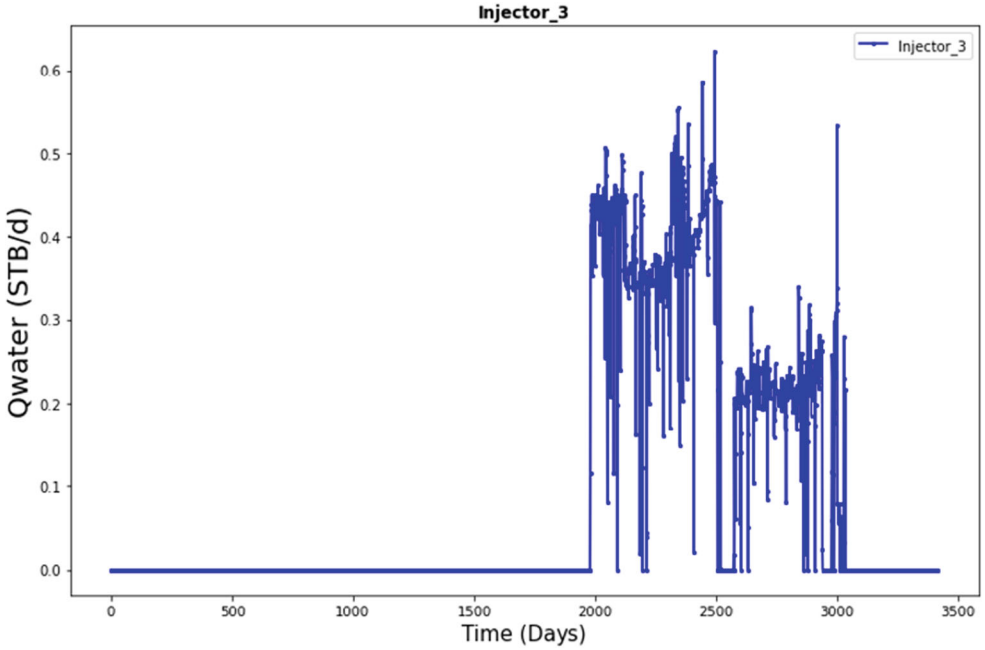

Fig. 22 Injector 3 water rate

loss in total injection observed in CRM (SLSQP) connectivity Table 9 could explain the low level of Producer 4, Producer 5 and Producer 6 liquid rates compared to the true liquid rate. Another way to compare the two approaches is to count the good matches with the green and cyan color in the two Tables 8 and 9. A debatable threshold to tell if the connection is good could be $f = 0.01$. That means if $f_{ij} \leq f$ then the connection is considered to be bad. We can see that Deep-CRM has identified 14 good connections ($f_{ij} > f$). All these 14 connections has green or cyan color in the prior Table 6. For the CRM with SLSQP we have 11 connections above the threshold that matches with the green or cyan color. We remind that the connections with yellow colors, were not studied by the asset, it is then not possible to

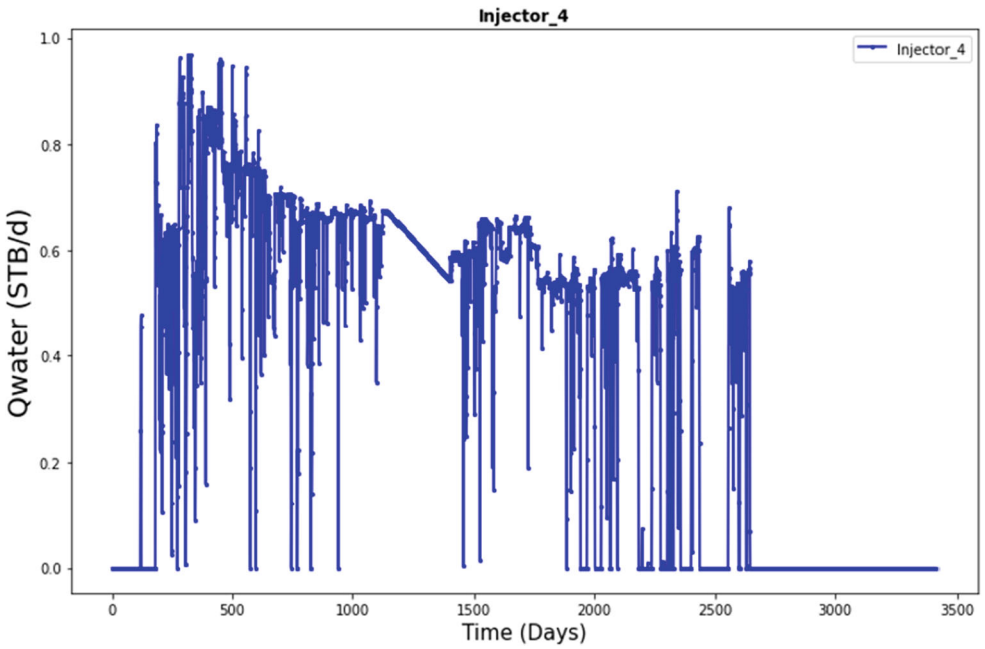

Fig. 23 Injector 4 water rate

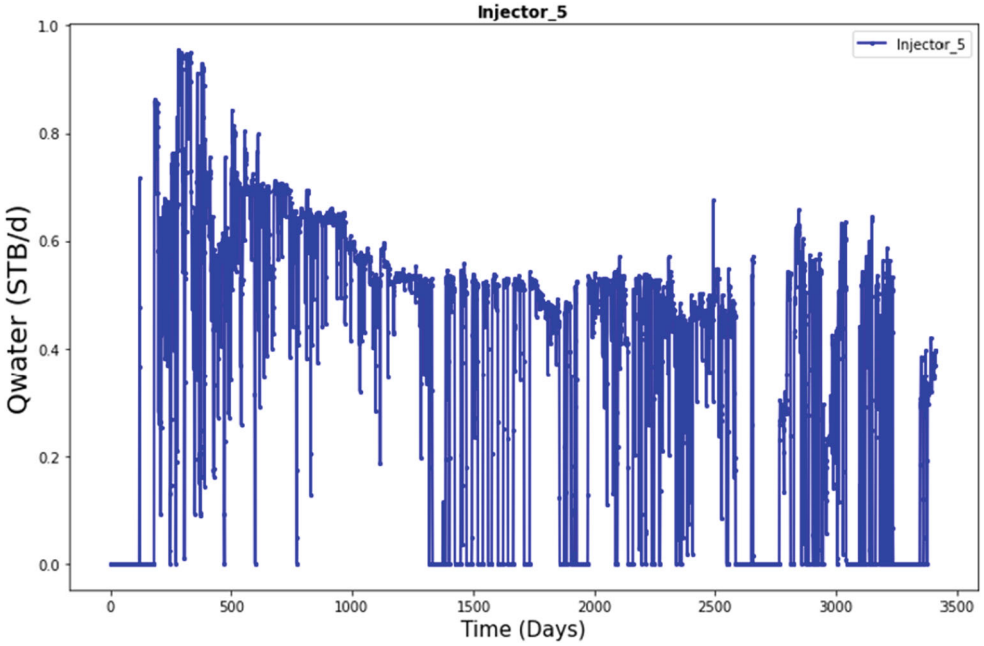

Fig. 24 Injector 5 water rate

conclude on these connections. However it is interesting to note that Deep-CRM seems to indicate some non-zero connections between those wells, as a result these connections should probably be reconsidered or analyzed more in depth by the asset team.

4.3.2 Deep-CRM vs ANNs

In this part we compare Deep-CRM with 1d convolution neural netrowks (CNNs) [21].

In this part we have trained Deep-CRM, in old fashion way, using only historical observation data on total production rates and on bottom hole pressure. The purpose of this experimentation is to quantify the effect of the no integration of CRM physics in the model. In Fig. 43 we plot the evolution of the training loss (in blue) and the validation loss (in yellow) over the number of iterations. One obvious observation is the validation loss evolution. After 12000 iterations the validation loss starts increasing while the training loss still decreasing. This behaviour indicates that

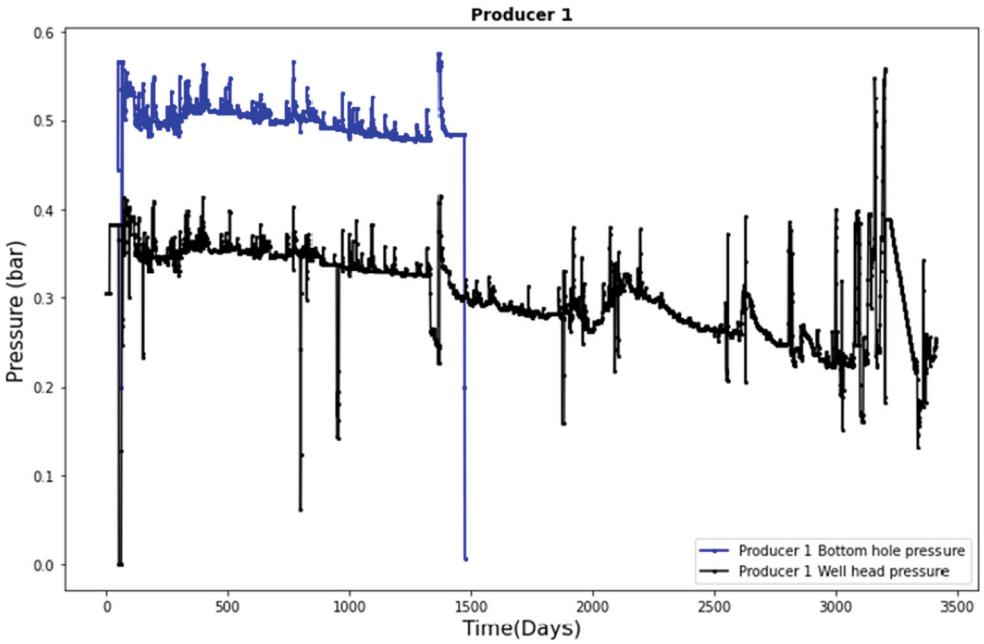

Fig. 25 Producer 1 bottom hole pressure and well head pressure)

