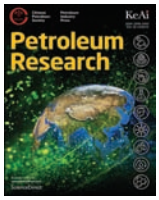
Petroleum Research 11 (2026) 743–760

Contents lists available at ScienceDirect

Petroleum Research

journal homepage: http://www.keaipublishing.com/en/journals/petroleum-research/



Full Length Article

Full-field petroleum production forecasting: Is deep learning all you need?



Leonardo Kubota *, Rafael dos Santos Gioria

Universidade de São Paulo Escola Politécnica, Brazil

A R T I C L E I N F O

Article history:
Received 25 May 2025
Received in revised form
9 January 2026
Accepted 14 January 2026
Available online 21 January 2026

Keywords:
Machine learning
Oilfield management
Data-driven proxy model
Oil production forecast
Green AI
Volve field

A B S T R A C T

We investigate full-field petroleum production forecasting using machine learning. A data-driven proxy model that predicts field-scale production under various operational scenarios is proposed. It offers an alternative to time-consuming numerical simulators for assisting operational oilfield management decisions like designing waterflooding schedules and quantifying production loss caused by schedule delays. Its accuracy is validated with real-world datasets. Beyond its practical application, we leverage the proposed workflow to conduct a systematic comparison of several machine learning and deep learning algorithms using four datasets and three forecast horizons. Such critical and systematic analysis is long overdue and seems to be extremely rare in related works, which consistently favor increasingly intricate neural networks, despite the presence of relatively limited and sparse datasets, low-dimensional input variables and the existence of well-defined physical laws to guide input selection. Our comparative study suggests deep learning is an overkill in the context of full-field production forecasts, without any tangible benefits in terms of accuracy. Kernel-based models consistently outperform neural-based architectures in this multi-dimensional analysis. In addition, we find evidence that algorithmic generalization in the full-field context seems to be more influenced by the spatial density of production data than by dataset size. The data-driven proxy model and the investigative findings of this work can help engineers apply machine learning to petroleum production forecasting. On a broader scale, our results endorse the Green AI paradigm, which advocates for the use of compact architectures that deliver good accuracy with a low computational cost.

© 2026 Chinese Petroleum Society. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Production forecasting plays a central role in oilfield management because it affects cash-flow evaluation. The field operator's main source of revenue is the produced oil, hence the need for accurate forecasts. Long-term production forecasts (several years) are important for strategic planning, such as making long-term investment decisions. Short and medium-term forecasts (several months to a few years) can assist field managers in operational decisions, such as resource allocation. The application of machine learning algorithms to forecast oil production has become an established practice in the petroleum industry (Bahaloo et al., 2023; Sircar et al., 2021; Wang and Chen, 2023). Machine-learning algorithms are computer programs capable of extracting, out of a dataset, a function mapping inputs to outputs (Kelleher, 2019). These models are more flexible and assumption-free than analytical methods, such as Decline Curve Analysis (Arps, 1945) and Capacitance Resistance Models (Holanda et al., 2018), and do not require the use of computationally expensive reservoir simulators (Cao et al., 2022; Hunyinbo et al., 2022). Atadeger et al. (2023) highlights the benefits of machine-learning proxy models (simplified tools that simulate the behavior of more complex systems for a range of input variables, Zubarev, 2009) over high-fidelity numerical simulators in well-level production optimization.

Authors concerned with production forecasts over monthly horizons have focused on the problem of one-month ahead forecasts (Kumar et al., 2023; Zainab and Horne, 2023). This is not ideal for operational and financial planning, which requires multi-step forecasting. We propose a workflow for building a full-field data-

* Corresponding author. Av Prof. Mello Moraes 2231, 05508-030, Sao Paulo, Brazil.
E-mail address: LKUBOTA@usp.br (L. Kubota).
Peer review under the responsibility of Editorial Office of Petroleum Research.

https://doi.org/10.1016/j.ptlrs.2026.01.007
2096-2495/© 2026 Chinese Petroleum Society. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

driven proxy model for multi-step forecasting. A machine learning algorithm is trained on low-dimensional historical production data to learn patterns between operational inputs and the full-field oil production. Since the proxy model is not restricted to identifying and extrapolating the flowrate trend based on a univariate paradigm, our multivariate model can be used as a prescriptive tool to guide operational business decisions in the context of oilfield management, such as assessing the impact of different waterflooding schedules on full-field oil production over short and medium-term horizons. The proposed workflow also implements a novel setup for multi-step forecasting leveraging generative data augmentation, which is shown to alleviate the issue of error propagation in long forecast horizons.

