Journal of Petroleum Science and Engineering 210 (2022) 109937

Contents lists available at ScienceDirect

Journal of Petroleum Science and Engineering

journal homepage: www.elsevier.com/locate/petrol

Data-driven deep-learning forecasting for oil production and pressure

Rafael de Oliveira Werneck [a,*], Raphael Prates [a], Renato Moura [a], Maiara Moreira Gonçalves [b], Manuel Castro [a], Aurea Soriano-Vargas [a], Pedro Ribeiro Mendes Júnior [a], M. Manzur Hossain [b], Marcelo Ferreira Zampieri [b], Alexandre Ferreira [a], Alessandra Davólio [b], Denis Schiozer [b,c], Anderson Rocha [a]

[a] RECOD.ai, Institute of Computing, University of Campinas – UNICAMP, 13083-852 Campinas, SP, Brazil
[b] CEPETRO, University of Campinas – UNICAMP, 13083-970 Campinas, SP, Brazil
[c] School of Mechanical Engineering, University of Campinas, 13083-970 Campinas, SP, Brazil

ARTICLE INFO

Keywords:
Forecasting
Data-driven
Deep learning
Oil production
Pre-salt

ABSTRACT

Production forecasting plays an important role in oil and gas production, aiding engineers to perform field management. However, this can be challenging for complex reservoirs such as the highly heterogeneous carbonate reservoirs from Brazilian Pre-salt fields. We propose a new setup for forecasting multiple outputs using machine-learning algorithms and evaluate a set of deep-learning architectures suitable for time-series forecasting. The setup proposed is called *N-th Day* and it provides a coherent solution for the problem of forecasting multiple data points in which a sliding window mechanism guarantees there is no data leakage during training. We also devise four deep-learning architectures for forecasting, stacking the layers to focus on different timescales, and compare them with different existing off-the-shelf methods. The obtained results confirm that specific architectures, as those we propose, are crucial for oil and gas production forecasting. Although LSTM and GRU layers are designed to capture temporal sequences, the experiments also indicate that the investigated scenario of production forecasting requires additional and specific structures.

1. Introduction

Managing hydrocarbon reservoirs is challenging as it requires integrating different areas of knowledge such as reservoir and production engineering and geosciences. Through field data and production measurements, we gain insights into the reservoir behavior and can perform estimations for the reservoir's future, e.g., production forecasting (Ertekin and Sun, 2019). Reservoir simulation models are the most common tool used to forecast the production of petroleum fields. Although they are a consolidated tool to assist the decision-making process, they have some drawbacks, especially for short-term contexts. These models can also be very time-consuming depending on the field size and the complexity of the reservoir. These issues are significant for some Brazilian pre-salt fields since they are giant reservoirs formed by highly heterogeneous carbonate rocks under a complex production strategy such as WAG (water alternating gas) injection. These reservoir models are commonly used for long-term decisions, which involve several years of production forecast. On the other hand, short-term forecasting requires specific tuning of model properties to avoid high production fluctuations in the transition period from past to future, a common issue observed given the changes in the operational controls used in both periods. Therefore, the short-term forecast is usually done by analytics or machine-learning approaches (Tadjer et al., 2021).

More recently, the use of data-driven approaches to perform production forecasts has gained attention. These approaches only consider field response data and machine-learning techniques to perform forecast (Kubota and Reinert, 2019; Davtyan et al., 2020; Liu et al., 2020; Zhong et al., 2020), which are critical for pre-salt and unconventional reservoirs (Sun et al., 2018). A data-driven approach can be of important aid (directly or as a complement) to model-based approach approximations, providing production forecasts within a window of several days or weeks.

Accurate forecasting is an essential part of a reservoir's operation, as it helps engineers make proper designs and developments for the field (Liu et al., 2020). However, it is difficult to predict well production and bottom-hole pressure (BHP) accurately, as the reservoir properties and its dynamics influence them, i.e., injection of gas and water, maintenance of reservoir pressure, and interference from other wells. In addition, the well monitoring data are non-linear, non-stationary, non-parametric, noisy, and of a chaotic nature.

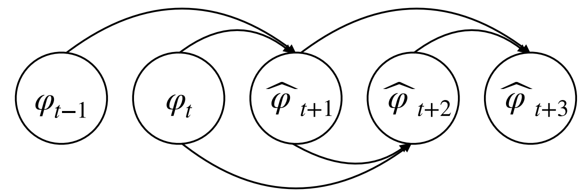
* Corresponding author.
E-mail address: rafael.werneck@ic.unicamp.br (R.d.O. Werneck).

https://doi.org/10.1016/j.petrol.2021.109937
Received 15 April 2021; Received in revised form 3 November 2021; Accepted 6 November 2021
Available online 13 December 2021

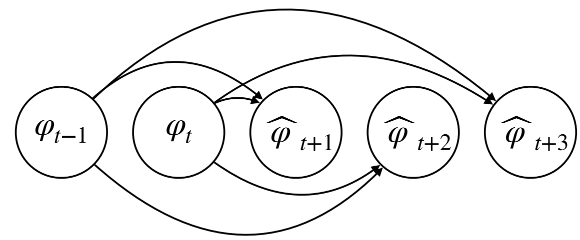
This work focuses on data-driven approaches to perform well production and pressure forecasting in the short term. We aim to combine past information from several sensors from producer and injector wells into an end-to-end solution. The solution is expected to predict several days, shaping a multiple-output regression model based on multivariate time-series data. The short-term oil and pressure forecasting scenarios are much longer than the traditional scope in the machine-learning literature. Therefore we must leverage state-of-the-art machine-learning techniques and appropriately adapt them to cope with these challenges (Ertekin and Sun, 2019). In this case, we compare machine-learning with analytics approaches and not model-based ones.

We propose: (a) a new forecasting setup focused on the prediction window's last day, which considers the challenging problem of forecasting for several days; (b) the use of stacked Recurrent Neural Networks (RNN) in forecasting, allowing them to focus on different timescales, and (c) a discussion on the use of off-the-shelf methods to forecast oil production and pressure.

This paper can be divided into three main parts: Information and Theory, which comprises the Introduction and Forecasting in the literature, respectively Sections 1 and 2. The work performed is represented in Sections Section 3 (Proposed Method), 4 (Experimental protocol), and 5 (Experiments and Results). Finally, we present the Conclusions and Future Work in Section 6.



(a) Iterated prediction setup.



(b) Direct prediction setup.

Fig. 1. Common setups on multi-step-ahead and multi-outputs according to Bontempi (2008). The example considers two input and three output sequences. $\hat{\varphi}$ denotes the predicted values.

2. Forecasting in the literature

This paper investigates forecasting for oil production and bottom-hole pressure (BHP) of producer wells in a carbonate reservoir, considering only data from the floating production storage and offloading units (FPSOs). Traditional methods in the oil and gas industry leverage numerical reservoir models to perform production forecasts, a model-based approach known to be time-consuming and computationally expensive. The literature for forecasting well production based on data-driven features, mostly with deep learning techniques, is still scarce and not well documented (Cao et al., 2016; Kubota and Reinert, 2019; Zhan et al., 2019; Davtyan et al., 2020; Liu et al., 2020; Zhong et al., 2020).

This section presents different setups for time-series forecasting in the prior art that considers a broader horizon, i.e., a setup that performs multiple output predictions for a target variable. The multiple output prediction aims at predicting a sequence of two or more data points based on a sequence of input data. We discuss the scarce literature on data-driven oil production forecasting. Finally, we describe off-the-shelf and state-of-the-art methods for forecasting general data that could be adapted to the present problem.

2.1. Forecasting setups

The literature of time-series forecasting lacks a default setup on how to perform forecasts considering multiple outputs. Bontempi (2008) describes the two most common approaches to multiple outputs as a conditional distribution over input and output sequences. Fig. 1 depicts these setups in a graphical model, where Fig. 1(a) is the iterated prediction, and Fig. 1(b) is the direct prediction. The iterated prediction is an iterative one-step-ahead prediction, in which the dependencies are preserved, but the error is propagated throughout the iterations. On the other hand, the direct approach has different models for each next step, making it a conditional independent problem.

Brownlee proposed another setup for multiple outputs on his website (Brownlee, 2018). In this approach, each prediction window's output is concatenated in a sequence, considering a one-step sliding window, and then evaluated. The pros of this approach are that every multiple output prediction is considered when calculating the metrics. However, it is impossible to have a plot representation of the prediction results for this method, as it has more than one result for each data point.

Pao (2007) proposed a rolling cross-validation setup, in which the testing set is separated into folds. The training for each fold is composed of the previous data in the time series. This approach is similar to what we call *retraining*. When using retraining, we improve the learning model in each step of the testing by incorporating the most recent data points. However, one disadvantage of this method is that retraining in each step is time-consuming.

Bedi and Toshniwal (2019) studied a different technique. They used an output window of multiple outputs and slid the window the size of the output for the next prediction with the ground-truth values as input. Thus, they combined their results at the end of the prediction and presented a plot of their predictions. In this method, they avoid multiple predictions for the same day. However, they combined different confidences of the prediction when concatenating the last day of a window with the first day of the next window.

We detail these different setups in the following figures. For each figure, one must consider each column as a day in the test set and the different rows as different predictions. The concatenation approach performs its multiple output prediction and slides the window by one day to perform the next prediction. All predictions are then concatenated and the representation is used to calculate the metrics. In Fig. 2, we have three predictions (blue, green, and red, with yellow representing each input) of three days, and they are concatenated at the end. Hence, the last day of the first prediction (day 3 in blue) is succeeded by the first day of the second prediction (day 2 in green), and so on. This final concatenation is used to calculate the metrics of this setup.

We included a yellow strip to represent the training set in the retraining setup, showing that we have a larger training set for each forecast. The method retrains before performing the next prediction. In the end, the predictions complete the test set (last row), and we then evaluate the method. Fig. 3 depicts this setup.

Finally, the sliding window setup performs a prediction for multiple outputs without intersections between the columns, i.e., if the first prediction predicts day 1 to day 3, the second one predicts day 4 to day 6. Fig. 4 illustrates this method. These different setups and their problems (time-consuming and combining different confidences of prediction) motivated us to discuss and propose a new, more compatible setup with the challenge of forecasting for longer periods.

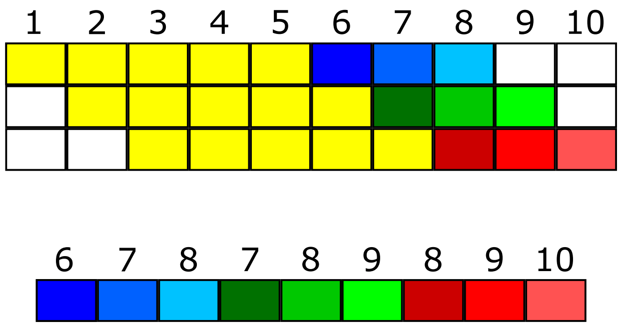


Fig. 2. Concatenation setup. For each prediction (lines in blue, green, and red), their values are concatenated in a new line, in which each number corresponds to a day (last blue day followed by first green day) to calculate the measures of the forecast. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

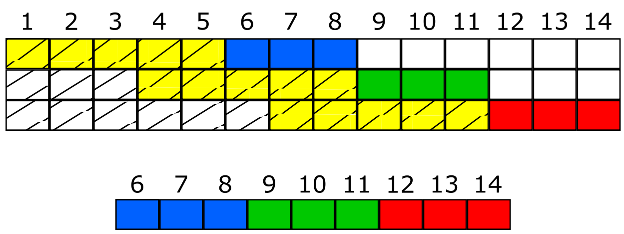


Fig. 3. Retraining setup. After each forecast, the method includes the ground-truth value of the predicted part and trains again for the next output window, represented by the hashed blocks.