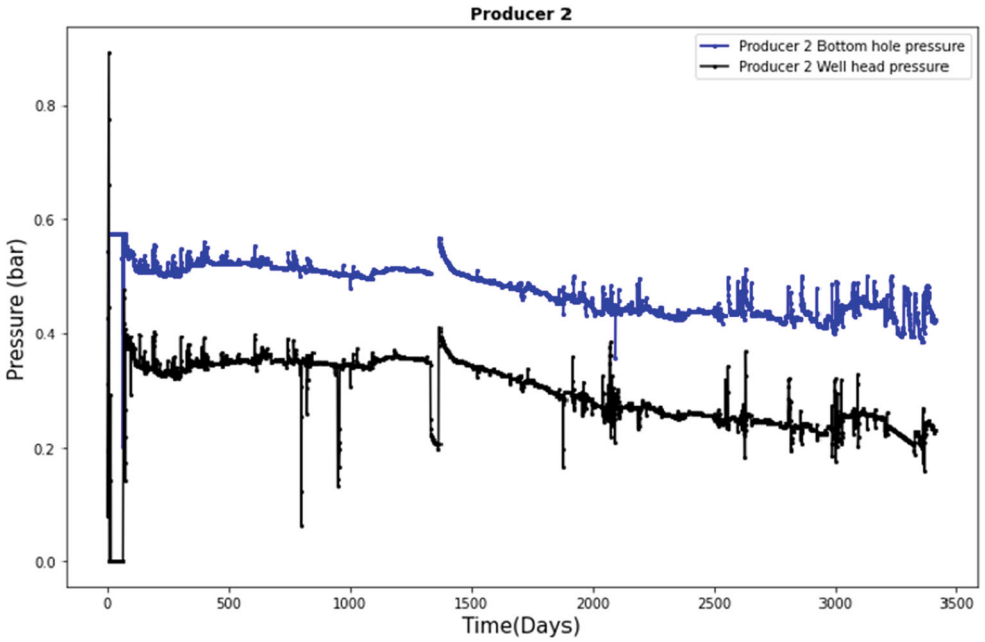

Fig. 26 Producer 2 bottom hole pressure and well head pressure)

the model is over-fitting. This effect should be seen on the test part, where the model is supposed to give poor results compared to the training part. Effectively and as it can be seen in Figs. 45, 46, 47, 48, 49, and 50 Deep-CRM (red curve) gives very poor results on the test part compared to the real data (blue curve). This behaviour is expected since the different producer' ANNs are only a function of time and thus in the future these models have no information about injection rate and BHP signal. On the other hand and in Fig. 44 we plot the evolution of the different losses after the integration of CRMs loss. We can see that the validation still decreasing when the training loss still decreasing. This behaviour indicates that the model is not over-fitting as it is the case with no CRMs physics. In Figs. 51, 52, 53, 54, 55 and 56 we recall the performance of Deep-CRM on the real dataset with CRMs physics integration.

The good performance of Deep-CRM with physics compared to Deep-CRMs with no physics is due mainly

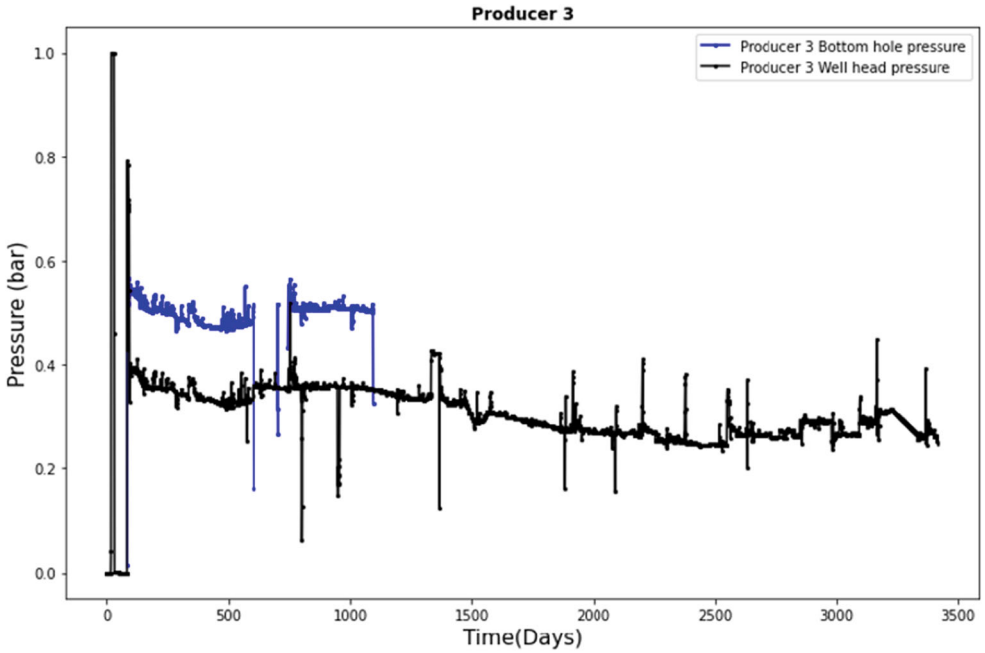

Fig. 27 Producer 3 bottom hole pressure and well head pressure)

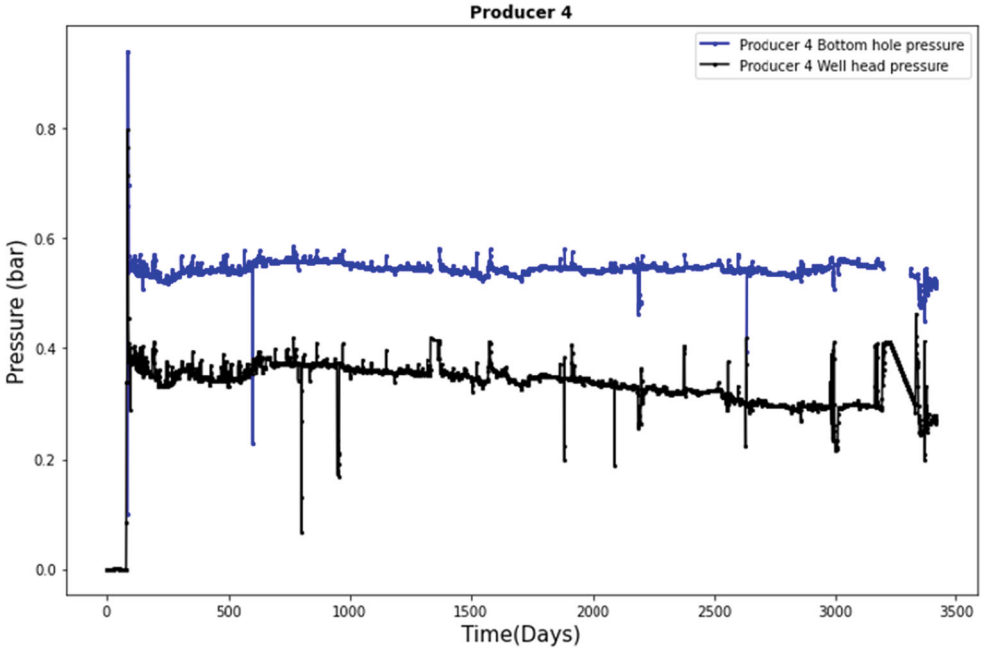

Fig. 28 Producer 4 bottom hole pressure and well head pressure)

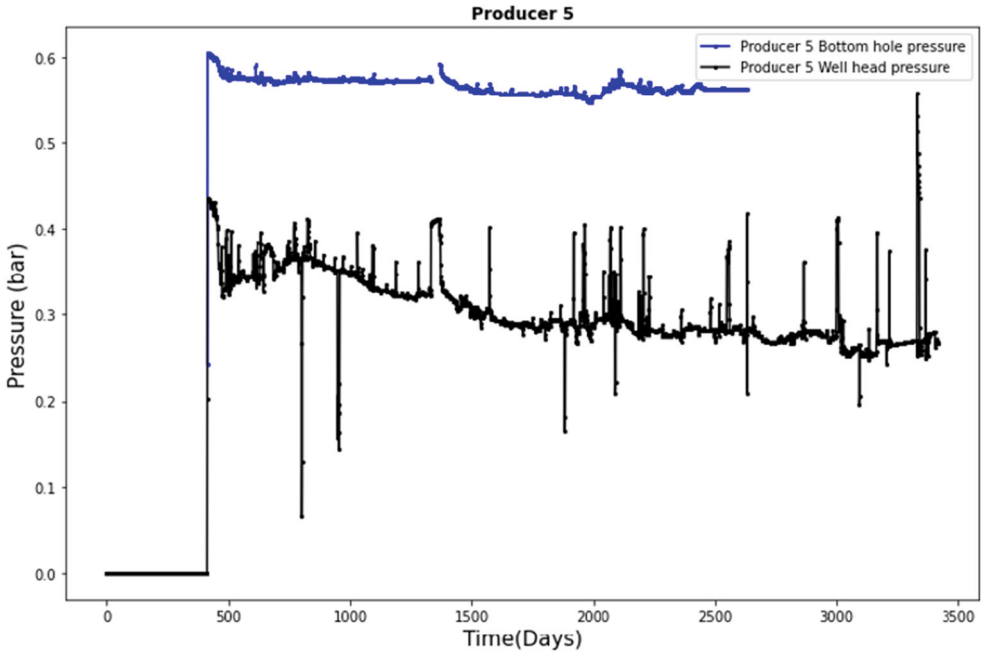

Fig. 29 Producer 5 bottom hole pressure and well head pressure)

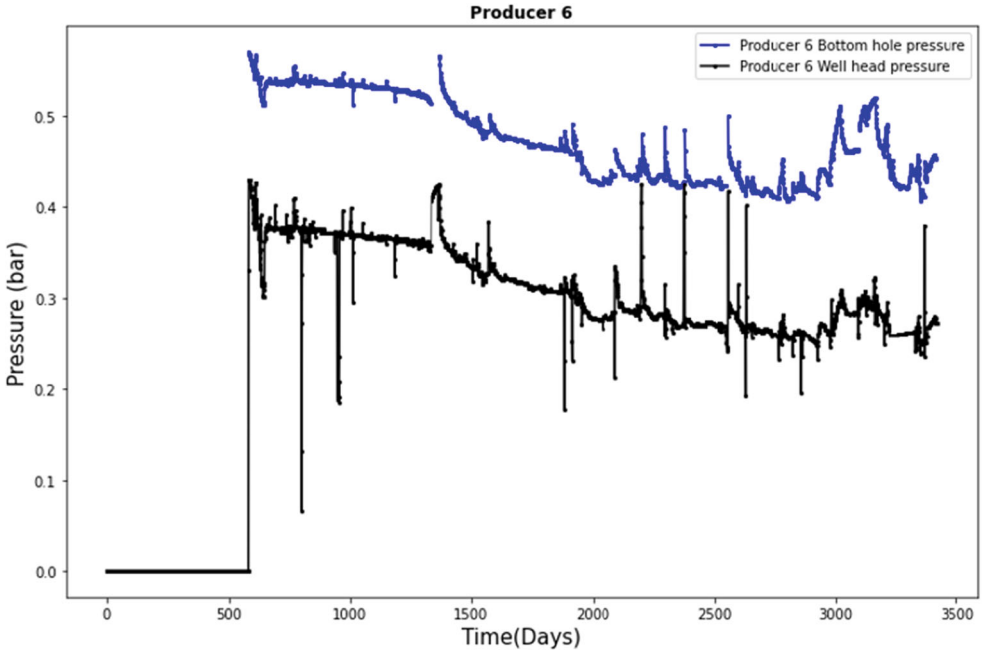

Fig. 30 Producer 6 bottom hole pressure and well head pressure)