Based on this multivariate setup, a critical and systematic comparison between traditional machine learning (such as kernel-based models) and deep learning algorithms in the context of production forecasting is carried out. This analysis is long overdue in the literature. Most works are heavily biased towards applying increasingly intricate neural networks: to name a few, Sagheer and Kotb (2019), Wang et al. (2020), Werneck et al. (2022), Alali and Horne (2023) and Cruz and Pyrcz (2024) all consider some variation of LSTM or Transformers architectures, despite the presence of relatively limited and sparse datasets, low-dimensional input variables and the existence of well-defined physical laws to guide input selection. These characteristics raise questions around the actual need for automated feature learning present in neural-based architectures. On the other hand, evidence in favor of kernel-based models for time-series problems with small datasets of low dimensionality is well-established (Sapankevych and Sankar, 2009; Wolberg, 2000). More recently, the Green AI paradigm (Schwartz et al., 2020; Xu et al., 2021) has been gaining momentum as it advocates against the use of large models (with equally large computational costs and carbon footprint) when comparable results can be obtained from more compact architectures.

We summarize the contributions of this work as follows:

- A comparative study of deep learning and traditional machine learning algorithms in the context of full-field production forecasting using data from four oil fields and three forecast horizons suggests neural-based architectures are an overkill: i) they offer no benefits in terms of generalization, and ii) they are consistently outperformed by kernel-based models. In addition:
  - o The degree by which kernel-based models outperform deep learning models seems to follow an *inverse relationship with the spatial sampling density of production data.*
  - o *High spatial density of production data*, proxied by the number of wells in an oil field, improves generalization in the full-field context. The generalization error is low in fields with numerous wells, even when simple models are considered. Conversely, the generalization error deteriorates in fields with few wells, regardless of the architectural choice.
  - o *Dataset spatial density* seems to be more influential in explaining variations in model generalization than *dataset size or model architectural choice.*
- A novel workflow for building a *full-field data-driven proxy model* that simulates oil production under different scenarios of water injection and well reopening. This data-driven proxy model can be used as a decision support system for the operational management of waterflooding. It can serve as an alternative to time-consuming numerical simulators.
  - o The workflow implements a novel setup for multi-step forecasting leveraging *generative data augmentation,* which seems to alleviate the issue of error propagation present in long forecast horizons.

A limitation of the proposed workflow is that the oilfield development plan present in the training data must be consistent with the development plan adopted in the forecasted scenarios. For example, if the historical training data reflects oil recovery via water injection, the future scenarios cannot contain polymer or chemical injections. In this case, we need to collect historical data featuring the application of those operational regimes in the oilfield under study or, at the very least, data from an analogous oil field. Simply put, we cannot repurpose a trained model across different oil recovery strategies.

The main application of the proposed data-driven proxy model is in the context of short and medium-term scenario analysis as a prescriptive tool to help guide the best course of action for some operational input variables (to be defined in Section 4.4) of an oilfield under waterflood. Therefore, the proxy model predictions must be conditioned on the effective realization of a given scenario.

2. Literature review

In this section, we propose an input-type classification inspired by the taxonomy presented in Cao et al. (2022). For the sake of simplicity, we have merged the 8 input-type categories proposed by Cao et al. (2022) into 3 groups listed in Table 1, namely (i) *production data*, (ii) *well and completion* and (iii) *reservoir and fluid data*. We discriminate the components of (i), (ii) and (iii) in Table 1. *We emphasize that the present work relies only on input variables derived from historical production data, as defined by the first three rows of the first column of* Table 1, to train machine learning algorithms to predict full-field oil production.

Related works on oil production forecasting are structured according to the forecast's scope: well-level and full-field predictions. While well-level forecasting has received more attention, this work focuses on short and medium-term full-field production forecasts, addressing a research gap that has yet to be fully explored in terms of appropriate algorithms, input space definition, and model evaluation methodologies. In addition, we further divide the literature survey into single-point and time-series forecasts. The former considers the prediction of a single variable, such as recovery factor or 180-day cumulative oil production, and the latter consists in the timeseries prediction of monthly or daily oil production curve.

For unconventional wells, where hydraulic fracturing plays a dominant role, the input space for oil production forecast includes well and completion data (Liang and Zhao, 2019). The best results

Table 1
Taxonomy of variables commonly used as inputs for machine learning algorithms in the oil production forecast literature.