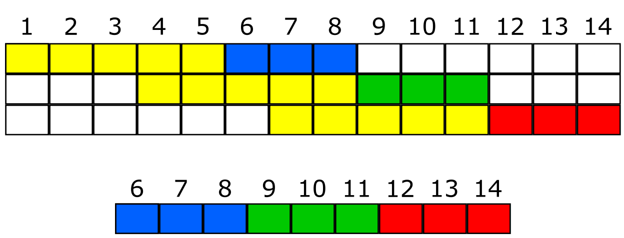


Fig. 4. Sliding window setup. For each prediction, there is no intersection between the forecast output windows.

2.2. Forecasting oil production

Some authors already addressed the problem of the production forecast using data-driven algorithms. Kubota and Reinert (2019) tackled this issue using linear regression techniques along with recurrent neural networks. The authors only considered three sets of time series: injection history, production history, and the number of producers. They performed experiments for longer prediction horizons (12 months but on monthly data). The authors showed that a reliable production forecast could be made with data-driven models, without a geological model or numerical simulators. These findings motivated us to design the methods we investigate herein.

Davtyan et al. (2020) also used linear regression methods to forecast oil production, in this case, based on sliding windows. They combined features from aggregated characteristics of oil fields, local information of the wells, pressure features, and autoregressive features to perform their regression. They applied their approach on a monthly-grained dataset, predicting the following month. They also performed an experiment considering a longer time frame but creating m independent regression models for a horizon of size m. Unfortunately, they did not compare their method with any other forecasting approaches and only applied simple linear regression.

Departing from the linear approach formulation, Liu et al. (2020) performed production forecasts using an ensemble empirical mode decomposition (EEMD), followed by Long-Term Short-Term learning. They split two daily datasets into training and test sets, decomposed the training set using an EEMD to choose the basis functions, applied a Genetic Algorithm to select the hyperparameters of their methods, and tested on the test set. However, they only compared their approach with other simple approaches that are based on their own. There is no detailed information on how they performed the forecasting, whether it was daily or if they used a longer time frame.

Other works focus on Multilayer Perceptron (MLP) networks (Aizenberg et al., 2016; Amirian et al., 2018; Yuan et al., 2021) to perform forecasting. However, these approaches do not consider the information's propagation through time. The literature is more recently moving to Recurrent Neural Networks (RNN) (Song et al., 2020) and taking advantage of stacking RNN layers (Al-Shabandar et al., 2021; Chaikine and Gates, 2021), which are aligned with our proposed RNN. However, such approaches have fewer layers and do not consider multi-output — several days at a time.

In this line, some studies in the literature focus on long-term forecasting with RNNs. Pan et al. (2019) proposed two different scenarios that use specific types of LSTMs trained considering hidden periods of the time series. In the first scenario, a Denoising Long Short-Term Memory (DeLSTM) only considers oil production rate, followed by a Decline Curve Analysis (DCA) to perform the forecasting. The second scenario uses Savitzky-Golay Cascaded Long Short-Term Memory (SG-CLSTM) to smooth, fill hidden periods, and forecast. In this case, both oil production rate and pressure are used as input data. Although the results indicate a long-term production forecasting ability, they did not focus on the forecasting window or compare the forecasting results against other strategies.

Another type of recurrent network used for time-series forecasting is Echo State Networks (ESN), which is based on reservoir computing theory (Bianchi et al., 2021). Deng and Pan (2020) took advantage of ESN to capture the inherent dependencies among the wells and added empirical fractional-flow relationships to perform well-control optimization. However, the proposed approach is designed for mature fields where a certain amount of water has reached the producers, which is not the case of the pre-salt field in focus here.

Different from prior works, which are data-driven, Zhong et al. (2020) designed a proxy model using a conditional convolutional generative neural network to predict field production considering waterflooding for oil recovery. First, they used geostatistical methods to generate stochastic input parameters. Then, they applied a numerical simulator to obtain a train and test sets and trained the proxy model. Finally, they applied the material balance rule to calculate the oil production rate. The authors performed three experiments of their approach but they only compared it with the simulator model.

Kim and Durlofsky (2021) use an RNN-based proxy to predict different oil and water rates. It uses the well's BHP data sequence as input. Unlike our goal, the authors have trained the network with 256 simulated BHP profiles, and the obtained predictions are used as input for a constrained production optimization problem. The proposed RNN-based proxy model is based on a sequence-to-sequence LSTM layer, which is quite similar to what we propose in this paper. However, our network is formed by two stacked LSTM layers and several dense layers at the top, enabling it to capture diverse aspects of the input time series. In addition, we propose a multivariate input where BHP is one variable among others. It is therefore impossible to fairly compare it to our approach for forecasting.

Razak et al. (2021) proposed an encoder–decoder approach for long-term production forecasting combined with well properties and future controls of the producer. They also perform transfer learning, i.e., they train the network in a collection of historical production data from other wells and perform a fine-tuning for the target well. Thus, the model can exploit other dynamical trends to improve the generalization. After concatenating the input encodings, the decoder predicts oil, water, and gas as multivariate time series. Although the results indicate a long-term forecasting ability, they only perform 6-data-points-ahead forecasting and do not compare it with other strategies.

Most of these studies are data-driven, the same category of our proposed approach. Nevertheless, such methods comprise simple combinations of linear regressions or recurrent networks with fewer layers. This paper proposes solutions with deeper architectures and, most importantly, different (and more complete) evaluation setups. Therefore, we opted not to compare the proposed methods with the ones mentioned above but, instead, with more recent deep-learning derived methods and off-the-shelf solutions.

2.3. Baseline methods

We adopted two baselines for our experiments: Pressure-Normalized Decline Curve Analysis and a simple Recurrent Neural Network. Decline Curve Analysis (DCA) is a traditional method used in the oil and gas industry to predict future production (Belyadi et al., 2019) using a graphical procedure to analyze the declining production rates. Lacayo and Lee (2014) proposed a modified curve analysis method for unconventional reservoirs that did not achieve the pressure stabilized state. The Pressure-Normalized Decline Curve Analysis (PN-DCA) performs a decline curve analysis using pressure-normalized production rates. This pressure-normalized rate can be described by Eq. (1).

$$\Delta pN = \frac{q}{p_i - p_{wf}} \tag{1}$$

where q is the production rate, p_i the average initial reservoir pressure, and p_{wf} is the flowing pressure. This pressure-normalized rate ΔpN can be calculated by Eq. (2).

$$\frac{1}{\Delta pN} = m\sqrt{t} + b' \tag{2}$$

where m is the slope of the straight line and b' is where the curve intercepts the y-axis.

RNNs are one of the most promising techniques for time-series forecasting. Their main advantage is the presence of *memory cells* capable of propagating information through time. At a specific time-step t, the output h_t is calculated based on the current time-step input X_t and the previous time-step output h_{t-1}. The RNN network is a simple network composed of an RNN layer, followed by a single dense layer to our output size.

2.4. Off-the-shelf methods

As the literature of data-driven oil-production forecasting is still scarce, we also adopt state-of-the-art methods for general forecasting and compare them to our proposal. These methods are considered off-the-shelf, as they are general-purpose and not tailored for oil production forecasting.

Salinas et al. (2020) proposed the DeepAR, an autoregressive recurrent neural network for a probabilistic forecast. This method learns a global model for all time series in a dataset. The authors claim as an advantage are that the model learns seasonal behaviors across time series. They make a probabilistic forecast in the form of Monte Carlo samples learning from similar items. Moreover, their method can incorporate many likelihood functions to apply in the data. This DeepAR method was proposed to Amazon's retail businesses but was also evaluated on datasets of various problems.

Oreshkin et al. (2019) presented a neural architecture based on backward and forward residual links and fully connected layers for forecasting univariate time series. Their architecture is generic and straightforward; it does not rely on time series feature engineering; it is easy to interpret and extend. They also used ensembling to be comparable to other methods from the M4 forecasting competition. They evaluated their approach, called N-BEATS, on the M4 (Makridakis et al., 2018), M3 (Makridakis and Hibon, 2000), and Tourism (Athanasopoulos et al., 2011) datasets.

Vaswani et al. (2017) proposed an architecture based on attention mechanisms called Transformer. Their network model has an encoder–decoder structure using stacked self-attention and point-wise connected layers. As the model has neither recurrent nor convolution layers, they needed a positional encoding to have information about relative positions in the sequence. The Transformer architecture was evaluated on translation tasks, outperforming other architectures.

Taylor and Letham (2018) described a modular regression model that can be adjusted to different time series with the help of a specialist. Their method, named Prophet, uses a decomposable time series model to obtain three components: trend, seasonality, and holidays. The authors frame the problem as a curve-fitting exercise, as they claim it provides flexibility, no need for interpolating missing values, efficiency in fitting the model, and interpretable parameters. Taylor and Letham evaluated Prophet in business time series, specifically for Facebook Events.

3. Proposed method

In this section, we present our proposed approaches to deal with bottom-hole pressure and oil production forecasting. Our main contributions are divided into three fronts: first, in terms of validation, we propose a more realistic setup for comparing different forecasting methods when predicting multiple days. This setup allows us to plot results correctly and avoid mixing different prediction confidences in a long-range. Second, we introduce a series of pre-processing, data augmentation, and inclusion of injection data in the forecasting modeling. Finally, we propose methods to leverage cutting-edge deep-learning formulations for temporal data to tackle bottom-hole pressure and oil production forecasting. Fig. 5 presents a pipeline of our methodology.

3.1. Proposed evaluation setups

To avoid the problems raised in Section 2, we propose two forecasting setups for time series. The first, which we denominate *First Prediction*, slides a multi-output window one step at a time. This method obtains the first prediction for each test data point. We thus considered all the data from the first forecast window and, for the subsequent predictions, only the last data of the output. Fig. 6 details how this approach works. Considering each column a data point and each row the predictions of multiple outputs with a sliding window of one point each time, the *First Prediction* approach corresponds to the first prediction made for each data point. In this case, we select the three data predictions for the first forecasting and then the last data point of each subsequent prediction to compose our final forecast. We can plot the predictions with this setup, but it still combines different confidences considering the first prediction window.

Our second setup focused on the last prediction data, obtaining a result that is more compatible with the challenging problem of forecasting for longer periods. We name this approach *N-th Day*. In this case, we perform the same multi-output window with sliding steps of the previous setup but only consider each window's last day for the evaluation. Thus, the obtained results for this approach represent the most challenging forecast data, which is the most distant data in our output window from the input data. This setup has the advantage of not combining different prediction confidences in its results. However, it lacks all data points to plot, complemented by the *First Prediction* setup in this case.

This setup helps us to focus the evaluation on the behavior of the forecasting methods for the *N-th Day* prediction. Fig. 7 shows an example of this evaluation approach considering an output window of size 3. For each prediction (each row), each forecast's last predicted value (third data point) is selected to compose the final forecast series.