Table 6 Real dataset Connectivity Table

	Producer 1	Producer 2	Producer 3	Producer 4	Producer 5	Producer 6
Injector 1	Strong Connection	No Connection	Weak Connection	Strong Connection	No Information	Weak Connection
Injector 2	Weak Connection	Weak Connection	No Information	No Information	No Information	Weak Connection
Injector 3	Strong Connection	No Information	No Information	No Connection	No Information	No Information
Injector 4	No Information	No Connection	Weak Connection	Weak Connection	Strong Connection	No Connection
Injector 5	Weak Connection	Weak Connection	No Connection	No Connection	No Connection	No Information

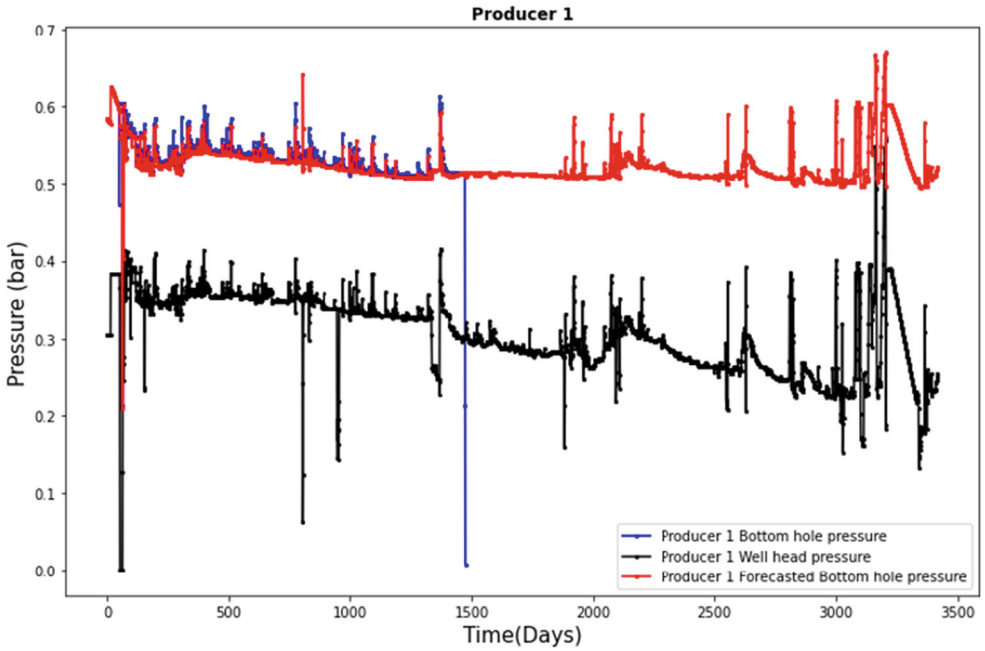

Fig. 31 Producer 1 smoothed BHP

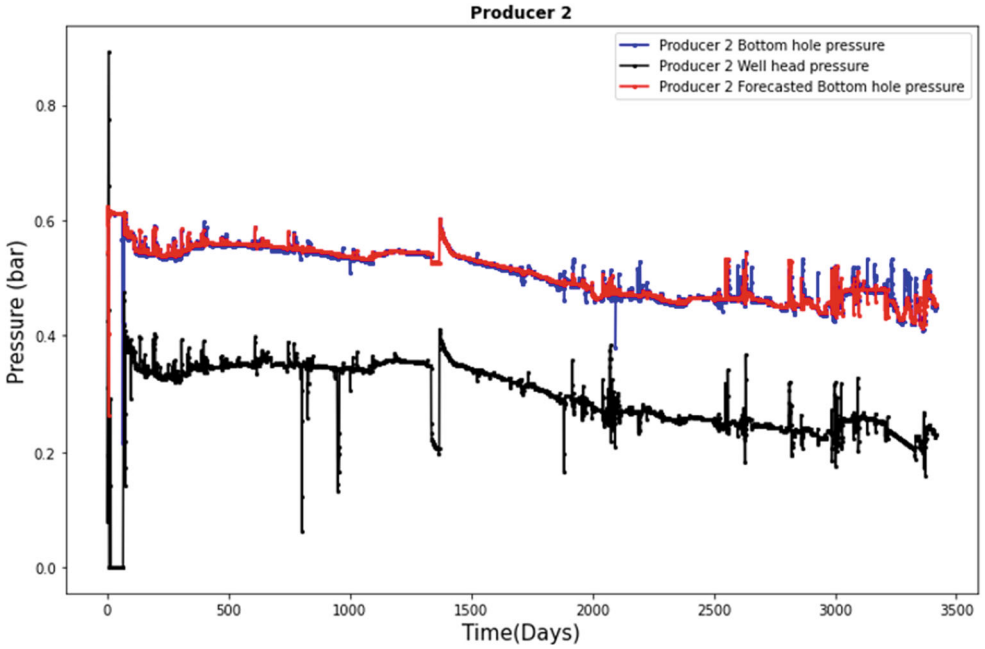

Fig. 32 Producer 2 smoothed BHP

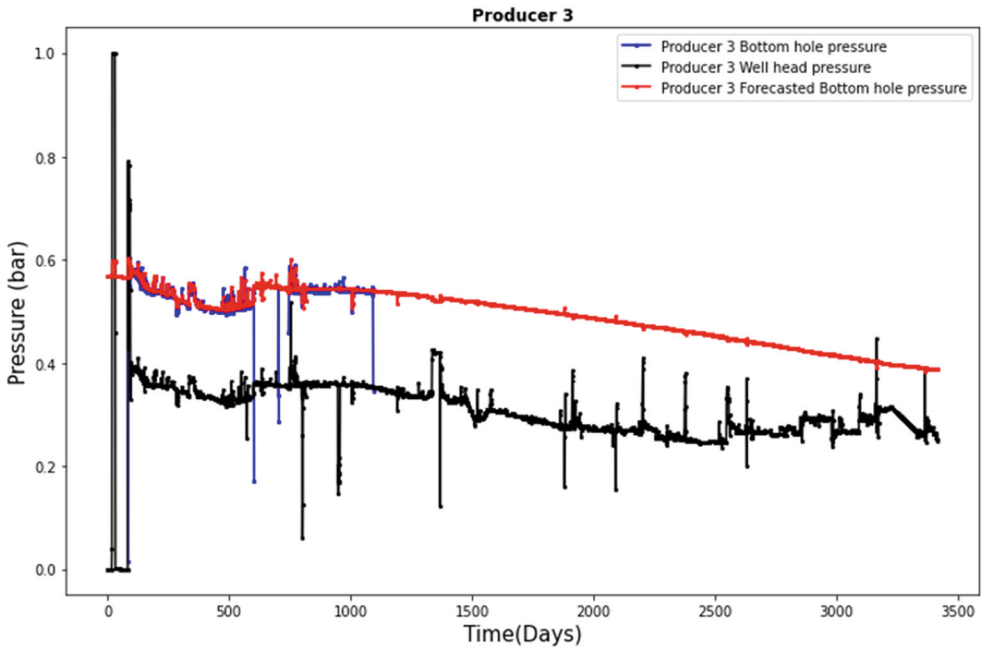

Fig. 33 Producer 3 smoothed BHP

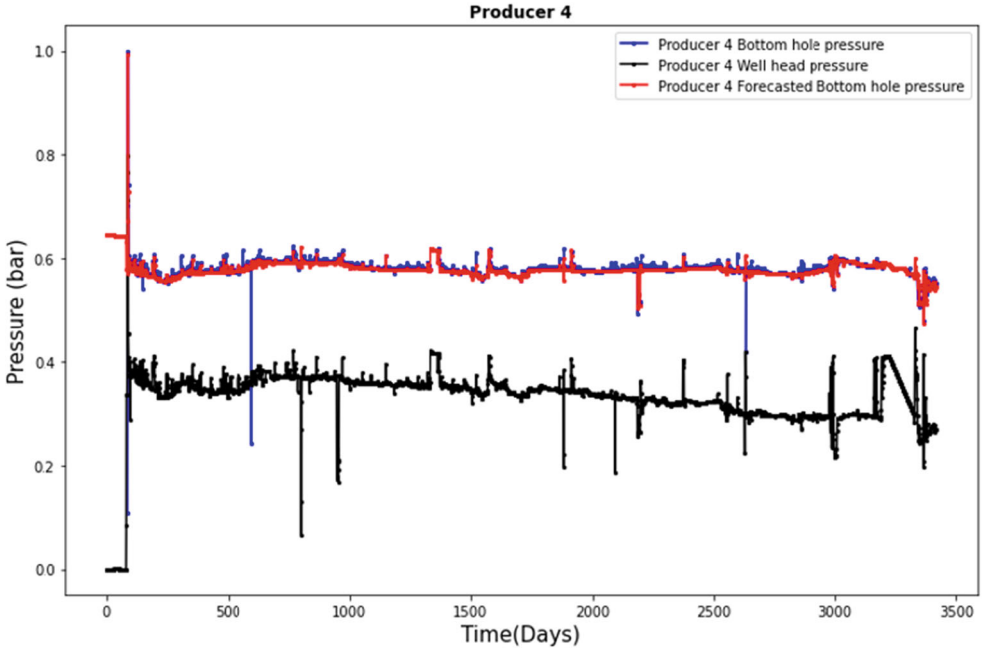

Fig. 34 Producer 4 smoothed BHP

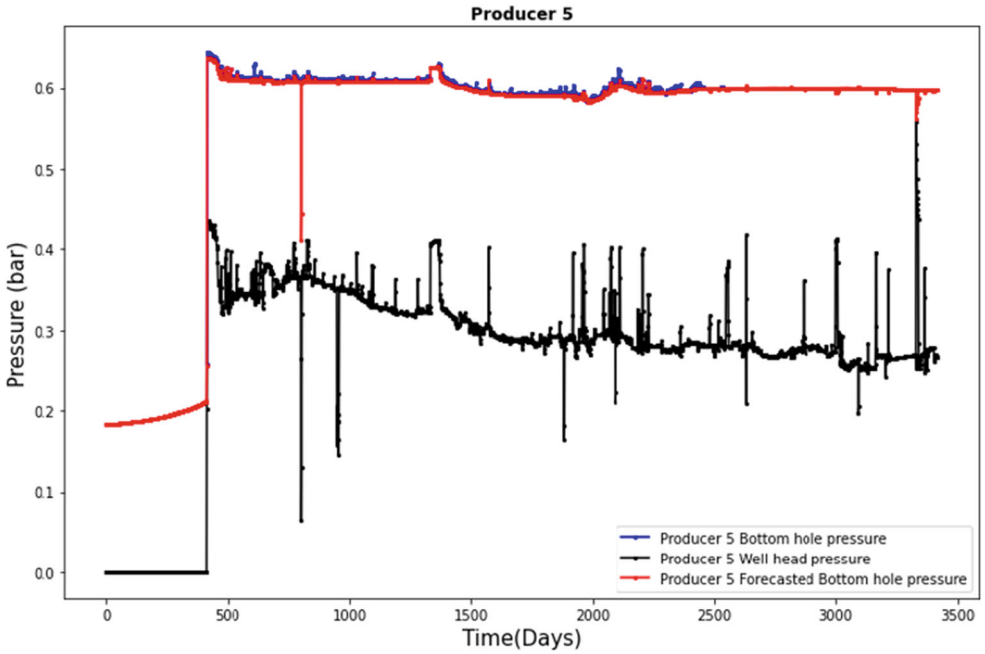

Fig. 35 Producer 5 smoothed BHP

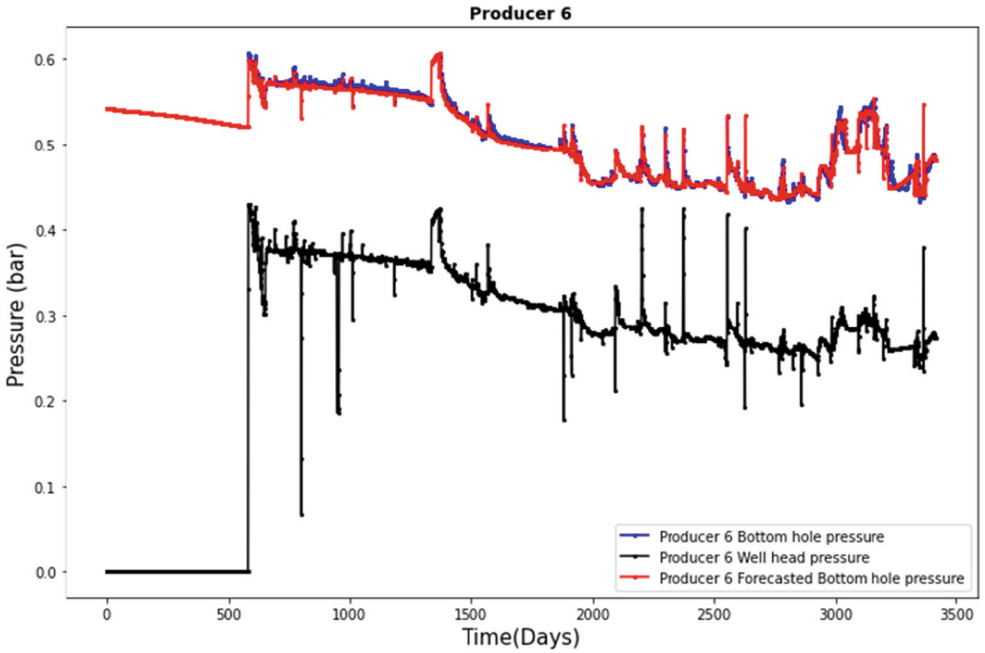