Production data	Well & Completion	Reservoir and Fluid
Produced fluid flow rates	Completion type	Well-logs
Injected fluid flow rates	Well depth	Petrophysical properties
Number of active wells	Perforation thickness	Geological properties
Bottomhole pressure	Hydraulic fracture parameters	Seismic models
	Type of stimulation	PVT data
	Proppant	Relative permeability curve

for single-point forecasts in unconventional plays have been found in tree-based ensemble models (Liang et al., 2023), such as random forest (RF) and gradient boosting (GB). These models excel in interpretability (Liu et al., 2022; Xue et al., 2021) and commonly use a low dimensional input space containing well and completion variables (Liao et al., 2020; Lolon et al., 2016). Despite their good performance in single-point forecasts, RF and GB are unsuitable for timeseries prediction of production data. Their inherent stepwise constant representation fails to express smooth functions (Hastie et al., 2009).

Estimation of well-level production in conventional fields require reservoir and production data. Noshi (2019) exemplifies this point by using inputs like past production, injection rates, and bottom-hole pressure to predict oil production at the well-level. We remark on the unjustified preference for deep learning models: Sheikhoushaghi et al. (2021) considers only deep-learning models (RoughNN, CNN, LSTM) for well-level production forecast, despite the low input space dimensionality (only five input variables). Werneck et al. (2022) only compares deep-learning architectures such as GRU, GRU + CNN and CNN to predict oil-rate and pressure at the well-level. Predictions are performed using a 30-day horizon and a single train-test split is used to evaluate the models. Alali and Horne (2023) and Cruz and Pyrcz (2024) implement the intricate Transformers (Vaswani et al., 2017) architecture for predicting oil production at the well level. Our work conducts a systematic and multi-dimensional comparison, across four oil fields and three forecast horizons, between traditional machine-learning (such as kernel-based models) and neural-based architectures in the context of full-field production forecast.

Noshi (2019), Sheikhoushaghi et al. (2021), Werneck et al. (2022), Ng et al. (2022) and Alimohammadi et al. (2020) all make predictions for a daily horizon. During such a short timeframe, the assumption of stationarity is valid, and, as a result, learned patterns remain unchanged. However, over the course of several months, non-stationarities caused by changes in reservoir and operational conditions are likely to disrupt the patterns learned by a data-driven model. In this work, we focus on a longer timeframe: 6, 12 and 24 months. During this timeframe, non-stationarity prevails, and the statistical model must be sequentially recalibrated and evaluated under different scenarios. As a result, we favor a walk-forward validation approach, applied in Kubota and Reinert (2019) and Davtyan et al. (2020), over a single train-test split to obtain a robust estimate of the generalization error.

For full-field production forecasting, well and completion features, capturing local information, become irrelevant. For field recovery factor (tabular single-point prediction), Tewari et al. (2019) and Makhotin et al. (2022) achieve high accuracy (R2 > 90%) using tree-based ensembles trained on reservoir and production data. Aliyuda et al. (2020) achieve good results for full-field maximum oil-rate with RF. The problem of estimating full-field oil production is discussed in Wang et al. (2020) and Sagheer and Kotb (2019). Only deep-learning models are considered, and a single train-test split is used for model evaluation. As explained, due to the non-stationarity, a single train-test split provides a biased estimate of the generalization error over monthly horizons (James et al., 2021). The walk-forward validation method, implemented in the present work, would have been preferable. It repeatedly evaluates the model under different out-of-sample sets of varying degrees of complexity, which captures the prediction error variance and model behavior across different regimes.

This work bypasses traditional statistical methods like ARIMA and NARX, focusing on deep learning and machine learning for oil production forecasting. Sagheer and Kotb (2019), Ning et al. (2022), and Lara-Benitez et al. (2021) demonstrate the consistent superiority of machine learning over traditional statistical methods due to limitations like stationarity requirements, ergodicity, and linear combination of inputs (Lara-Benítez et al., 2021). Deep learning and machine learning, on the other hand, can identify patterns in time series without any statistical assumptions regarding ergodicity and stationarity. They perform a non-linear composition of *high-dimensional hidden states*, which substantially increases their representational capacity relative to traditional linear statistical methods (Kumar et al., 2023).