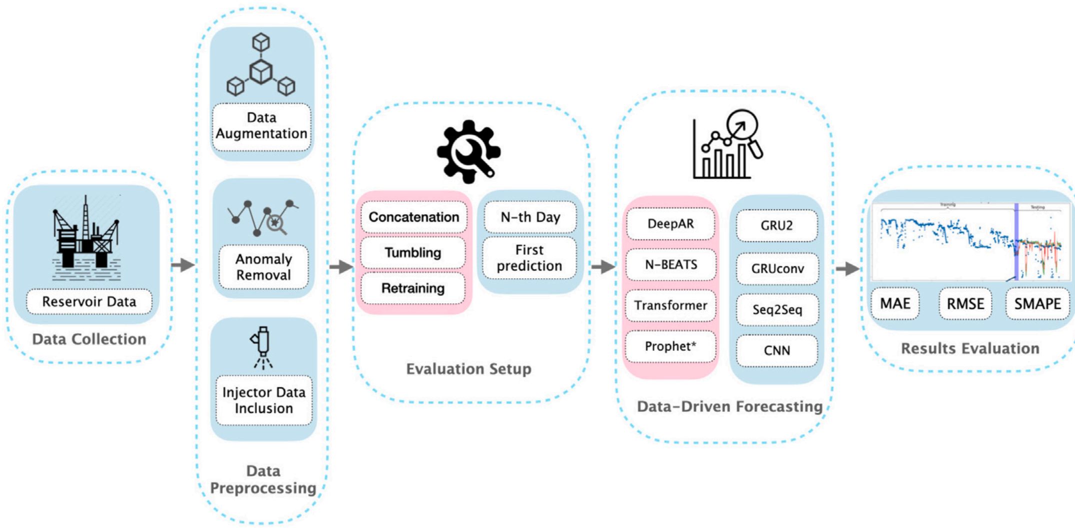


Fig. 5. Visual pipeline of our methodology.

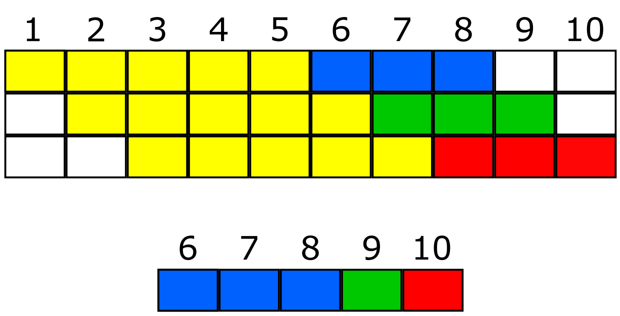


Fig. 6. The proposed First Prediction setup. For each prediction, we select the first prediction for each data. For the first prediction (blue), we select all predicted data and, for the subsequent predictions (green and red), we select the last data to form the last line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

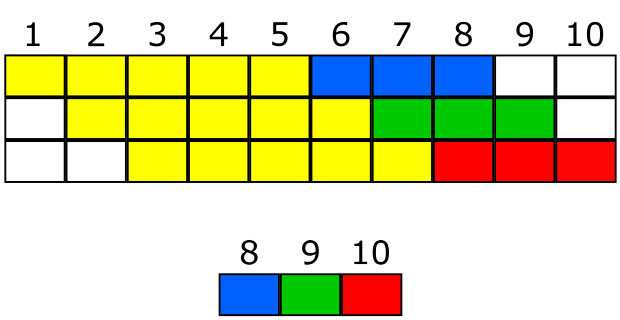


Fig. 7. The proposed N-th Day setup, in which we only select the last data (i.e., day 8 - blue, day 9 - green, and day 10 - red) for each forecast output window and create a new time series with these selected data. This new time series is the result for later assessment of accuracy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

While the *First Prediction* considers all the predicted data from the first prediction and then the last data from subsequent predictions, the *N-th Day* only considers the last data from all predictions. For long testing time series, both approaches tend to become comparable. In addition, the *N-th Day* prediction could be used to evaluate how the forecasting ability of the classifier decreases as the forecasting target goes further in the future.

3.2. Data pre-processing

Reservoir production time series might present some anomalies and even inconsistent or erroneous observations, e.g., due to unexpected events, defective observation or measurements, and human interventions. These events can alter the series, making it difficult for forecasting methods to learn a pattern for its prediction. Therefore, it is essential to identify and remove these anomalies in the data pre-processing. Data pre-processing is a vital step in supervised learning solutions.

For anomaly removal, we adopt z-score modeling. We calculate the mean and standard deviation for each time series and remove the data above or below one standard deviation from the mean.

However, after removing noisy data, we can have fewer data points than necessary for our forecast model to learn a time-series pattern. An approach to avoid this problem is performing data augmentation. Data augmentation comprises methods for increasing training data, including unobserved data related or transformed from the original training points (van Dyk and Meng, 2001). With more data, our methods and models can improve their learning.

We propose performing data augmentation using two approaches. The first is to create data points between two-time quanta with interpolation, e.g., between the data points **day 1 00:00** and **day 2 00:00**, this generates a point **day 1 12:00** through interpolation.

The second approach is to *perturb* the data during the training of the method. In this case, the training variability helps the method to be invariant to noise. In this approach, we set a probability of applying a transformation (a.k.a. perturbation) in subsets of the training data. The transformations could be: Add Noise, Convolution, Drift, Pool, Quantization, Reverse, and Time Warp. Table 1 describes each perturbation and Fig. 8 depicts these augmentations.

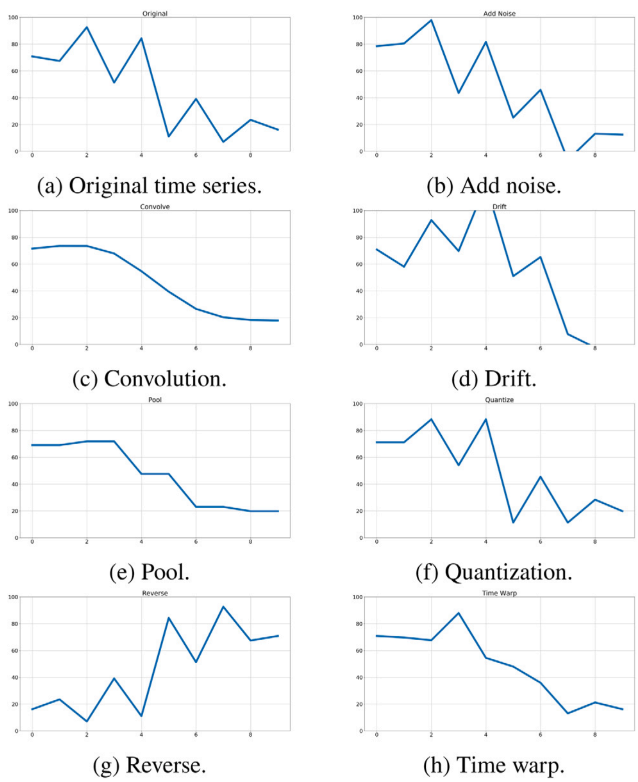
3.3. Injector data

Aside from the time series data from the producer well, we also have information from injector wells in the reservoir. The process of injecting water or gas takes time to influence the producer well. In other words, when injecting a fluid into the reservoir, it generates a pressure pulse, and this takes some time to reach the producer well (diffusivity time) (Johnson et al., 1966).

When considering injector well data, we have to examine the connectivity among them. One approach to determine interwell connectivity is the Time Lagged Cross-Correlation (TLCC), which uses Pearson's correlation coefficient applied on two time series shifted in time (Shen,

Table 1
Perturbations for the data augmentation in the training phase with their descriptions.

Perturbations	Description
Add noise	Adds random noise to the time series.
Convolution	Performs a convolution in the time series with a kernel window, i.e., a composition function between the time series' values and a kernel function (e.g., triangular, Hann window), which acts as a filter.
Drift	Adds a drift value to some points of the time series.
Pool	Divides the time series into windows and then applies a pooling function (e.g., maximum, minimum, average) to each window point.
Quantization	Defines level sets according to a distribution (e.g., uniform, quantile, k-means) and rounds the time series' values to the nearest level in the level set.
Reverse	Reverses the timeline of a series.
Time warp	The augmenter randomly changes the speed of the timeline.



(a) Original time series. (b) Add noise. (c) Convolution. (d) Drift. (e) Pool. (f) Quantization. (g) Reverse. (h) Time warp.

Fig. 8. Data augmentation in training phase. (a) shows the original time-series, and (b) to (h) shows the augmentation through perturbations.

2015). The TLCC helps identify lags of influence from an injector that might be useful to infer a producers' production (Menke and Menke, 2016).

Pearson's correlation coefficient defines the degree of linear correlation between two time series. The higher this coefficient, the more significant the correlation between them (Tian and Horne, 2016). Pearson's correlation coefficient is defined as

$$r = \frac{\sum_{i=1}^{n}\left(I_i - \overline{I}\right)\left(P_i - \overline{P}\right)}{\sqrt{\sum_{i=1}^{n}\left(I_i - \overline{I}\right)^2}\sqrt{\sum_{i=1}^{n}\left(P_i - \overline{P}\right)^2}}, \quad (3)$$

in which I_i and P_i are the injector and producer time series, respectively, n is the length of the series, and $\overline{I}$ and $\overline{P}$ are the mean value of the series I and P, respectively.

In this work, we considered the two correlations. The first is the correlation between the mass flow of water and gas of the injector with the producer's BHP derivative. The other correlation was made considering the BHP and reservoir pressure difference, both from the injector and producer.

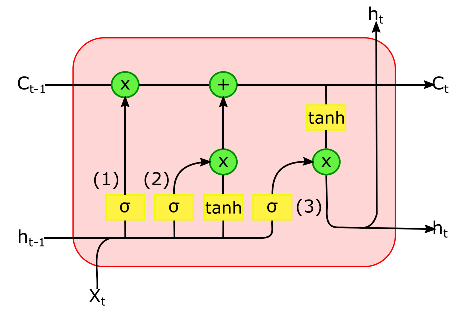


Fig. 9. Representation of an LSTM layer. X_t represents the input at instant t, C_{t-1} and C_t are the memory from the previous LSTM cell and current cell, and h_{t-1} and h_t are the output of the previous cell and the output of the actual cell. The gates of this cell are enumerated as follows: (1) is the forget gate, (2) is the input gate, and (3) is the output gate. In this recurrent cell, the input and the previous output decides to consider the memory from the last cell, which then update the memory state of the cell in the input gate, and finally, the memory of the cell combined with the inputs results in the output of the cell.

3.4. Data-driven techniques

Time series derived from petroleum field production data is highly non-linear; to perform forecast with these data, we need to rely on non-linear formulations, such as Artificial Neural Networks (ANN) (Li et al., 2013). The architecture's brain-inspired ANNs can help solve large and complex tasks, with applications both in academia and industry. The idea of ANNs is to combine a vast set of artificial neurons and connections with learned weights directly from the data aiming at solving a specific problem. The literature on data-driven techniques presents several neural network architectures designed for different tasks. For time series forecasting, some of the most suitable are RNN and Convolutional Neural Networks (CNN). However, it is unlikely that using an off-the-shelf solution would solve a complex problem as the one in this work.

The network we devise herein stacks multiple recurrent layers, creating a deep RNN, varying the input/output sequences among the layer (Géron, 2019). There are two main recurrent layers in the literature: LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit). LSTM is a recurrent layer that remembers previous steps using three gates that manipulate information. The first, called *forget gate*, removes irrelevant information from the previous steps. The second is the *input gate* which allows (or not) the input value to be accumulated and updates the cell memory. Finally, the *output gate* can shut off the output of the cell (Goodfellow et al., 2016). GRU differs from the LSTM by using a single gate to simultaneously control the forget function and the decision to update the actual state, named *update gate*, and the *reset gate*. Figs. 9 and 10 show these two recurrent layers and their gates.

We experimented with two different types of recurrent networks. In the first, Seq2Vector, we feed the network with a sequence of inputs and only the output vector from the last recurrent layer is considered. In the second one, Seq2Seq, we feed the network with a sequence of inputs and the network produces a sequence of outputs at every recurrent step. The advantage is that it will consider the output at every time step in the error calculation, which improves the training (Géron, 2019). We adopted the Seq2Vector in the last recurrent layer, followed by one or more dense layers to provide the multiple output prediction.