Fig. 36 Producer 6 smoothed BHP

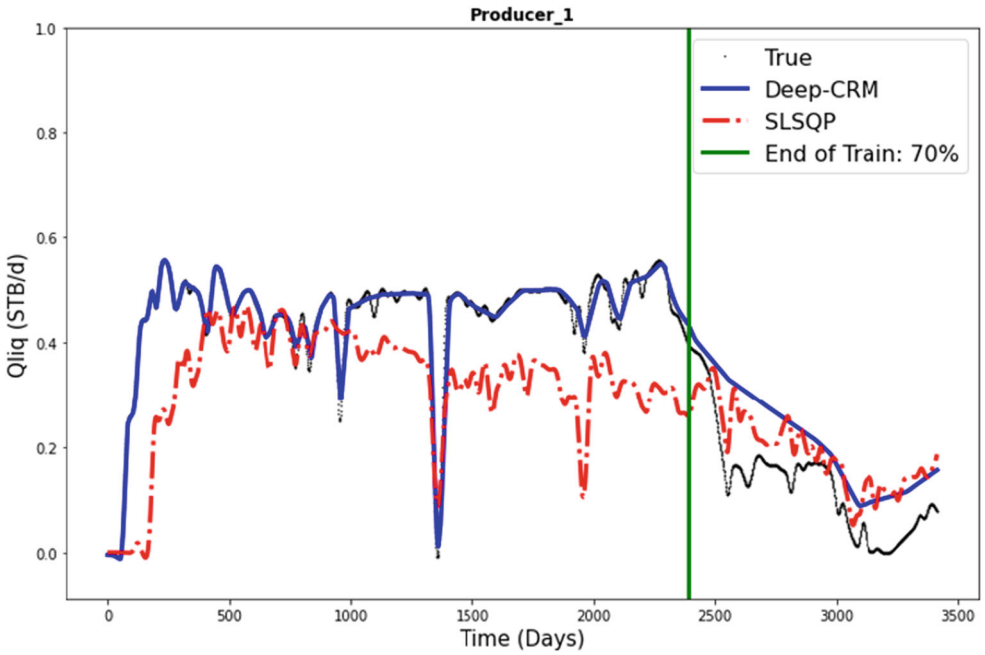

Fig. 37 Deep-CRM forecasts (blue, continuous line) on real dataset (dark, dot line), and SLSQP forecasts (red, dotted line), green line limits train and test data

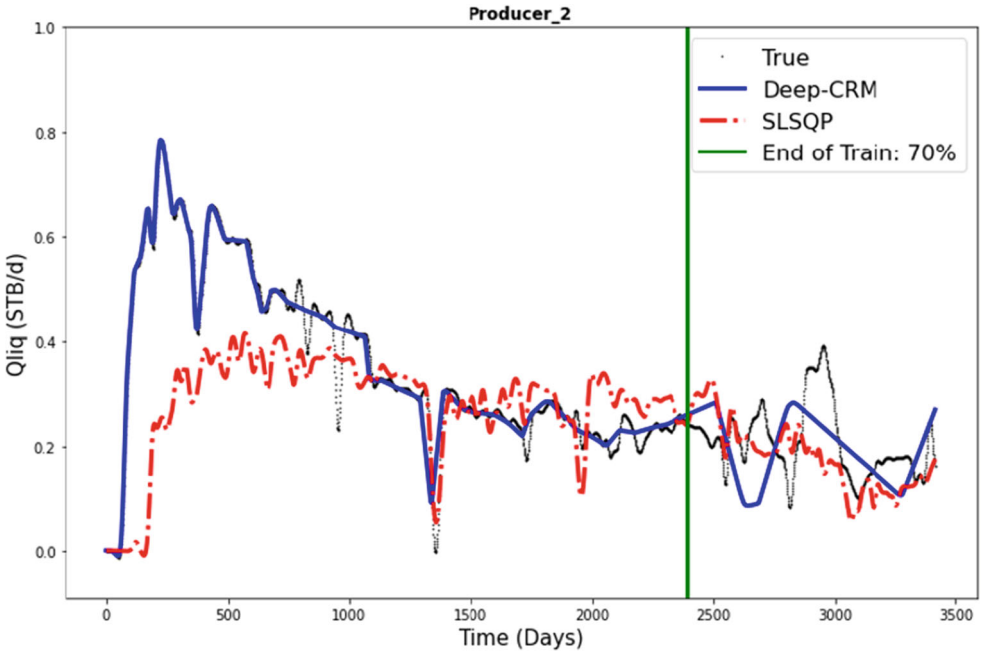

Fig. 38 Deep-CRM forecasts (blue, continuous line) on real dataset (dark, dot line), and SLSQP forecasts (red, dotted line), green line limits train and test data

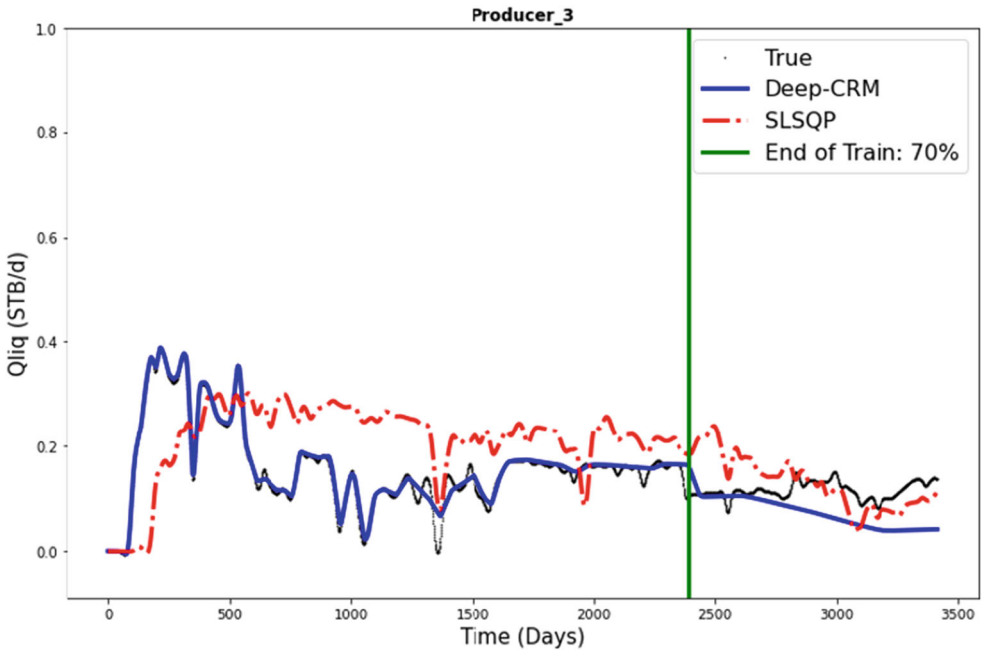

Fig. 39 Deep-CRM forecasts (blue, continuous line) on real dataset (dark, dot line), and SLSQP forecasts (red, dotted line), green line limits train and test data

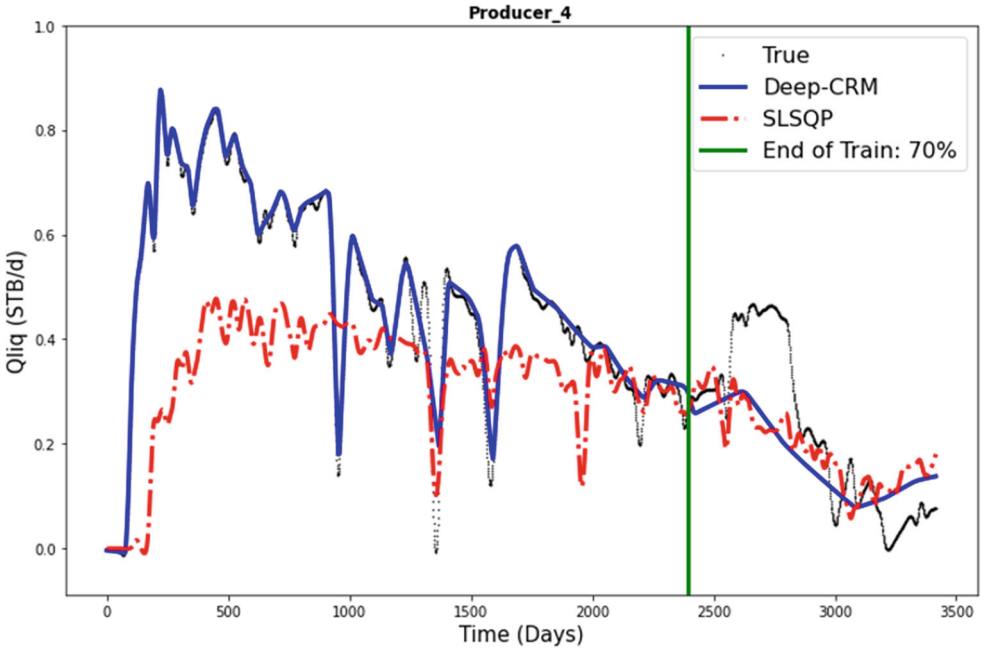

Fig. 40 Deep-CRM forecasts (blue, continuous line) on real dataset (dark, dot line), and SLSQP forecasts (red, dotted line), green line limits train and test data

to the integration of CRMs physics. In fact thanks to such physics the model different total production rates ANNs are trained to respect future injections rates and future bottom hole pressure value.