3. Datasets

The proposed methodology for full-field oil production forecast will be applied to three Brazilian oil fields (whose names will not be disclosed for the sake of confidentiality) and the Volve dataset. Details for each field are provided in Table 2. Field 1 (F1) and Field 2 (F2) are both located offshore in deep waters. They also share some geological similarities. Their main reservoirs are deep water marine turbidite sandstones. And they both feature a carbonate reservoir, albeit of lesser volume. In terms of rock and fluid properties, F1's permeability exceeds 2000 mD and the oil API ranges between 13 and 29. F2's permeability varies between 200 mD to 2000 mD, less permeable than F1, and its oil API varies between 17 and 31. F1 and F2 are in the top quartile of the most prolific sandstone oil fields in Brazil, which is the motivation for our choice. Field 3 (F3), on the other hand, is located onshore. F3 is composed of sandstone conglomerates formed in a depositional environment of three types: fluvial, fluvial deltaic and alluvial fan. Its oil is more viscous than F1 and F2 with API ranging from 18 to 22. F3 has been in production for over 50 years, and it is one of the most prolific and longest living onshore oil fields in Brazil. F3 provides us with a dataset of 633 monthly-sampled data.

Volve (or V) is an oil field in the North Sea located in shallow waters. The Volve dataset was made public by Equinor (2018). It contains a wealth of data such as geophysical, production, logging, well testing. Volve field has a sandstone reservoir of the Middle Jurassic age. The oil has an API of 27 (black oil) and a viscosity of around 1 cp. In this work, we have used only the production data from the Volve dataset. Following a good practice recommended in Lones (2021), we perform our study using multiple and dissimilar datasets. As shown in Table 2, the selected fields contain different rock and fluid characteristics. Following the taxonomy proposed in Table 1, we consider only production data. Produced and injected flow rates are frequently monitored as they form the basis for royalty calculation and reservoir material balance evaluation. Therefore, unlike geological, seismic and petrophysical data, production data is more likely to be readily available in any oil field. That allows for scalability and reproducibility of a production forecast methodology based exclusively on production data.

We are concerned with making production forecasts for assisting operational oilfield management where forecast horizons range from 12 to 24 months. As a result, the monthly-sampled datasets are relatively small containing a few hundred samples. Daily production forecasts are outside the scope of the present study. Fig. 1 shows for F1, F2, F3 and Volve the monthly-sampled time series of oil rate q_o, injected water rate q_{wi}, number of active producers n_p and the ratio f_w of produced water rate q_w and produced liquid rate defined as $q_o + q_w$, $f_w = \frac{q_w}{q_o+q_w}$. All flow rates are measured at the surface. We note, in Fig. 1, that the full-field f_w shows a *continuous and monotonic upward trend*, in particular for F3. This trend is broken in F1, F2 and Volve (all offshore oil fields) when new producers are added, draining oil from new and untapped reservoirs. In F3, the added producers are only performing

Table 2
Geological and fluid-related information for the three oil fields considered in this work.

Field	Location	Rock-Type	Permeability	API Range	Oil-rate	# of samples
F1	Offshore deep	Turbidite sandstones and carbonates	High (2000 mD)	13–29	18,000 m^3/d in Aug/2020	132 samples
F2	Offshore deep	Turbidite sandstones and carbonates	Medium (200–2000 mD)	17–31	24,000 m^3/d in Aug/2020	132 samples
F3	Onshore	Conglomerates and sandstone conglomerates	Medium (200–2000 mD)	18–22	1100 m^3/d in Aug/2020	576 samples
Volve	Offshore shallow	Sandstone	High (1000 mD)	27	235 m^3/d in Jul/2016	60 samples

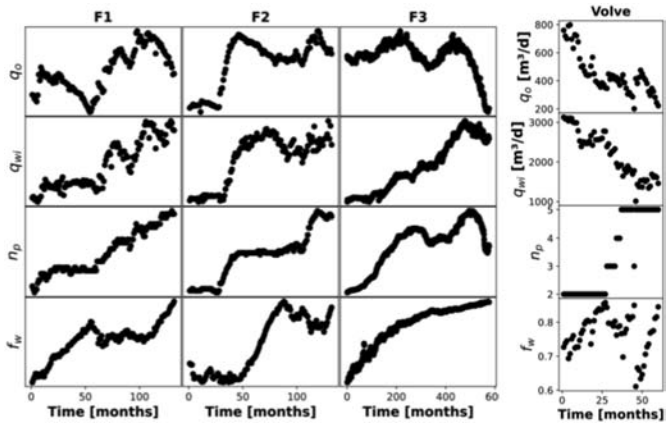


Fig. 1. Datasets for Field 1 (F1), Field 2 (F2), Field 3 (F3) and Volve. The vertical axis, for all four variables in F1, F2 and F3, is scaled to the 0–1 range for the sake of confidentiality. Monthly-sampled time series of full-field oil flow rate q_o, injected water rate q_{wi}, number of active producers n_p and the ratio f_w of produced water rate q_w and produced liquid rate, $q_o + q_w$, $f_w = \frac{q_w}{q_o+q_w}$.

infill drilling of the existing reservoirs, which are all produced in a commingled fashion.