CNNs consist of a different formulation than RNNs. They are well-known and widely successful networks used mainly for image classification (Krizhevsky et al., 2012). CNNs consist of sequences of convolutional layers, which perform convolutions between local regions of the input and a defined filter, or weight matrix, which slides over

Table 2
The ANNs details we have used in the experiments. A plot of the models can be found in Fig. A.1.

GRU2	GRUconv	Seq2Seq	CNN
2x gru: units = 128, return sequences = T/F	2x conv1D: filters = 64/32, kernel size = 4, strides = 2, padding = "valid"	2x gru: units = 128, dropout = 0.1, recurrent dropout = 0.5, return sequences = T	3x conv1D: filters = 64, kernel size = 2, padding = "same" + batch normalization
10x dense: units = 128/64/32/30	2x gru: units = 128, dropout = 0.1, recurrent dropout = 0.5, return sequences = T/F	lambda: last 30 days	global average pooling1D
	10x dense: units = 128/64/32/30	time distributed: dense(1)	1x dense: units = 30

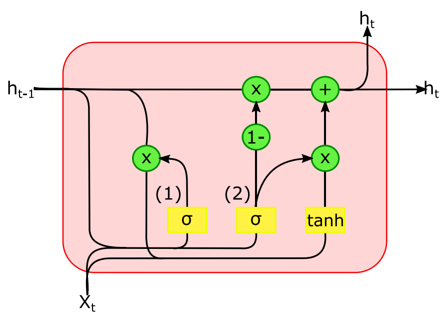


Fig. 10. Representation of a GRU layer. X_t represents the input at instant t and h_{t-1} and h_t are the output of the previous cell and the output of the actual cell. The gates of this cell are enumerated as follows: (1) is the reset gate, and (2) is the update gate. First, the data goes through the reset gate, which decides how much information from the past to forget, and then passes through the update gate that controls the information that flows into the memory.

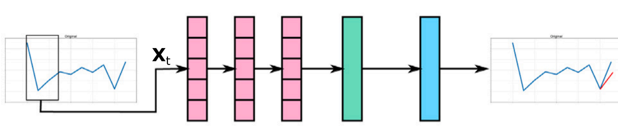


Fig. 11. Example of a forecasting network using CNNs.

the input. A CNN architecture learns the filters to be able to recognize patterns in the input data. Fig. 11 presents an example of a network using CNNs. In this network, a sliding window in the input goes through the convolutional layers to find the input patterns, next the pooling layer to obtain the most salient elements obtained in the convolution. Finally, a dense layer interprets the extracted features and returns the output.

Applying CNNs to time-series forecasting tasks consists of learning filters representing the patterns as a 1D feature map. With that, it uses these patterns to help forecast future values. One advantage of CNNs for forecasting is that they can access a broad range of history of the time series (van den Oord et al., 2016; Borovykh et al., 2017) and reliably capture local structures from the data. The created 1D feature map can also be mixed with recurrent layers, helping the recurrent layer detect more extended patterns (expanding a local view), especially when the convolutional layer reduces the sequence's size. Our experiments evaluated both setups for the forecasting, CNNs alone and CNNs allied with RNNs.

Table 2 summarizes the four network architectures (named GRU2, GRUconv, Seq2Seq, and CNN) adopted in this work. Note that we opted to use GRU layers instead of LSTM since they proved faster. For the sake of visualization, the number of layers and their respective type are defined on the cells' top. The table also describes the hyperparameter values, such as the number of neurons in each layer, dropout, filters, kernel size, stride, padding, and if it returns a sequence (T) or not (F). We selected these networks to represent the two approaches for forecasting (RNN and CNN) and a combination of them. Compared to forecasting oil production literature, these architectures are deeper and a little more complex, stacking up more recurrent layers and more dense layers. We intend to release the complete source code of our methods freely through GitHub upon acceptance of this paper.

4. Experimental protocol

This section describes the protocol guiding our experiments, including the selected datasets, the protocol for splitting the datasets between train and test sets, and the metrics applied to the results.

4.1. Datasets

For experiments and validation, we adopted two benchmarks from the literature that are not from the oil industry (Metro Interstate Traffic Volume and Appliances Energy Prediction) to consider the forecasting setup and a proprietary dataset from a pre-salt oil reservoir. For the sake of open science, we also present new data from a benchmark model for experiments in oil and pressure forecasting, which will be fully available upon acceptance of this paper, also with pre-salt conditions and characteristics. We adopted the first two datasets to show compatibility with previously evaluated methods in the literature and pinpoint that the methods we explore in this work might apply to other setups beyond the oil and gas industry.

The Metro Interstate Traffic Volume Dataset[1] is a multivariable time series benchmark created from hourly data of the Interstate 94 Westbound traffic volume for MN DoT ATR station 301 from 2012 to 2018. This station is located roughly midway between Minneapolis and St. Paul, MN, USA. The dataset also contains hourly weather features and information on holidays. It has 48,204 data points with 9 variables. We selected the numeric variables (`temp`, `rain_1h`, `snow_1h`, `clouds_all`, and `traffic_volume`) for our experiments, and defined the `traffic_volume` as the target.

The Appliances Energy Prediction Dataset (Candanedo et al., 2017) is a multivariate benchmark to perform regression of energy use in a low energy building. It comprises almost five months of information, measured every 10 min, resulting in 19,735 data points with 29 attributes.

Considering oil and pressure forecast, our primary focus, we worked with two datasets: one private and another generated from a benchmark model. The private dataset comprises production data from a Brazilian pre-salt oil reservoir. This dataset provides information on

[1] https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume, accessed on November 23rd, 2020.

fluid production (oil, gas, and water), pressure (bottom-hole), and the ratio between them (water cut, gas–oil ratio, and gas–liquid ratio). The reservoir contains 16 producers and 16 injector wells, divided into nine water injectors and seven WAG injectors. For the oldest producer well, we have five years of historical data.

The final dataset is the UNISIM-II-M-CO benchmark,[2] created by the UNISIM group at the University of Campinas. This synthetic benchmark, run on the CMG-GEM simulator,[3] has production and injection trends similar to the private dataset apart from being a carbonate reservoir based on real field data. The model is a synthetic light-oil based on a combination of Pre-Salt characteristics, such as fractures, Super-K layers, and high heterogeneity (Correia et al., 2015). The fluid model is compositional, with seven components in the oil phase. The simulation model has 6.5 years of production history and contains eight WAG injectors alternating every six months and ten producer wells. All the producers and injectors of the simulation model present total and partial closure frequency similar to real cases.

4.2. Protocol

In this subsection, we describe the adopted evaluation protocols. We selected the Metro and the Energy datasets for the experiments comparing the forecasting setups to show that our proposed setup can generalize to other forecasting problems.

We reserved 10% of each dataset for testing (4820 and 1973 data points, respectively), following the procedure done by Hu and Zheng (2020). We applied an RNN network (GRU2 network, described in Section 3.4) with Huber loss in all experiments as our proposed baseline, unless stated otherwise. As the approach is stochastic, we performed ten runs to obtain a margin of its results. We obtained the mean of these predictions as our final result. We fed batches of data containing 85 points as input to learn how to predict the following 30 data points (output) for training the network. All experiments were performed with 100 epochs, using an early stopping approach after 10 epochs without improving the validation loss, a common practice in machine learning.

For the oil datasets, we performed experiments considering our *N-th day* approach, as it is better for assessing the quality of the forecasting further in the future, as discussed in Section 3.1. As the datasets contain daily data, we performed a forecast of one month ahead (30 data points output), given an input of 85 data points.

Thirty days might seem a small time frame for petroleum engineers, as they are used to forecasting years using numerical simulations. However, it is a larger enough window to help decide interventions in the field's operation. A reservoir simulator is usually not predictive enough for short-term events, particularly for large and heterogeneous reservoirs, due to the complexity of representing such reservoirs and computational limitations. Machine-learning approaches can deal better with the complex data available from different sources and high-frequency data of the oil and gas industry. Therefore, through the more refined data, these machine-learning approaches can be more accurate to predict a near-future event, such as kicks, hydrate formation, or early water and gas breakthroughs.

All the networks tested in these datasets use Huber loss. We selected two targets for our experiments in these datasets: daily oil production and well bottom-hole pressure. Each data point input of the networks consists of all available variables for the given data point, e.g., production data, injection data, and well pressure.

After obtaining the forecast results, we performed post-processing in the data. We ensure that the target was not negative for the oil datasets, as our targets are daily pressure and daily production. We also removed any outliers above one standard deviation.

[2] https://www.unisim.cepetro.unicamp.br/benchmarks/en/unisim-ii/overview, accessed on November 23rd, 2020.

[3] https://www.cmgl.ca/, accessed on September 29th, 2020.

Table 3
Forecasting setups on Metro and Energy datasets.

Dataset	Metrics	Forecasting setups			
		Concatenation	Tumbling	Retraining + Tumbling	Nth Day
Metro	MAE	1062.65	1054.47	1046.67	1432.57
	RMSE	1405.41	1391.53	1383.81	1764.99
	SMAPE	43.49	43.30	43.03	54.21
Energy	MAE	43.29	43.25	43.21	45.78
	RMSE	91.56	91.33	88.56	93.79
	SMAPE	36.08	36.15	36.21	38.25

Upon obtaining the forecasting results, we evaluated the approaches through three metrics commonly used in forecasting problems (Kubota and Reinert, 2019; Oreshkin et al., 2019; Liu et al., 2020): Mean Absolute Error (MAE), that shows the magnitude of errors,

$$\mathrm{MAE}(X,h) = \frac{1}{m}\sum_{i=1}^{m} |h(x^i) - y^i|,$$

Root Mean Square Error (RMSE), which measures how spread out the errors are

$$\mathrm{RMSE}(X,h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(h(x^i) - y^i)^2},$$

and Symmetric Mean Absolute Percentage Error (SMAPE), which measures the percentage error of the predicted values,

$$\mathrm{SMAPE}(X,h) = \frac{100}{m}\sum_{i=1}^{m} \frac{|h(x^i) - y^i|}{\frac{(|y^i|+|h(x^i)|)}{2}},$$

where X are the predicted values and y the ground-truth.

5. Experiments and results

In this section, we present the sets of experiments and discuss their results. The first set of experiments (Section 5.1) compares different forecasting setups. The second set (Section 5.2) shows our forecasting method applied in the oil and gas field and highlights the importance of data pre-processing. The third set (Section 5.3) adds information about the injection data, considering the delay of influence extracted from correlation techniques. Finally, the last set (Section 5.4) compares the proposed approach with a number of off-the-shelf data-driven solutions.

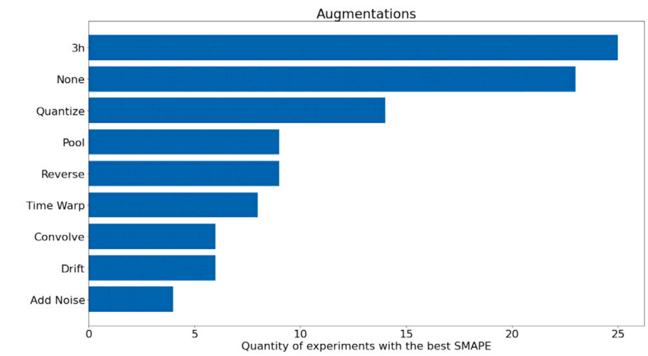
5.1. Comparing forecast setups

Our first experiment compared the different forecast setups in the literature (Concatenation, Tumbling, and Retraining) with our proposed approach (*N-th Day*). All these setups perform multiple output forecasting, they comtemplate a more extensive testing set, and combine their outputs differently. Table 3 shows how these methods perform in the Metro and Energy datasets.