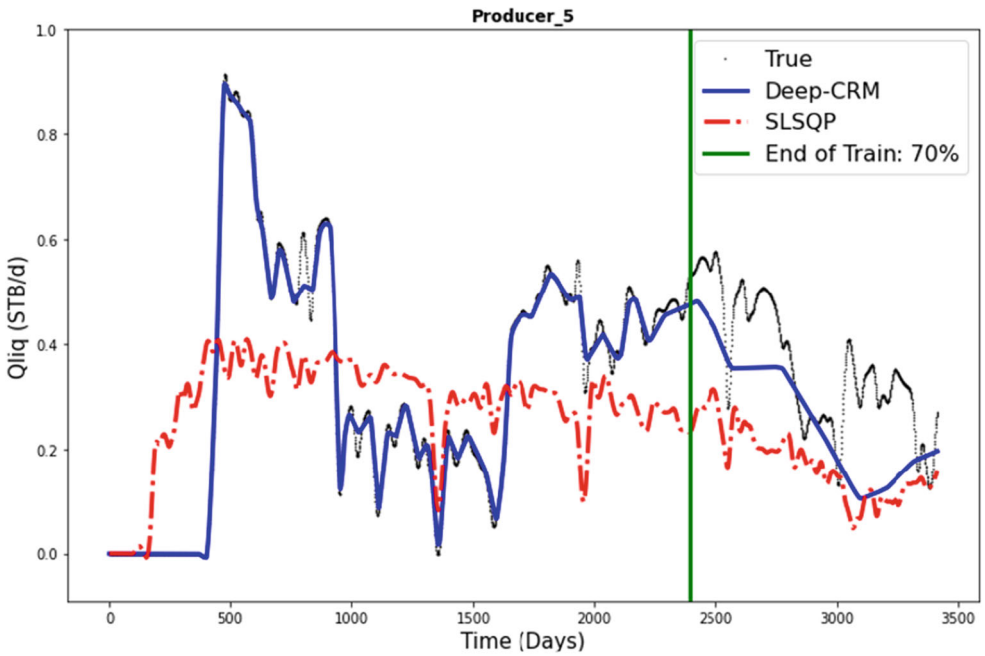

Fig. 41 Deep-CRM forecasts (blue, continuous line) on real dataset (dark, dot line), and SLSQP forecasts (red, dotted line), green line limits train and test data

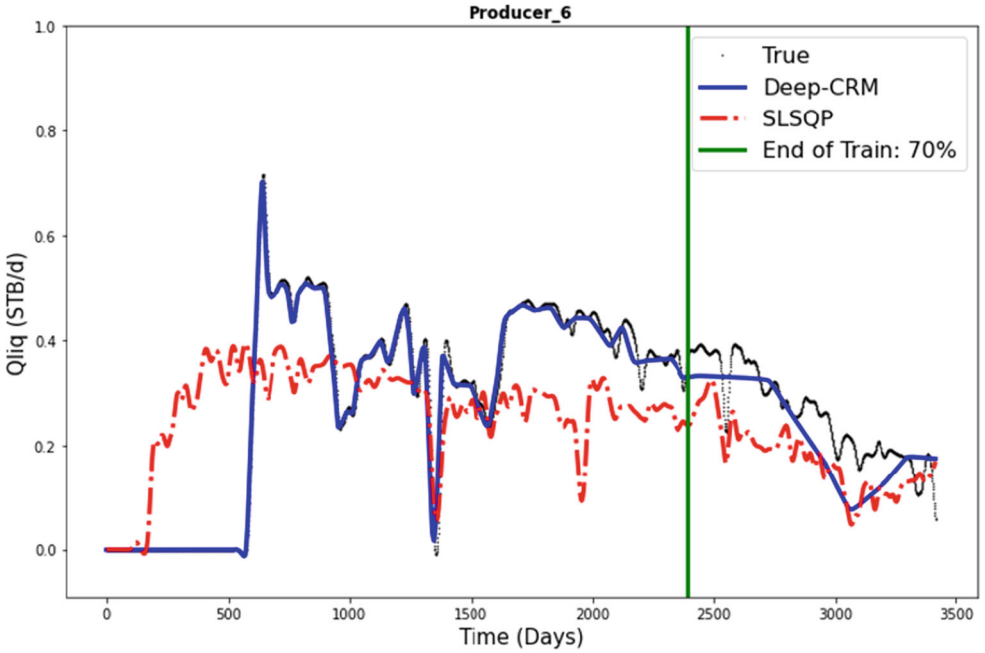

Fig. 42 Deep-CRM forecasts (blue, continuous line) on real dataset (dark, dot line), and SLSQP forecasts (red, dotted line), green line limits train and test data

Table 7 Mean Absolute Error Comparison between Deep-CRM and CRM (SLSQP)

	Producer 1	Producer 2	Producer 3	Producer 4	Producer 5	Producer 6	Mean
Deep-CRM	0.14	0.05	0.03	0.05	0.10	0.03	0.06
CRM (SLSQP)	0.07	0.06	0.04	0.08	0.18	0.08	0.09

Table 8 Connectivity between Producers and Injectors on real dataset Deep-CRM

	Producer 1	Producer 2	Producer 3	Producer 4	Producer 5	Producer 6	Sum
Injector 1	0.13	10^{-5}	0.04	0.20	0.29	0.31	0.97
Injector 2	0.14	0.14	0.11	0.19	0.19	0.21	0.98
Injector 3	0.30	10^{-5}	0.03	0.005	0.40	0.24	0.97
Injector 4	0.31	0.0003	0.08	0.47	0.11	10^{-5}	0.97
Injector 5	0.19	0.56	0.0007	0.0013	10^{-5}	0.003	0.75

Table 9 Connectivity between Producers and Injectors on real dataset with CRM SLSQP

	Producer 1	Producer 2	Producer 3	Producer 4	Producer 5	Producer 6	Sum
Injector 1	0.06	0	0.0122	0	0.104	0.036	0.20
Injector 2	0.383	0.126	0.027	0.173	0.142	0.057	0.90
Injector 3	0.0069	0	0.015	0.00	0.214	0.112	0.34
Injector 4	0.206	0	0.02	0.03	0.10	0	0.33
Injector 5	0.03	0.32	0	0	0	0.01	0.36