The time series for oil flowrate q_o, injected water flowrate q_{wi}, number of active producers n_p and the ratio f_w are scaled to the 0–1 range using the relationship $z = \frac{x-x_{min}}{x_{max}-x_{min}}$ (except for the Volve field). This is applied to all plots in this work featuring data from F1, F2 and F3 due to confidentiality. We note that the raw data does not come with any missing values or outliers.

4. Methodology

4.1. Machine learning algorithms: a unified description in the context of production forecast

In this work, full-field oil production forecast is posed as a supervised learning problem, where the relationship between full-field oil rate q_o and the input space X (to be defined in Section 4.4) is learned using machine learning algorithms trained on historical production data. We consider three traditional machine learning algorithms: linear regression (LR), kernel-ridge regression (KR) and support-vector regression (SVR). We also consider three deep learning methods: multi-layer perceptron (MLP), convolutional neural network (CNN) and long short-term memory (LSTM) network. In this work, the term deep learning emphasizes the use of hierarchical representations within neural architectures, which may exist independently of the number of stacked layers. All six algorithms can express non-linear and smooth functions (Murphy, 2012). Even LR, when implemented with a sliding-window approach, can also span a globally nonlinear function by stitching together piecewise linear segments, as shown in Kubota and Gioria (2022) for reservoir average pressure prediction and Kubota and Reinert (2019) for oil production forecast. Since the production forecast curve must be smooth, K-nearest neighbors and tree-based ensembles, such as random forests and gradient boosting, are undesirable, as they express piecewise constant functions (Hastie et al., 2009).

For the sake of brevity, we choose to skip providing a detailed description of the six algorithms used in this work. We refer the interested reader to Hastie et al. (2009) and Murphy (2012) for information on LR, KR and Goodfellow et al. (2016) and Aggarwal (2018) for further details in MLP, CNN and LSTM. *Instead, we would like to structure the presentation and comparison of the six algorithms in terms of how the raw inputs are mapped to the full-field oil rate:* from a linear combination of the raw inputs progressively evolving to a linear combination of learned high-dimensional representations. Each algorithm has a distinctive approach for learning those representations (Aggarwal, 2018). Deep learning implements a more complex and abstract raw input transformation than traditional machine learning algorithms.

The most basic choice is represented by LR. In this work, we implement LR using sliding windows. As a result, LR can span a nonlinear function by stitching together piecewise linear segments. LR expresses flowrate as a linear combination of the low-dimensional raw inputs, as shown in Eq. (1), where the weights w_i are learned from the samples inside the fixed-size sliding window. Fairly accurate production forecasts are reported in Kubota and Souto (2019) and Kubota and Reinert (2019) from the application of LR with sliding windows for the problem of full-field production forecast. However, the authors fail to address whether the introduction of more complex and abstract representations of the input space X could improve generalization. Eq. (1) could be viewed as a model *without any hidden layer* as the inputs are directly mapped, via output layer (containing a single linear activation), to the full-field flowrate. What if we introduce one hidden layer that expands the input space dimensionality with the creation of hidden representations?

$$q_o(x) = \sum_{i=1}^{p} w_i x_i \tag{1}$$

Kernel-based models such as KR and SVR (Hastie et al., 2009; Haykin, 2009; Murphy, 2012) can expand the input space dimensionality. Both models can be jointly represented by Eq. (2) and depicted in Fig. 2. The differences are (i) in KR, N refers to number of training samples and in SVR N would be replaced by the number of support vectors (#SV), (ii) the cost function minimized in KR is the squared-error with ridge regularization (yielding a dense solution), whereas in SVR the minimized cost function is the hinge-loss, (Awad and Khanna, 2015), which leads to a sparse solution in terms of SVs, and (iii) SVR contains an explicit bias term, which is omitted in Eq. (2) for the sake of simplicity.

$$q_o(x) = \sum_{i=1}^{N \text{ or } \#SV} w_i K(x, x_i) \tag{2}$$

Both KR and SVR could be viewed as a network featuring *only one hidden layer* before the output layer (Bengio, 2011). The hidden layer is trained in an unsupervised manner using kernel-based similarity functions, K, while the output layer is trained in a