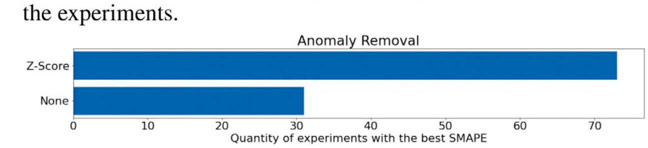
As Table 3 shows, the *N-th Day* evaluation setup does not have the best result, as expected. This is coherent with its proposition of considering only the last data of each horizon window and integrate this data at the end. The last day is the most challenging data point to be predicted, as it is the furthest from the input data.

All other setups consider the other predicted data points (e.g., 1-day prediction) in their evaluations and, because of this, their results tend to be higher. For instance, the first predicted data point is straightforward, as the method can even use the same last data point seen without any learning method and still obtain reasonable results. We can see in the Energy dataset that the results for the *N-th Day* setup are not far from the other approaches. The subsequent experiments present results using only the *N-th Day* setup for forecasting.

It is interesting to notice that the Retraining setup outperforms the Concatenation and Tumbling methods. However, as it trains every new prediction, its costs are multiple times the cost of the other methods.

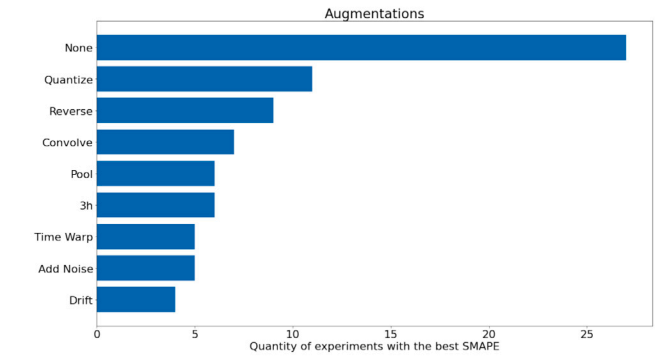


(a) The augmentation by data points using 3h was better in 25 of the experiments.

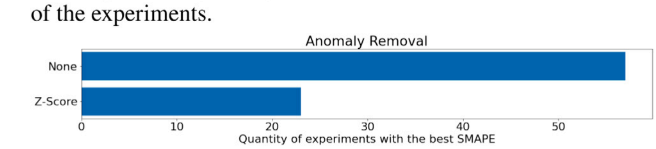


(b) The detection using z-score improved the obtained results in 73 of the experiments.

Fig. 12. Augmentation and anomaly detection experiments on the private dataset, considering 104 experiments (4 networks, 13 wells, and 2 target variables).



(a) Not performing an augmentation in this dataset was better in 27 of the experiments.



(b) Not performing an anomaly detection was better in 57 of the experiments.

Fig. 13. Augmentation and anomaly detection experiments on the UNISIM-II-M-CO dataset, considering 80 experiments (4 networks, 10 wells, and 2 target variables).

5.2. Comparing different pre-processing techniques

Our second experiment focuses on field response data, i.e., fluid production and wells pressure from a producing reservoir. We want to remove erroneous data that we cannot predict, such as human intervention.

We separated the anomaly removal and the data augmentation to see how they improved the forecasting results for these experiments. We used a z-score method for anomaly removal. Considering data augmentation, we created data points that correspond to intervals of 3 h in our original daily data, interpolating these data linearly.

The perturbations for data augmentation are presented in Table 1. Using our two targets, we performed all augmentations combined with our four networks on 13 producer wells of the private dataset (producer with enough data to perform the perturbations). We compared the results to select the augmentation approach that performed better in more experiments. Fig. 12(a) shows the number of experiments that have the best result with the corresponding augmentation. This figure shows that the augmentation by data points performed better in more wells than the other approaches. Moreover, Fig. 12(b) considers the same experiments and shows the influence of the use of an anomaly detector.

We also performed these experiments in the UNISIM-II-M-CO dataset, considering all its ten producer wells. Figs. 13(a) and 13(b) presents these results for augmentation and anomaly detection, respectively. It is clear that, for the UNISIM-II-M-CO dataset, the removal of anomalies worsens both results (augmentation and anomaly detection). We believe this is because the interference in this benchmark model was artificially generated, following a distribution. Thus the anomaly and synthetic data are intrinsic to the time series data so that any anomaly removal would disrupt the dataset model, and interpolation would only provide noise.

The next experiments in this work consider the best results on augmentation and anomaly removal for each dataset. For experiments with the private dataset, we used augmentation by 3 h and z-score anomaly removal. For the UNISIM-II-M-CO dataset, we do not perform an augmentation nor do we remove anomalies. Figs. 14 and 15 present plots for three wells and two targets, `DailyProdOil` (daily oil production) and `DailyPressureBHP` (daily measure of well bottom-hole pressure), respectively. We can see how the forecast (red) performs in these plots compared to the ground-truth (blue).

Table 4
SMAPE results for 3 wells on the private dataset using GRU2 method and considering (or not) their correlated injectors. Well P1 is connected to injectors I1 with 9 days delay and I2 with 5 days delay. Well P2 has 3 connected injectors: I2, I3, and I4, with 1 day, 15 days, and 1 day delays, respectively. Producer well P3 is correlated with 6 different injector wells: I5 without delay, I6 with 2 days delay, and wells I7, I8, I9, and I10 with 1 day delay each.

Well	Target	Without injector	With correlated injectors
P1	`DailyProdOil`	29.12	**28.73**
	`DailyPressureBHP`	0.99	**0.96**
P2	`DailyProdOil`	46.12	**38.94**
	`DailyPressureBHP`	**2.60**	2.69
P3	`DailyProdOil`	**7.01**	7.44
	`DailyPressureBHP`	**0.35**	1.00

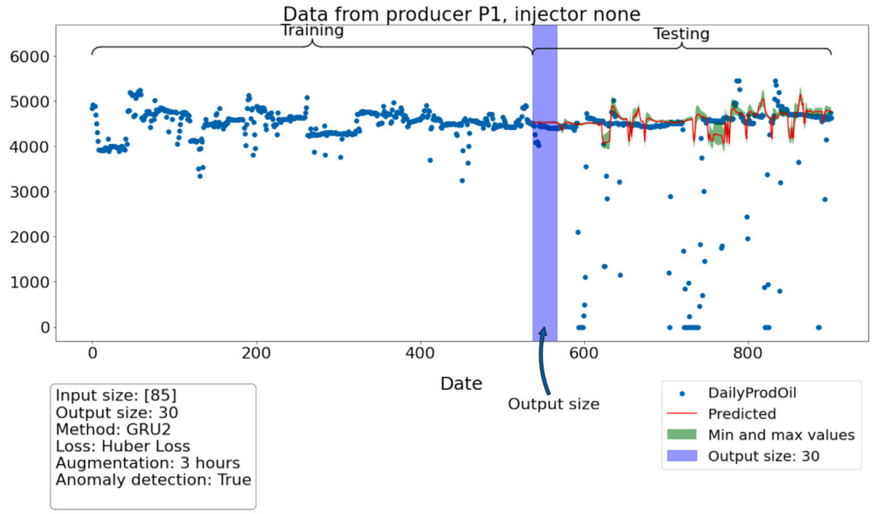
5.3. Injector data

Our following experiments considered injector data as input of our forecasting approaches. In these experiments, for each producer well, we devised the TLCC to determine which injector wells are connected to the producer and the lag between them. Tables 4 and 5 show the SMAPE metric for our two datasets, considering the oil production of a well with and without correlated injector wells.

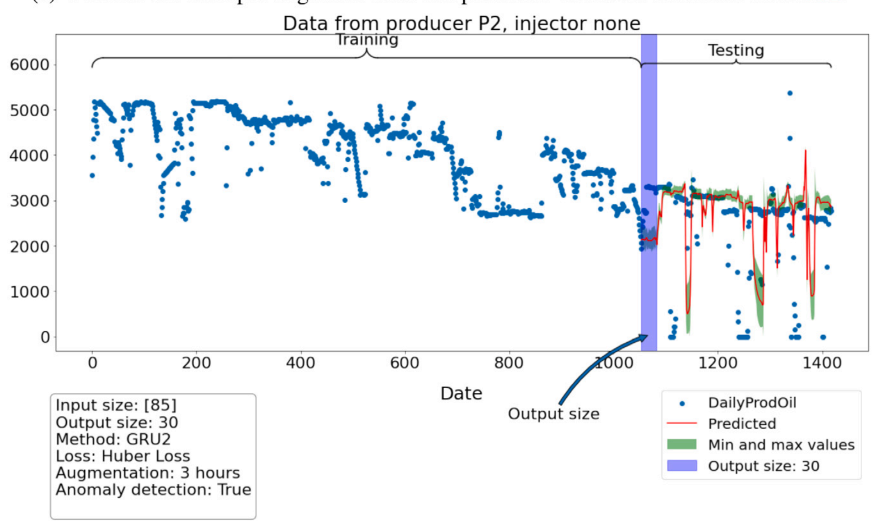
As shown in Tables 4 and 5, we do not have a definitive answer on whether it is better to use data from injector wells, especially considering the private dataset. Intuitively, we think it is better to include this information. We believe future investigations should be performed to enhance the correlation of producer and injector wells so that more precise information could be used as input to our algorithms.

5.4. Comparing data-driven techniques

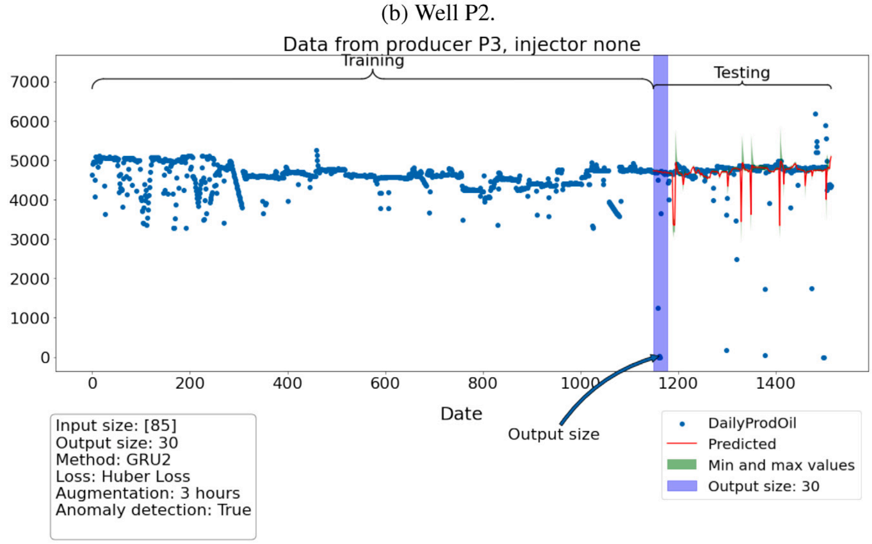
Our last round of experiments compares our methods with baselines and different off-the-shelf deep-learning networks from the literature,



(a) Well P1. An example of ground truth and predicted values are shown in Table A.2.