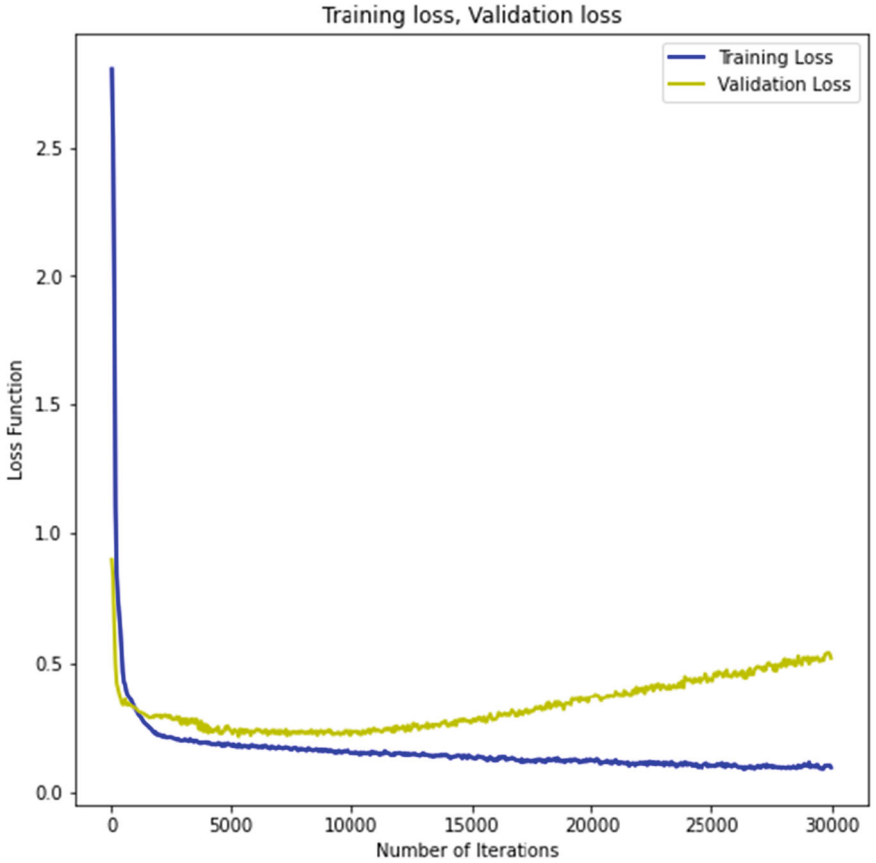

Fig. 43 Training loss, validation loss, and CRMs loss. No physics integration

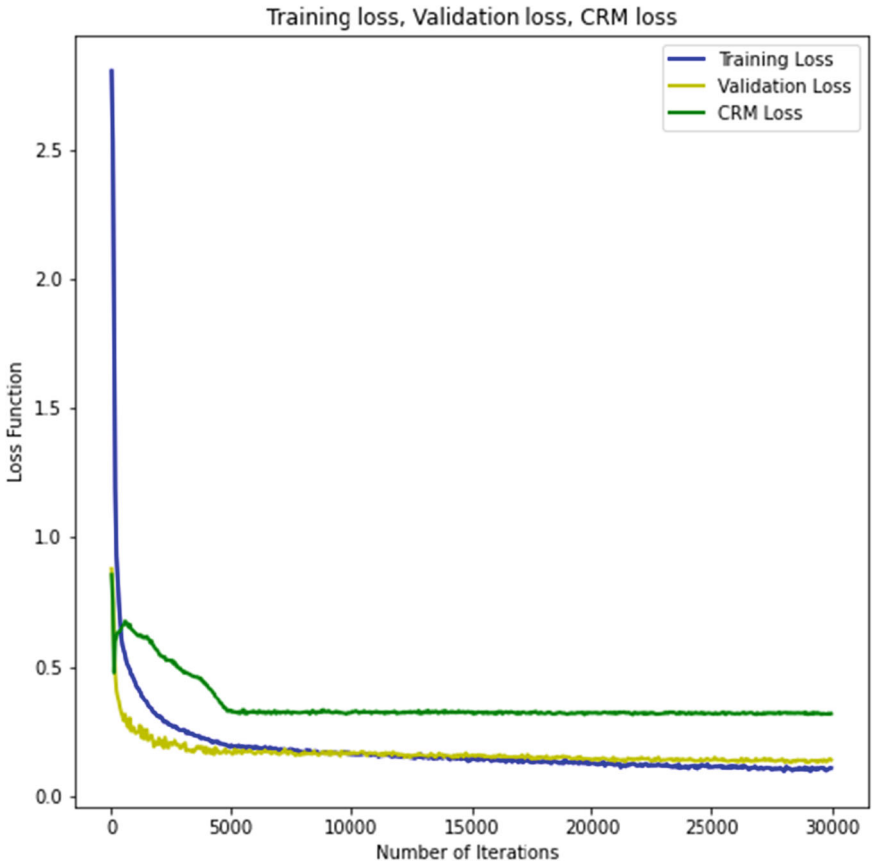

Fig. 44 Training loss, validation loss, and CRMs loss. With physics integration

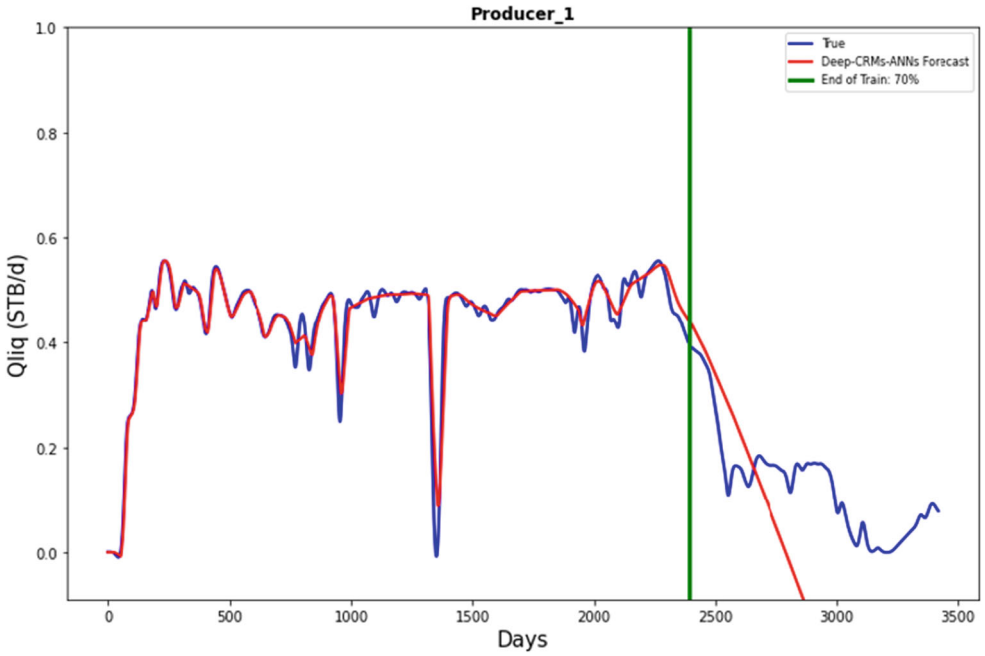

Fig. 45 Deep-CRM with no CRMs physics

5 Conclusions and future work

In this work, we have presented a new approach Deep-CRM, based on PINNs, to identify CRM parameters and to perform production rate forecasting. Compared to the classic CRM analytical solution approach the new method does not require any assumptions on injections and on producers bottom hole pressure. Deep-CRM was tested on two datasets: in the first synthetic dataset we show that Deep-CRM can explain the underlying geology (e.g. presence of a fault in SONDOUS) and perform better forecasting than the analytical solution using the SLSQP https://docs.scipy.org/doc/scipy/reference/optimize.minimize-slsqp.html optimizer. In the second real field dataset, Deep-CRM provide better results than CRM with SLSQP both in terms of forecasting and parameter identification.

Future work will involve testing the method on a larger dataset with potentially more injectors and producers