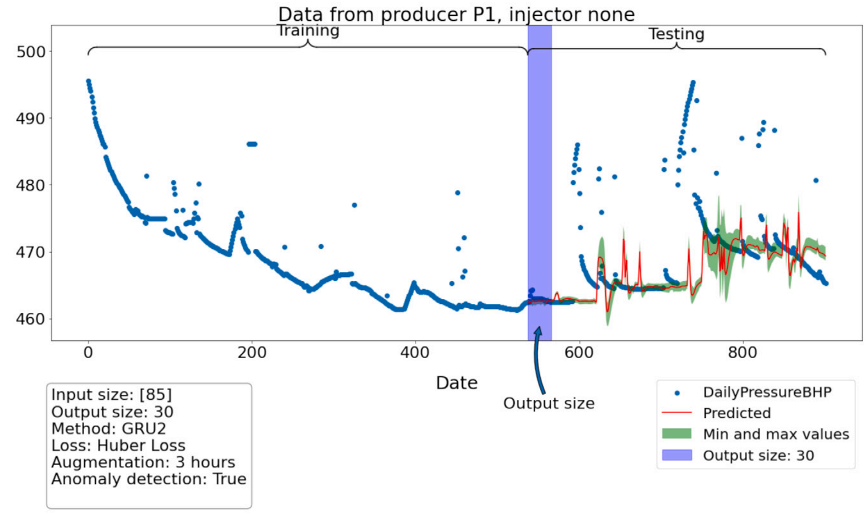


(b) Well P2.

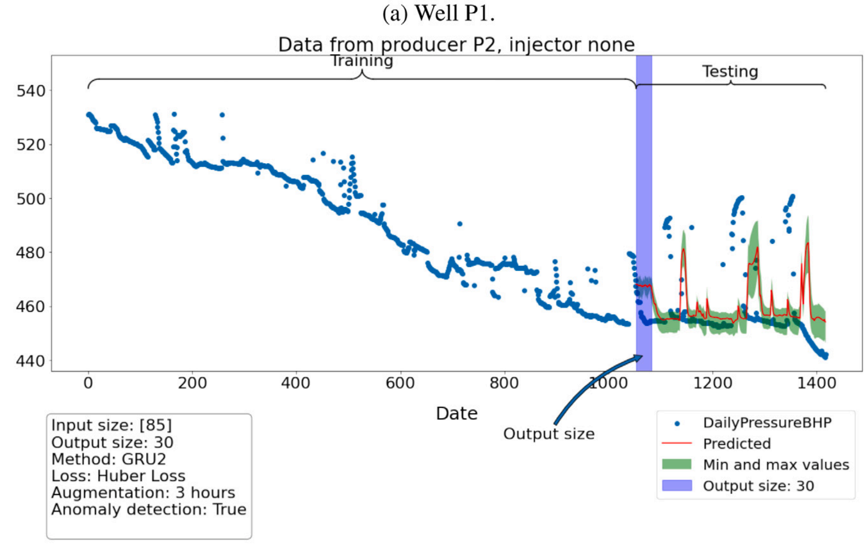


(c) Well P3.

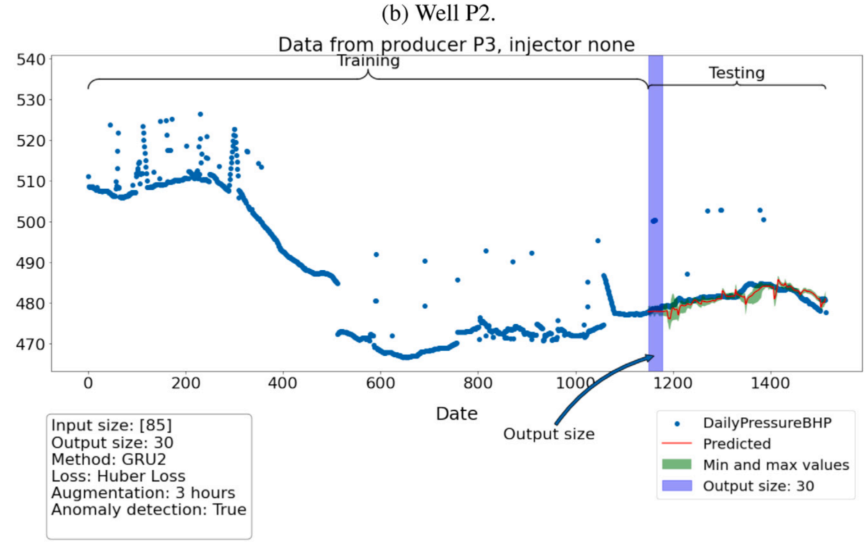
Fig. 14. `DailyProdOil` forecasting in the private dataset. The purple strip represents the start of the test set and has the size of the output window. The red line is the forecasted data, and the blue points are the ground-truth data. The green shadow is the maximum and minimum values obtained considering all 10 runs. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Well P1.



(b) Well P2.



(c) Well P3.

Fig. 15. `DailyPressureBHP` forecasting in the private dataset The purple strip represents the start of the test set and has the size of the output window. The red line is the forecasted data, and the blue points are the ground-truth data. The green shadow is the maximum and minimum values obtained considering all 10 runs. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
SMAPE results for 3 wells on the UNISIM-II-M-CO dataset using GRU2 method and considering (or not) their correlated injectors. Well PRK028 is connected to injectors IRK004 with 8 days delay and IRK049 with 5 days delay. Well PRK045 has just 1 connected injector: IRK049, with 4 days delay. Producer well PRK014 is correlated to the injector wells IRK004 and IRK049, both with 2 days delay.

Well	Target	Without injector	With correlated injectors
PRK028	DailyProdOil	45.74	**23.09**
	DailyPressureBHP	**6.86**	7.03
PRK045	DailyProdOil	15.70	**15.52**
	DailyPressureBHP	7.64	**6.81**
PRK014	DailyProdOil	106.12	**90.98**
	DailyPressureBHP	3.52	**2.99**

e.g., Transformer, N-BEATS, DeepAR, among others. For these new networks, we used a toolkit for time series modeling called GluonTS.[4]

We adapted these baselines and off-the-shelf networks to our *N-th Day* setup. In that way, we modified our test set to contain just the last 100 days of our datasets and maintained our batch output size of 30 data points. That way, we have 71 predictions, applying our setup on the 30th day. However, we could not apply this setup to the Prophet network, as their solution does not consider the test's input. We therefore included its results considering a retraining setup combined with our *N-th Day* setup.

For comparison, we selected baselines, off-the-shelf solutions for forecasting, a mechanism to improve recurrent networks, and a state-of-the-art forecasting approach. Details of these networks can be found in Vaswani et al. (2017), Taylor and Letham (2018), Oreshkin et al. (2019), Salinas et al. (2020). Figs. 16 and 17 show the SMAPE metric for three producer wells and two target variables.

As we can see in Figs. 16 and 17, no one network performs better than the rest. Thus, for oil production forecasting, it is not enough to select an off-the-shelf method and apply it to the problem. It is essential to recognize the problem, understand the available variables, and propose an approach to resolve it. One possible reason for PN-DCA to perform worse than machine-learning techniques in these experiments is their ability to adapt to various changes in the data. In contrast, PN-DCA depends on some conditions of the reservoir that are not true in our cases (Arps, 1945; Lacayo and Lee, 2014), such as: not being influenced by injection support, long production history, and performing under pseudo-steady-state conditions or boundary-dominated flow conditions. As we consider the normalized pressure rate in the PN-DCA's forecasting, we had to define the bottom-hole pressure for the testing data. For that, we used linear regression to obtain these values, as we lack access to the ground-truth data of the testing data.

Each reservoir is unique and specific forecasting methods are needed for the Oil and Gas industry. In particular, the forecasting window is much larger, the data suffers high interference from unknown correlations, and there are many anomalous data (such as human interference). Our applied approaches consider these variables, but there is still room for improvement.

6. Conclusion and future work

We focused this work on short-time 30-day forecasting of fluid rates and bottom-hole pressure for hydrocarbon reservoirs using data-driven procedures. We used different datasets to evaluate various forecasting setups on broader horizons, applied a number of pre-processing techniques, and studied their impact on forecasting, including information on injector data. We also evaluated several off-the-shelf approaches to this problem.

The proposed *N-th Day* approach highlights the importance of having an accurate method to forecast multiple outputs. Compared to the literature, we identified that forecasting multiple outputs is challenging due to the sliding window overlapping values already predicted. We identified that several approaches continuously update the forecasting values, leading to a wrong interpretation of the multiple outputs comparable to a single output, which is crucial for managing oil fields. Thus, we aim for this proposed method to be the standard for multiple output forecasting scenarios. The differences between the proposed method and the standard methods found in the literature are detailed throughout several experiments.

Considering the data-driven oil production forecasting scenario, we evaluated the impact of different pre-processing techniques, such as augmentation and removal of anomalies. The experiments were conducted using both real data and synthetic data from the UNISIM-II-M-CO benchmark. Unlike common forecasting examples, such as weather prediction, positive influence by pre-processing was dataset dependent. We also studied the influence of injector data in forecasting oil production, from which we could not determine if it is better to use injector data or not, so additional studies on this problem are required.

We performed experiments comparing different off-the-shelf data-driven techniques with network configurations to demonstrate the complexity of the oil production forecasting context. We verified that selecting an off-the-shelf method to apply to the oil production forecast with these experiments is not enough. We also confirmed that forecasting multiple outputs is tricky, which may lead to erroneous evaluation of the designed model if a specific approach, such as the proposed *N-th Day*, is not considered. Further, it is essential to understand the data and the problem at hand.

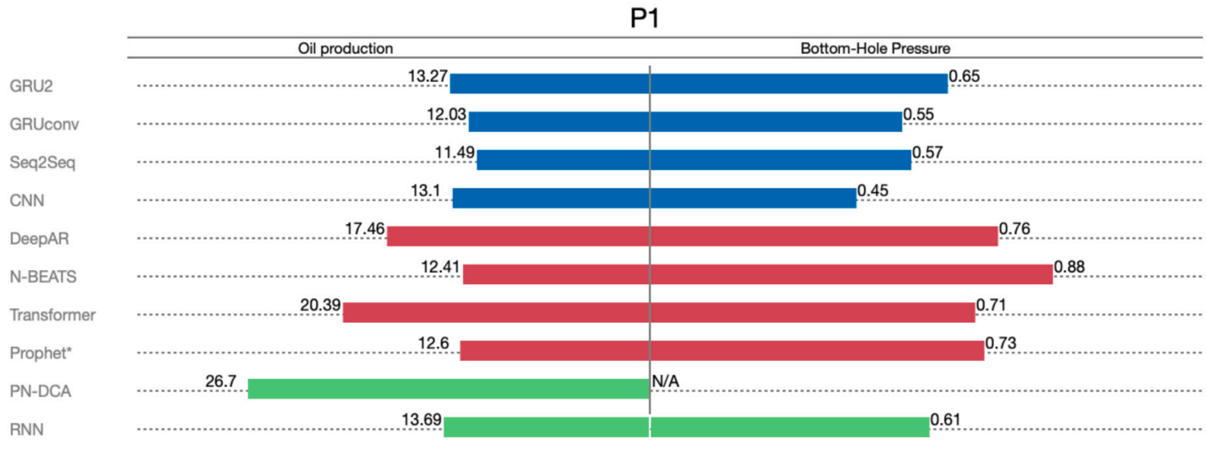
Based on the obtained results, the best ones were achieved using stacked recurrent network layers to consider long time-frames as input, such as LSTM and GRU. Harnessing these networks is already an improvement compared to the linear regression and single-layer recurrent networks commonly found in the oil production forecasting literature. Experiments with several recurrent networks indicate we do not simply need a more extended time frame as input. Instead, we need to design specific temporal representations highlighting crucial input snippets from different time series to support oil and pressure forecasting. We see it as a natural evolution of our proposed Seq2Seq and GRU2 architectures.