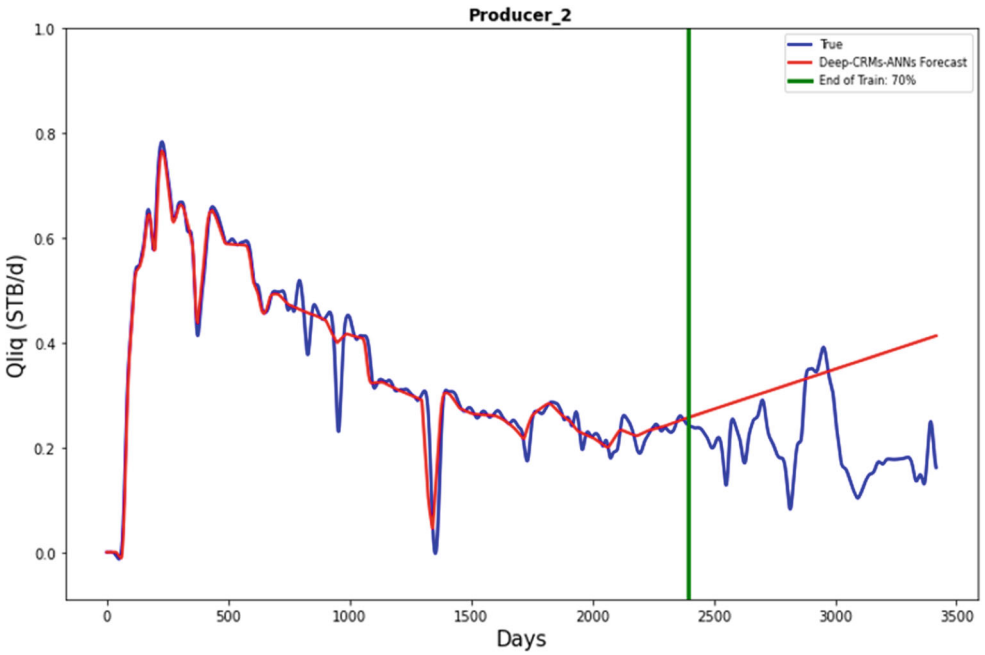

Fig. 46 Deep-CRM with no CRMs physics

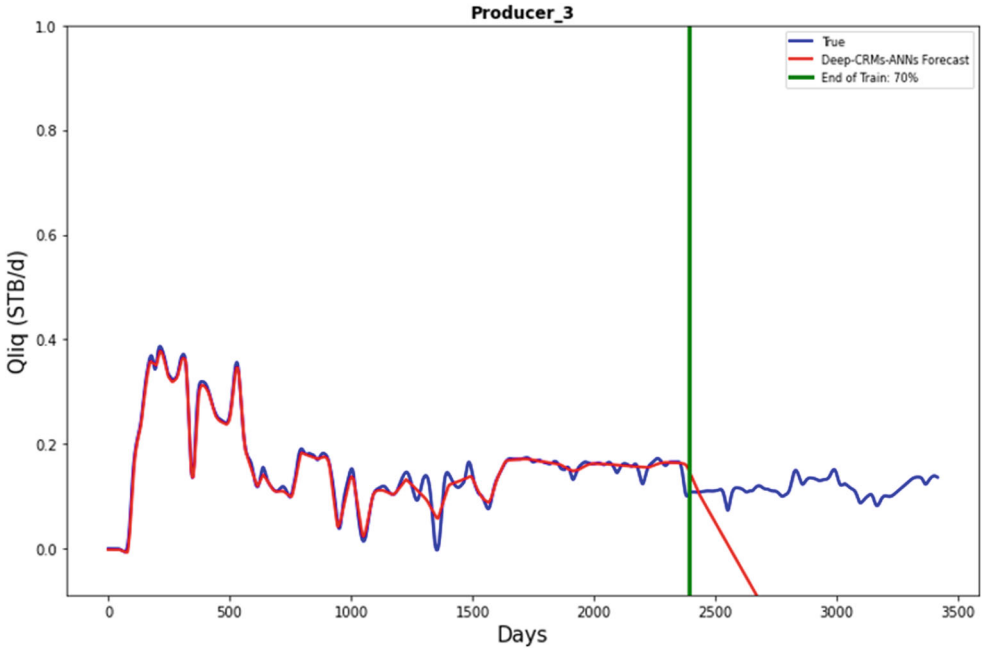

Fig. 47 Deep-CRM with no CRMs physics

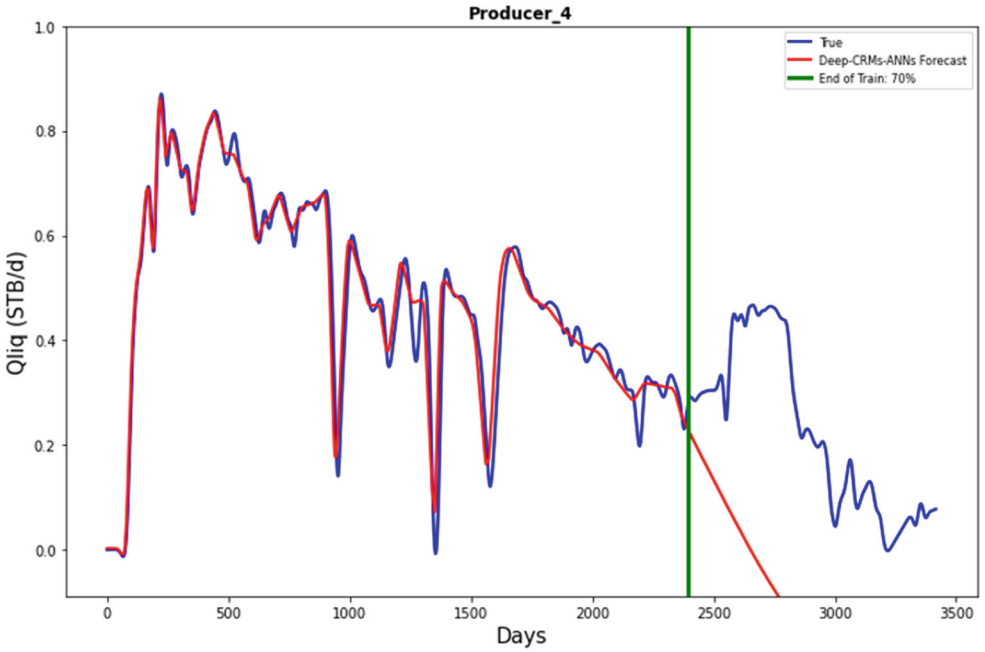

Fig. 48 Deep-CRM with no CRMs physics

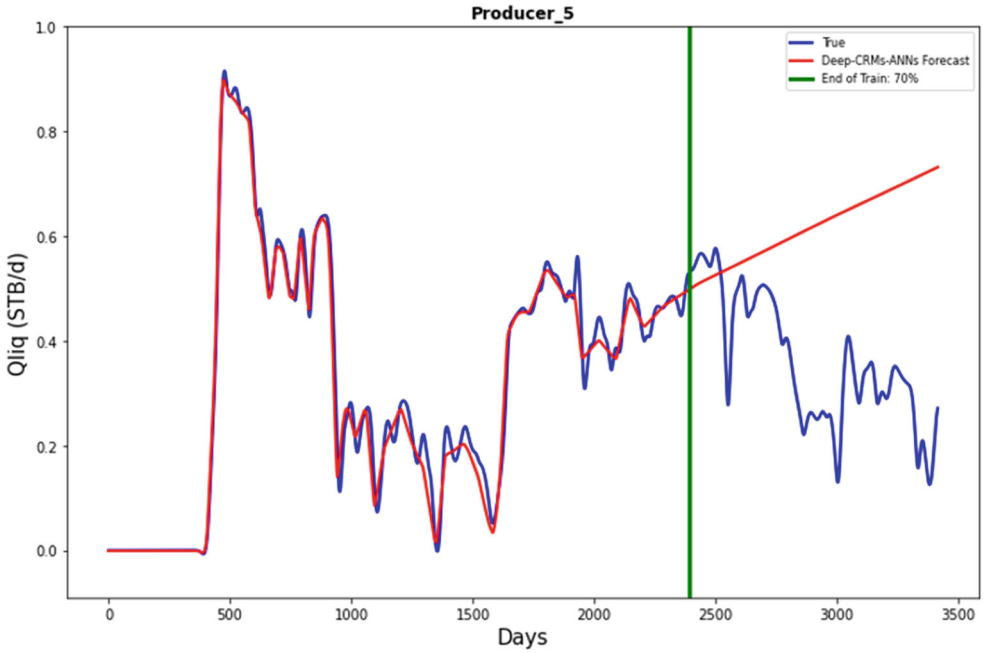

Fig. 49 Deep-CRM with no CRMs physics

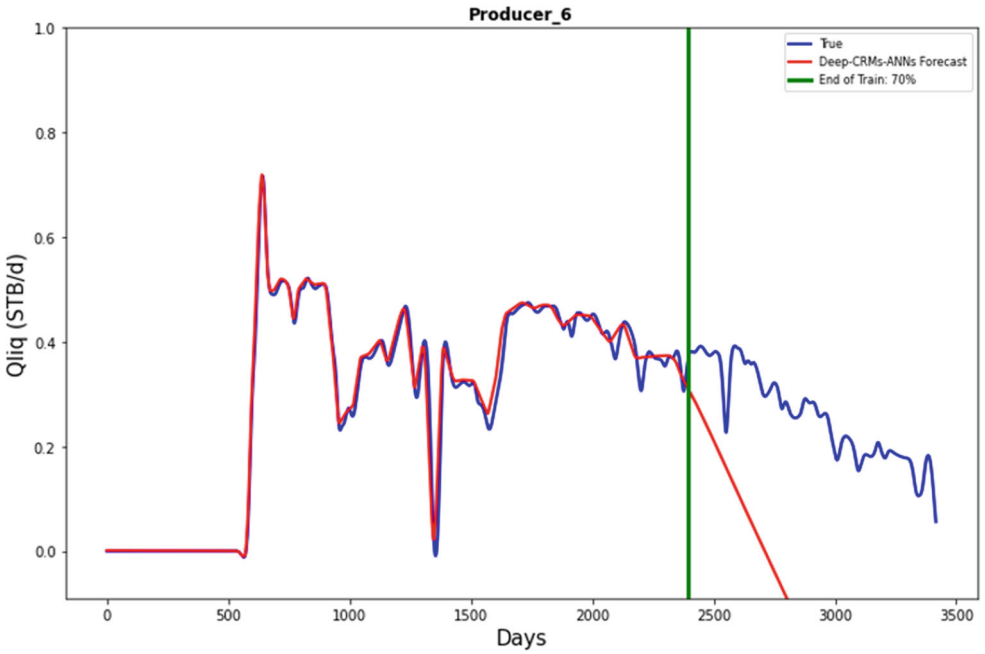

Fig. 50 Deep-CRM with no CRMs physics

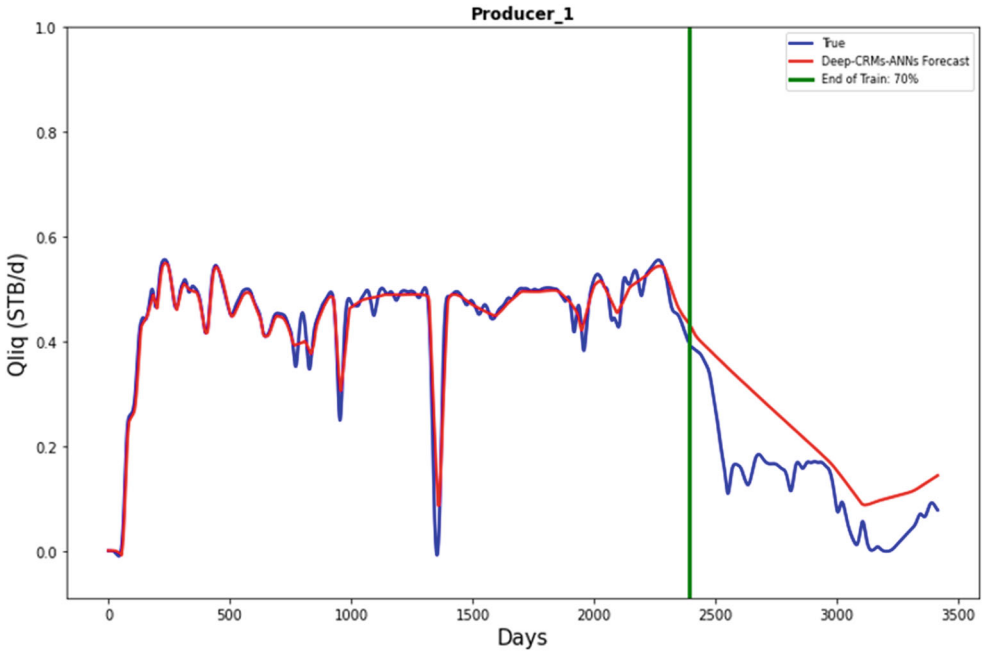

Fig. 51 Deep-CRM with CRMs physics

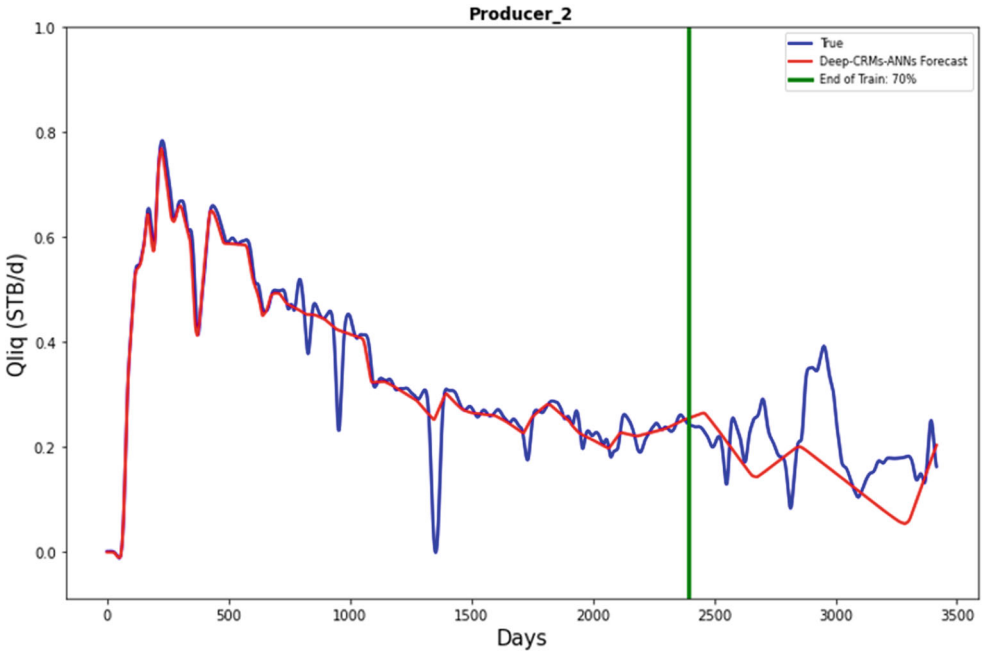

Fig. 52 Deep-CRM with CRMs physics

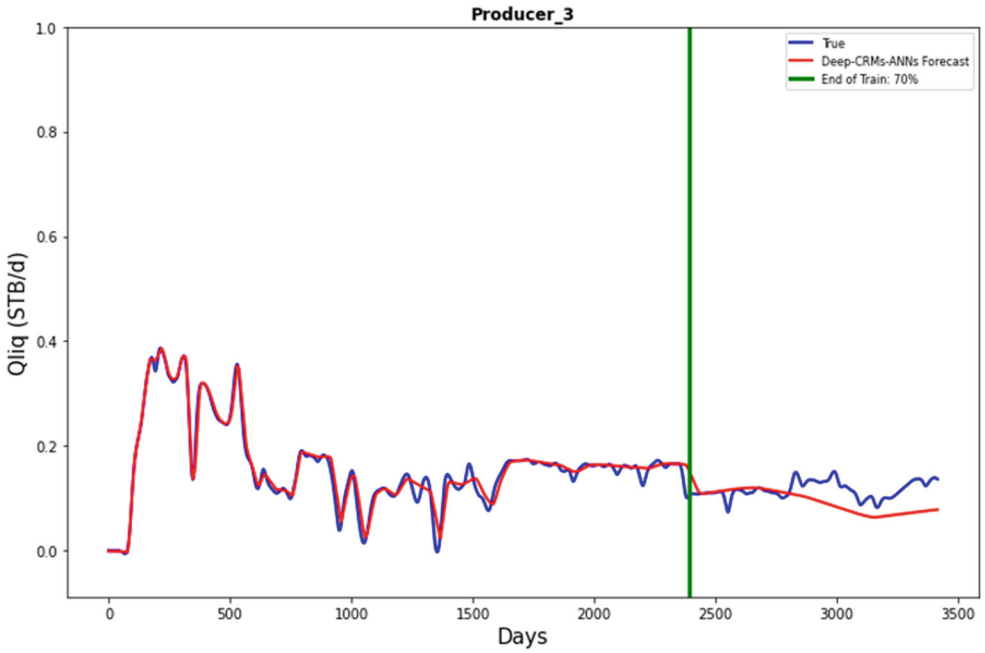

Fig. 53 Deep-CRM with CRMs physics

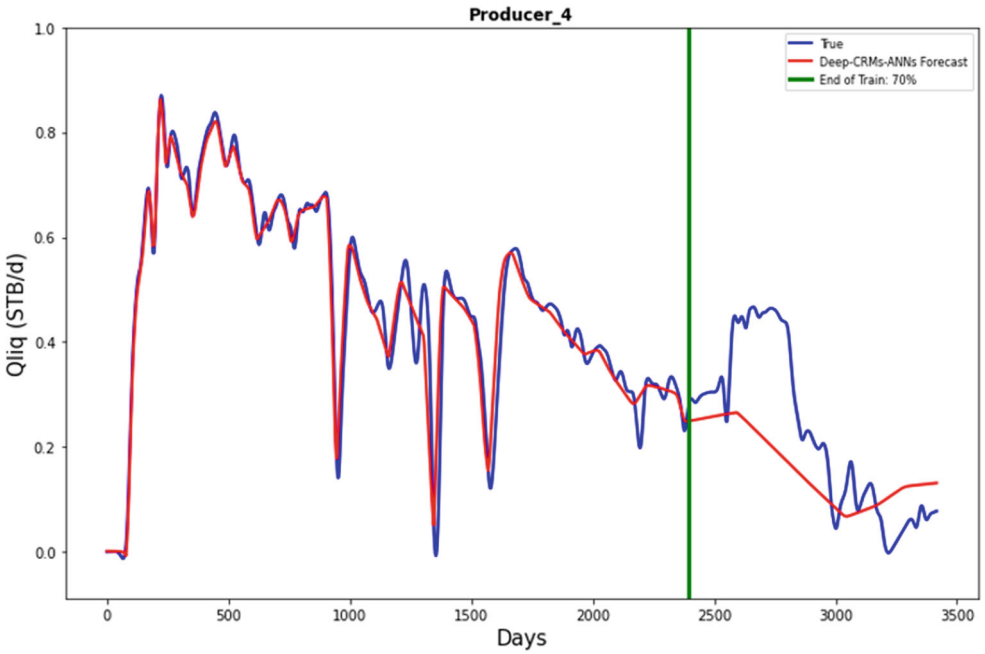

Fig. 54 Deep-CRM with CRMs physics

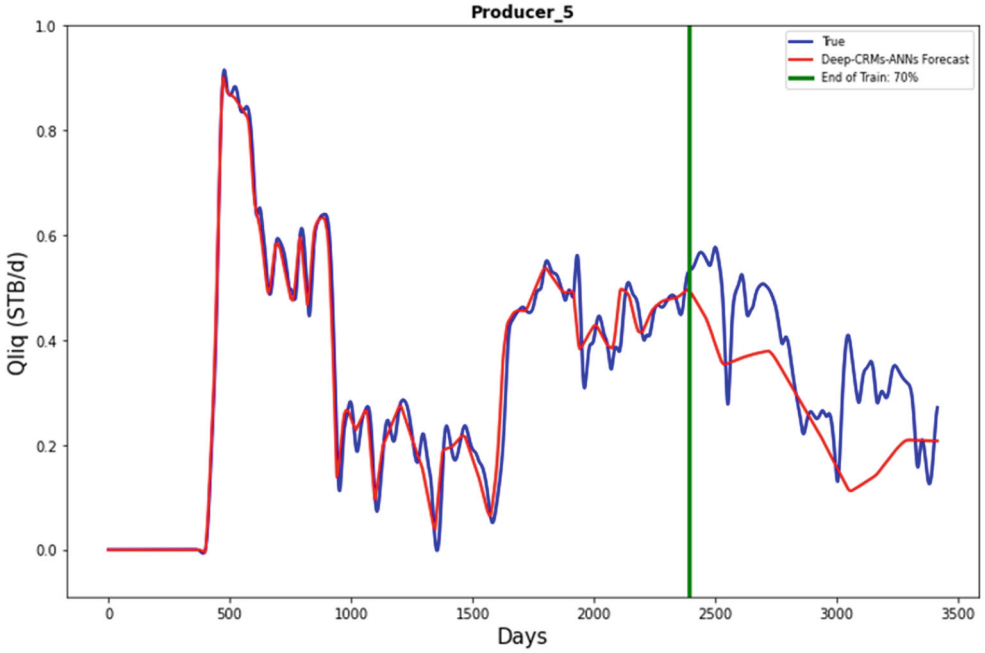

Fig. 55 Deep-CRM with CRMs physics

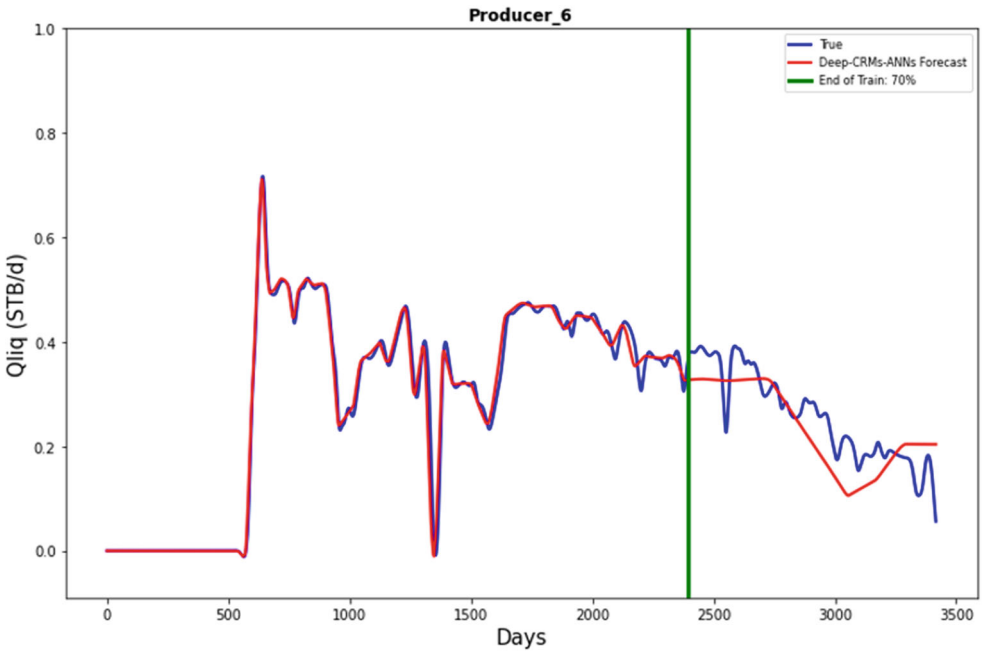

Fig. 56 Deep-CRM with CRMs physics

couple and taking into account uncertainty using different solutions already available for handling uncertainty in neural networks (see for instance [15]).

References

1. Wanderley de Holanda, R., Gildin, E., Jensen, J., Lake, L., Kabir, C.: State-of-the-art literature review on capacitance resistive models for reservoir characterization and performance forecasting. Energies (2018)
2. Yousef, A., Gentil, P.: Jensen, J. A capacitance model to Infer Interwell connectivity from production and injection rate fluctuations. Soc. Pet. Eng., Lake, L. (2006)
3. Al-Yousef, A.: Investigating Statistical Techniques to Infer Interwell Connectivity from Injection and Production Rate Fluctuations. Ph.D. Dissertation, University of Texas, Austin, TX, USA (2006)
4. Sayarpour, M., Zuluaga, E.: Kabir C. The use of capacitance-resistive models for rapid estimation of waterflood performance. Soc. Pet. Eng., Lake L (2007)
5. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning, pp. 20–206. MIT Press, Cambridge (2016)
6. Panda, M.N., Chopra, A.K.: An Integrated Approach to Estimate Well Interactions. Society of Petroleum Engineers India Oil and Gas Conference and Exhibition, New Delhi (1998)
7. Cheng, H., Vyatkin, V., Osipov, E., Zeng, P., Yu, H.: LSTM Based EFAST Global Sensitivity Analysis for Interwell Connectivity Evaluation Using Injection and Production Fluctuation Data. In: IEEE Access, vol. 8, pp. 67289–67299 (2020). https://doi.org/10.1109/ACCESS.2020.2985230
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural. Comput. **9**(8), 1735–1780 (1997)
9. Saltelli, A., Tarantola, S., Chan, K.P.-S.: A quantitative model-independent method for global sensitivity analysis of model output. Technometrics **41**(1), 39–56 (1999)
10. Raissi, M., Perdikaris, P., Karniadakis, G.: Physics informed deep learning (part I): Data-driven solutions of nonlinear partial differential equations. arXiv:1711.10561 (2017)
11. Raissi, M., Perdikaris, P., Karniadakis, G.: Physics informed deep learning (part II): Data-driven discovery of nonlinear partial differential equations. arXiv:1711.10561 (2017)
12. Geneva, N.: Zabaras N. Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. J. Comput. Phys. (2020)
13. Zhu, Y., Zabaras, N., Koutsourelakis, P.S., Perdikaris, P.: Physics-Constrained Deep learning for high dimensional surrogate modelling and uncertainty quantification without labelled data. J. Comput. Phys. **394**, 56-81 (2019)
14. Yang, C., Yang, X., Xiao, X.: Data-driven projection method in fluid simulation. Comp. Anim. Virtual Worlds **27**, 415–424 (2019)
15. Kendall, A.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. Proceedings of the IEEE conference on computer vision and pattern recognition, Cipolla, R. (2018)
16. Holanda, R.W.D., Gildin, E., Jensen, J.L.: Improved Waterflood Analysis Using the Capacitance-Resistance Model within a Control Systems Framework. In: Proceedings of the SPE Latin American and Caribbean, Petroleum Engineering Conference, Quito, Ecuador, pp. 18–20. Society of Petroleum Engineers: Richardson, TX, USA (2015)
17. Mehlig, B.: Artificial Neural Networks. Arxiv (2019)
18. Baydin, A., Pearlmutter, B., Radul, A., Siskind, J.: Automatic differentiation in machine learning: a survey. arXiv:1502.05767Comment (2015)
19. Bishop, C.M.: Neural Networks for Pattern Recognition (2006)
20. Akinsete, O., Adesiji, B.A.: Bottom-Hole Pressure Estimation from Wellhead Data Using Artificial Neural Network SPE-198762-MS (2019)
21. Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D.J.: 1D convolutional neural networks and applications: A survey. Mech. Syst. Signal Process. **151** (2021)

Affiliations

Abderrahmane Yewgat[1,2] · Daniel Busby[1] · Max Chevalier[2] · Corentin Lapeyre[3] · Olivier Teste[2]

Daniel Busby
Daniel.busby@totalenergies.com

Max Chevalier
Max.chevalier@irit.com

Corentin Lapeyre
Lapeyre@cerfacs.fr

Olivier Teste
Olivier.Teste@irit.com

1 TotalEnergies R&D, Pau, France

2 Toulouse Computer Science Research Institute (IRIT), Université Toulouse III - Paul Sabatier, Toulouse, France

3 European Center of Research and Advanced Training in Scientific Computing (CERFACS), Toulouse, France