Future work

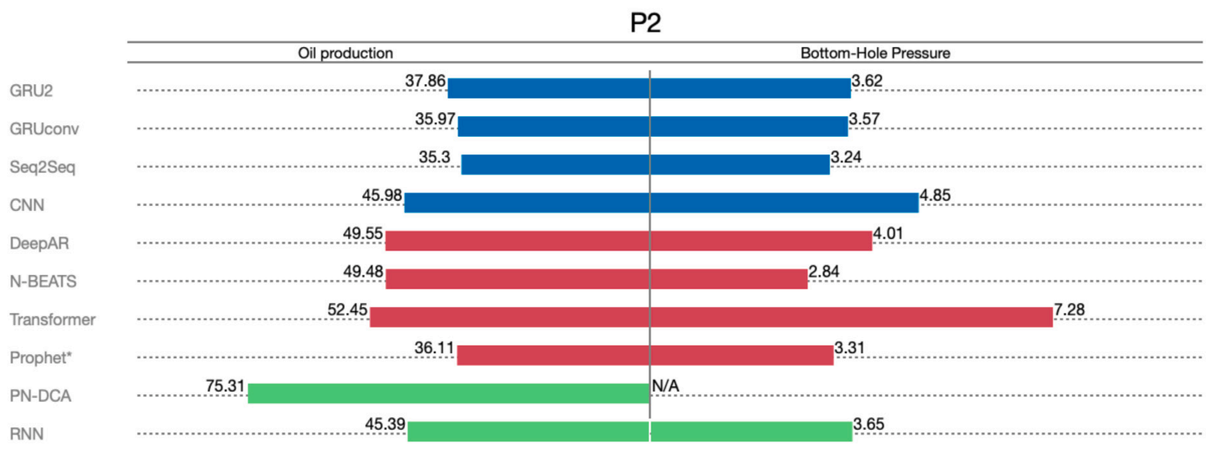
Our proposed approach for oil production forecasting can be improved considering, for example, attention mechanisms to quantify the interdependence between the input and output or within input elements, and a new hyperparameter considering the input size of the data, in which we can apply an auto-correlation method to obtain this parameter. Regarding the model hyperparameters, we have used the default values from Keras, which have proved to be a good option by the literature. For instance, a hyperbolic tangent is used as the activation function for all recurrent layers, which is the default, and it is important to reduce the issues with unstable gradients. Searching into the hyperparameters space can be pretty costly, as the deep learning model has many parameters (batch size, learning rate, loss function, number of epochs, early stopping, to name a few), so we plan to adopt a Genetic Algorithm approach.

Another future work is to consider the information of other producer wells in the training step of the target well. With more data, we can better understand the different well production evolution and production behaviors. It is crucial to have additional studies on the correlation between producer and injector wells to consider the interactions between them and improve the input data. In future work, we can include more details about the correlation between two wells, the delay for one well to influence the other, and the strength of the correlation between two wells. Another investigation proposal is to program the neural networks to learn this delay from the data in such a way that current data from

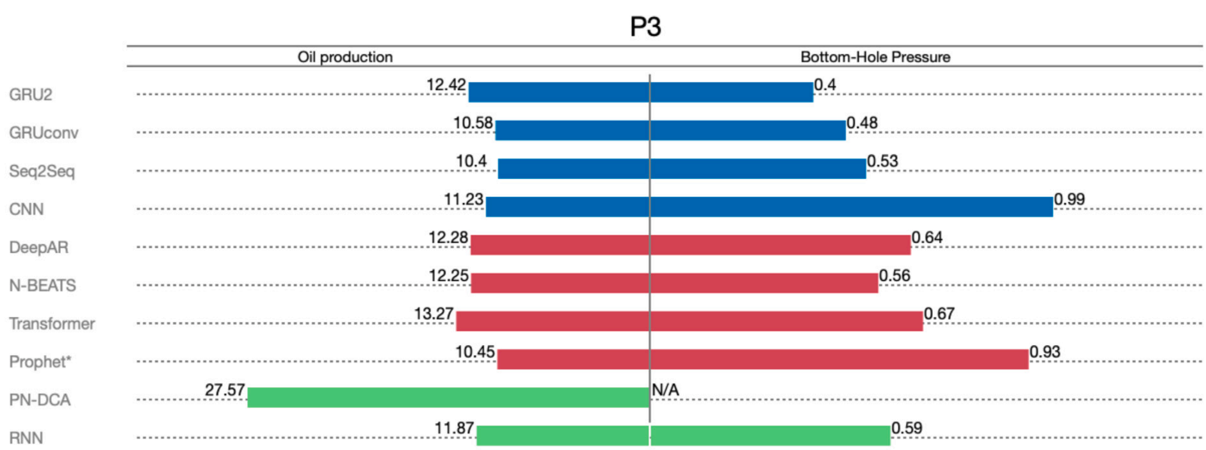
[4] https://ts.gluon.ai, accessed on October 27th, 2020.



(a) Well P1.



(b) Well P2.

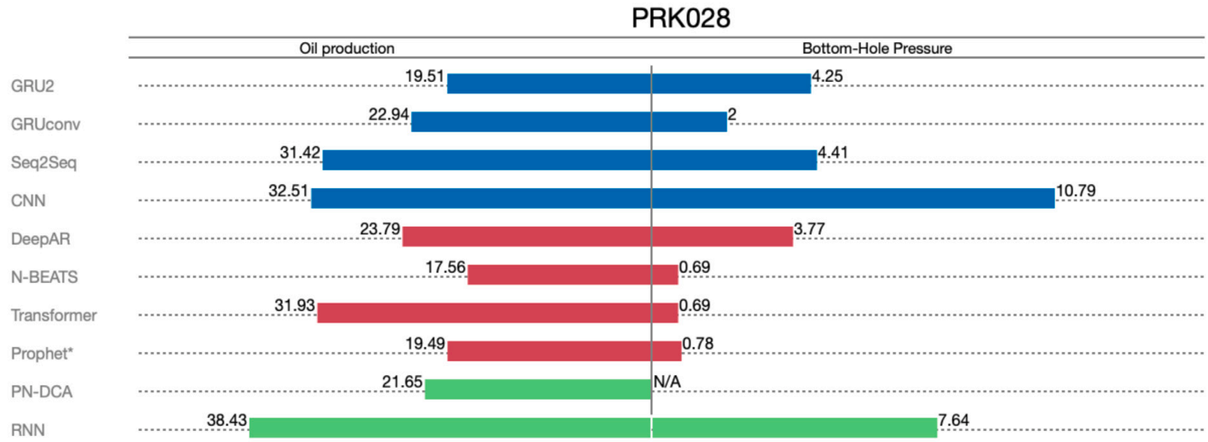


(c) Well P3.

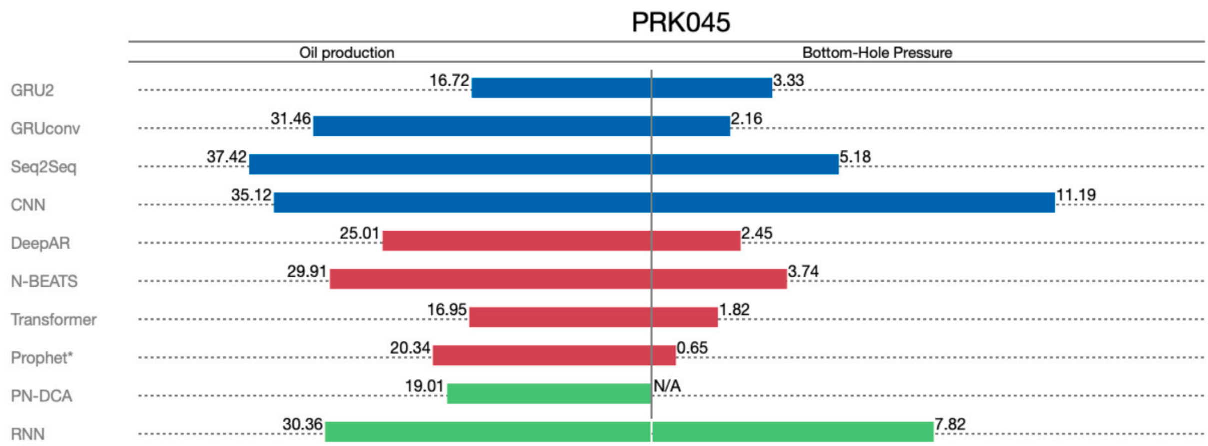
Fig. 16. SMAPE results for the private dataset considering oil production and bottom-hole pressure. The blue bars are the networks proposed for use in this work, the red bars are the off-the-shelf methods from the literature, and the green bars are the baselines methods. All experiments were performed with augmentation and anomaly removal. Prophet* means that this network uses a retraining approach. Note that the PN-DCA method is not applicable (N/A) to perform BHP forecasting. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

both injector and producer are fed to the network, disregarding the actual delay.

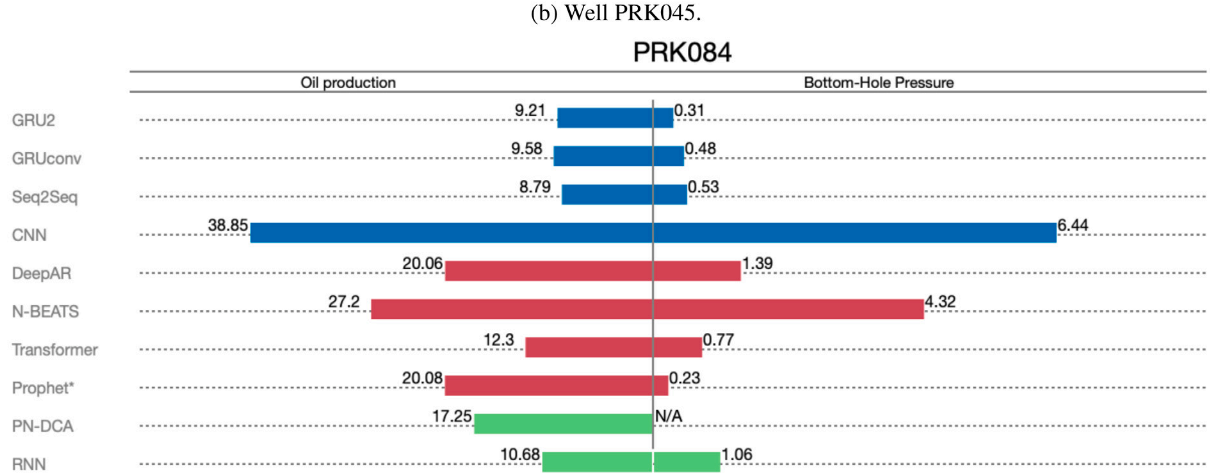
Thirty days of production forecast can be used to assist short-term decisions, such as kicks or early water and gas breakthrough.



(a) Well PRK028.



(b) Well PRK045.



(c) Well PRK084.

Fig. 17. SMAPE results for the UNISIM-II-M-CO dataset considering oil production and bottom-hole pressure. The blue bars are the networks proposed for use in this work, the red bars are the off-the-shelf methods from the literature, and the green bars are the baselines methods. These experiments considered the best pre-processing for the UNISIM-II-M-CO dataset (no augmentation and no anomaly removal). Prophet* means that this network uses a retraining approach. Note that the PN-DCA method is not applicable (N/A) to perform BHP forecasting. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

As mentioned before, this time window is much smaller than those usually considered in model-based approaches, whose forecasts cover more than ten years. We believe that data-driven and model-based approaches are complementary. An even better solution may be found by joining the two into a hybrid procedure, which is a topic for future research.

CRediT authorship contribution statement

Rafael de Oliveira Werneck: Conceptualization, Methodology, Software, Investigation, Writing – original draft. **Raphael Prates:** Software, Investigation, Writing – review & editing. **Renato Moura:** Software, Investigation, Writing – review & editing. **Maiara Moreira Gonçalves:** Data curation, Writing – review & editing. **Manuel Castro:** Data curation, Writing – review & editing. **Aurea Soriano-Vargas:** Visualization, Writing – review & editing. **Pedro Ribeiro Mendes Júnior:** Formal analysis, Writing – review & editing. **M. Manzur Hossain:** Resources, Writing – review & editing. **Marcelo Ferreira Zampieri:** Resources, Data curation, Writing – review & editing. **Alexandre Ferreira:** Conceptualization, Supervision, Writing – review & editing. **Alessandra Davólio:** Supervision, Writing – review & editing. **Denis Schiozer:** Supervision, Writing – review & editing. **Anderson Rocha:** Conceptualization, Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.petrol.2021.109937.

Acknowledgments

This work was conducted in association with the ongoing Project registered under ANP number 21373-6 as "Desenvolvimento de Técnicas de Aprendizado de Máquina para Análise de Dados Complexos de Produção de um Campo do Pre-Sal" (UNICAMP/Shell Brazil/ANP) funded by Shell Brazil, under the ANP R&D levy as "Compromisso de Investimentos com Pesquisa e Desenvolvimento". The authors also thank Schlumberger and CMG for software licenses and Vitor Ferreira for helping with the PN-DCA method.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.petrol.2021.109937.

References

Aizenberg, I., Sheremetov, L., Villa-Vargas, L., noz, J.M.-M., 2016. Multilayer neural network with multi-valued neurons in time series forecasting of oil production. Neurocomputing 175, 980–989. http://dx.doi.org/10.1016/j.neucom.2015.06.092.

Al-Shabandar, R., Jaddoa, A., Liatsis, P., Hussain, A.J., 2021. A deep gated recurrent neural network for petroleum production forecasting. Mach. Learn. Appl. 3, 100013. http://dx.doi.org/10.1016/j.mlwa.2020.100013.

Amirian, E., Fedutenko, E., Yang, C., Chen, Z., Nghiem, L., 2018. Artificial neural network modeling and forecasting of oil reservoir performance. In: Applications of Data Management and Analysis : Case Studies in Social Networks and beyond. Springer International Publishing, Cham, pp. 43–67. http://dx.doi.org/10.1007/978-3-319-95810-1_5.

Arps, J., 1945. Analysis of decline curves. Trans. AIME 160 (01), 228–247. http://dx.doi.org/10.2118/945228-G.

Athanasopoulos, G., Hyndman, R.J., Song, H., Wu, D.C., 2011. The tourism forecasting competition. Int. J. Forecast. 27 (3), 822–844. http://dx.doi.org/10.1016/j.ijforecast.2010.04.009.

Bedi, J., Toshniwal, D., 2019. Deep learning framework to forecast electricity demand. Appl. Energy 238, 1312–1326. http://dx.doi.org/10.1016/j.apenergy.2019.01.113.

Belyadi, H., Fathi, E., Belyadi, F., 2019. Chapter seventeen - decline curve analysis. In: Belyadi, H., Fathi, E., Belyadi, F. (Eds.), Hydraulic Fracturing in Unconventional Reservoirs (Second Edition), second ed. Gulf Professional Publishing, pp. 311–340. http://dx.doi.org/10.1016/B978-0-12-817665-8.00017-5.

Bianchi, F.M., Scardapane, S., Løkse, S., Jenssen, R., 2021. Reservoir computing approaches for representation and classification of multivariate time series. IEEE Trans. Neural Netw. Learn. Syst. 32 (5), 2169–2179. http://dx.doi.org/10.1109/TNNLS.2020.3001377.

Bontempi, G., 2008. Long term time series prediction with multi-input multi-output local learning. In: 2nd European Symposium on Time Series Prediction, pp. 145–154.

Borovykh, A., Bohte, S., Oosterlee, C.W., 2017. Conditional time series forecasting with convolutional neural networks. arXiv:1703.04691.

Brownlee, J., 2018. How to develop multi-step LSTM time series forecasting models for power usage. https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/, (accessed: Aug 3rd, 2020).

Candanedo, L.M., Feldheim, V., Deramaix, D., 2017. Data driven prediction models of energy use of appliances in a low-energy house. Energy Build. 140, 81–97. http://dx.doi.org/10.1016/j.enbuild.2017.01.083.

Cao, Q., Banerjee, R., Gupta, S., Li, J., Zhou, W., Jeyachandra, B., 2016. Data driven production forecasting using machine learning. In: SPE Argentina Exploration and Production of Unconventional Resources Symposium. pp. 01–10. http://dx.doi.org/10.2118/180984-MS.

Chaikine, I.A., Gates, I.D., 2021. A machine learning model for predicting multi-stage horizontal well production. J. Pet. Sci. Eng. 198, 108133. http://dx.doi.org/10.1016/j.petrol.2020.108133.

Correia, M., Hohendorff, J., Gaspar, A.T., Schiozer, D., 2015. UNISIM-II-D: Benchmark case proposal based on a carbonate reservoir. In: SPE Latin America and Caribbean Petroleum Engineering Conference, Vol. Day 3 Fri, November 20, 2015. pp. 1–21. http://dx.doi.org/10.2118/177140-MS, D031S020R004.

Davtyan, A., Rodin, A., Muchnik, I., Romashkin, A., 2020. Oil production forecast models based on sliding window regression. J. Pet. Sci. Eng. 195, 107916. http://dx.doi.org/10.1016/j.petrol.2020.107916.

Deng, L., Pan, Y., 2020. Machine-learning-assisted closed-loop reservoir management using echo state network for mature fields under waterflood. SPE Reservoir Evaluation & Engineering 23 (04), 1298–1313. http://dx.doi.org/10.2118/200862-PA.

van Dyk, D.A., Meng, X.L., 2001. The art of data augmentation. J. Comput. Graph. Statist. 10 (1), 1–50. http://dx.doi.org/10.1198/10618600152418584.

Ertekin, T., Sun, Q., 2019. Artificial intelligence applications in reservoir engineering: A status check. Energies 12 (15), 2897–2919. http://dx.doi.org/10.3390/en12152897.

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

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, http://www.deeplearningbook.org.

Hu, J., Zheng, W., 2020. A deep learning model to effectively capture mutation information in multivariate time series prediction. Knowl.-Based Syst. 203, 106139. http://dx.doi.org/10.1016/j.knosys.2020.106139.

Johnson, C.R., Greenkorn, R.A., Woods, E.G., 1966. Puk-testing: A new method for describing reservoir flow properties between wells. J. Pet. Technol. 18 (12), 599–604.

Kim, Y.D., Durlofsky, L.J., 2021. A recurrent neural network–based proxy model for well-control optimization with nonlinear output constraints. SPE J. 26 (04), 1837–1857. http://dx.doi.org/10.2118/203980-PA.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems, Vol. 25. Curran Associates, Inc., pp. 1097–1105, URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Kubota, L., Reinert, D., 2019. Machine learning forecasts oil rate in mature onshore field jointly driven by water and steam injection. In: SPE Annual Technical Conference and Exhibition. Day 2 Tue, October 01, 2019, pp. 1–18. http://dx.doi.org/10.2118/196152-MS.

Lacayo, J., Lee, J., 2014. Pressure normalization of production rates improves forecasting results. In: SPE Unconventional Resources Conference / Gas Technology Symposium, Vol. Day 1 Tue, April 01, 2014. http://dx.doi.org/10.2118/168974-MS.

Li, X., Chan, C., Nguyen, H., 2013. Application of the neural decision tree approach for prediction of petroleum production. J. Pet. Sci. Eng. 104, 11–16. http://dx.doi.org/10.1016/j.petrol.2013.03.018.

Liu, W., Liu, W.D., Gu, J., 2020. Forecasting oil production using ensemble empirical model decomposition based long short-term memory neural network. J. Pet. Sci. Eng. 189, 107013. http://dx.doi.org/10.1016/j.petrol.2020.107013.

Makridakis, S., Hibon, M., 2000. The M3-competition: results, conclusions and implications. Int. J. Forecast. 16 (4), 451–476. http://dx.doi.org/10.1016/S0169-2070(00)00057-1.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2018. The M4 competition: Results, findings, conclusion and way forward. Int. J. Forecast. 34 (4), 802–808. http://dx.doi.org/10.1016/j.ijforecast.2018.06.001.

Menke, W., Menke, J., 2016. Environmental Data Analysis with Matlab. Academic Press.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., 2016. Wavenet: A generative model for raw audio. arXiv:1609.03499.

Oreshkin, B.N., Carpov, D., Chapados, N., Bengio, Y., 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint arXiv:1905.10437.

Pan, Y., Bi, R., Zhou, P., Deng, L., Lee, J., 2019. An effective physics-based deep learning model for enhancing production surveillance and analysis in unconventional reservoirs. In: SPE/AAPG/SEG Unconventional Resources Technology Conference. OnePetro, http://dx.doi.org/10.15530/urtec-2019-145.

Pao, H.T., 2007. Forecasting electricity market pricing using artificial neural networks. Energy Convers. Manage. 48 (3), 907–912. http://dx.doi.org/10.1016/j.enconman.2006.08.016.

Razak, S.M., Cornelio, J., Cho, Y., Liu, H.-H., Vaidya, R., Jafarpour, B., 2021. Transfer learning with recurrent neural networks for long-term production forecasting in unconventional reservoirs. In: SPE/AAPG/SEG Unconventional Resources Technology Conference. OnePetro, http://dx.doi.org/10.15530/urtec-2021-5687.

Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T., 2020. Deepar: Probabilistic forecasting with autoregressive recurrent networks. Int. J. Forecast. 36 (3), 1181–1191.

Shen, C., 2015. Analysis of detrended time-lagged cross-correlation between two nonstationary time series. Phys. Lett. A 379 (7), 680–687.

Song, X., Liu, Y., Xue, L., Wang, J., Zhang, J., Wang, J., Jiang, L., Cheng, Z., 2020. Time-series well performance prediction based on long short-term memory (LSTM) neural network model. J. Pet. Sci. Eng. 186, 106682. http://dx.doi.org/10.1016/j.petrol.2019.106682.

Sun, J., Ma, X., Kazi, M., 2018. Comparison of decline curve analysis DCA with recursive neural networks RNN for production forecast of multiple wells. In: SPE Western Regional Meeting. Day 4 Wed, April 25, 2018, http://dx.doi.org/10.2118/190104-MS.

Tadjer, A., Hong, A., Bratvold, R.B., 2021. Machine learning based decline curve analysis for short-term oil production forecast. Energy Explor. Exploit. 01445987211011784. http://dx.doi.org/10.1177/01445987211011784.

Taylor, S.J., Letham, B., 2018. Forecasting at scale. Amer. Statist. 72 (1), 37–45.

Tian, C., Horne, R.N., 2016. Inferring interwell connectivity using production data. In: SPE Annual Technical Conference and Exhibition.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, u., Polosukhin, I., 2017. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. In: NIPS'17, Curran Associates Inc., Red Hook, NY, USA, pp. 6000–6010.

Yuan, Z., Huang, H., Jiang, Y., Li, J., 2021. Hybrid deep neural networks for reservoir production prediction. J. Pet. Sci. Eng. 197, 108111. http://dx.doi.org/10.1016/j.petrol.2020.108111.

Zhan, C., Sankaran, S., LeMoine, V., Graybill, J., Mey, D.O.S., 2019. Application of machine learning for production forecasting for unconventional resources. In: Unconventional Resources Technology Conference, Denver, Colorado, 22–24 July 2019. pp. 1945–1954. http://dx.doi.org/10.15530/urtec-2019-47.

Zhong, Z., Sun, A.Y., Wang, Y., Ren, B., 2020. Predicting field production rates for waterflooding using a machine learning-based proxy model. J. Pet. Sci. Eng. 194, 107574. http://dx.doi.org/10.1016/j.petrol.2020.107574.