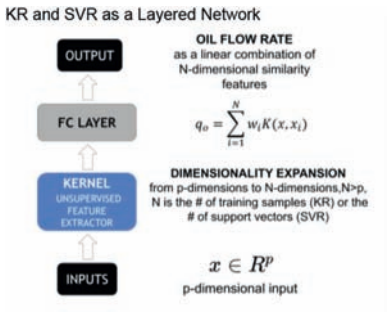


Fig. 2. Representation of kernel-ridge regression (KR) and support vector regression (SVR) as a layered network. The hidden layer implements dimensionality expansion and unsupervised feature extraction using similarity kernels. The output layer is a fully-connected (FC) one that linearly combines the similarity features $K(x, x_i)$ using weights defined in a supervised manner.

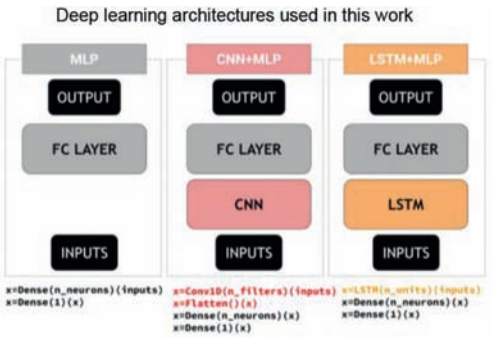


Fig. 3. Three deep-learning architectures implemented in this work. The CNN and LSTM layers are introduced as specialized feature extractors leveraging domain-specific properties such as local correlation and translation invariance. They are inserted between the input layer and the fully connected (or dense) layers from MLP. The notation from Keras functional API is used to represent the deep learning models in this figure.

supervised manner. Unsupervised feature learning affords robustness to output noise, which helps avoid overfitting. In analogy with Eq. (1), we could say Eq. (2) expresses the oil-rate as a *linear combination of high-dimensional similarity features* computed between the raw input x and the training samples x_i, in KR, or support vectors, in SVR, by means of kernels $K(x, x_i)$. Kernel-based models can handle problems featuring relatively small datasets, such as oil production forecast (Davtyan et al., 2020), skin factor monitoring in oil producers (Kubota and Gioria, 2022) and financial time series (Wolberg, 2000).

In the deep learning paradigm, instead of expressing the full-field oil rate as a linear combination of the low dimensional raw inputs (as in LR), or unsupervised high-dimensional similarity features (as in KR and SVR), we could write it as a *linear combination of problem-specific features, or hidden representations* h_i, learned from data in a modular, hierarchical and supervised manner using sequential layers of transformations (Lim and Zohren, 2021). The parameter n in Eq. (3) denotes the number of neurons in the last hidden layer. As a result, while x is p-dimensional, h is n-dimensional. Deep learning architectures can leverage both dimensionality expansion or width (when n > p) and depth (composition of nonlinear functions via sequential layers) to enhance their representational power (Chollet, 2017).

$$q_o(x) = \sum_{i=1}^{n} w_i h_i(x) \tag{3}$$

The representation at layer l h_l relates to the representation in the previous layer l-1 h_{l-1} by means of $h_l = a(W_{l-1}h_{l-1} + b_{l-1})$ where the weight matrix W_{l-1} (learned via gradient-based optimization) is dense for MLP and sparse in CNN. In addition, a denotes a nonlinear activation function (applied element-wise on the result of the affine transformation $W_{l-1}h_{l-1} + b_{l-1}$) and b is the bias term.

Standard MLPs lack inductive biases (Murphy, 2012) which capture local patterns and recent trends (CNNs), and long-term dependencies (LSTMs). We leverage CNN and LSTM modules to learn these specialized features, which are then fed into a fully connected layer, thus improving upon the basic MLP architecture that relies solely on raw input data, as shown in Fig. 3. CNNs efficiently learn local features using sliding filters (or kernels, defined by the hyperparameter "kernel_size"), while LSTMs process data sequentially to capture temporal relationships. Both models implement weight sharing and produce feature maps (whose dimension is set by the "n_filters" hyperparameter in CNN and "n_units" in LSTM) which are used as inputs to the dense, fully connected layers on top. Due to our limited dataset size, we avoid stacking multiple layers to maintain model simplicity and help generalization (Lones, 2021). We emphasize the term deep learning, in this work, is associated with the use of neural-based modules leveraging hierarchical representations. They are not deep in the sense of stacking multiple layers. In Fig. 3, we use the notation from Keras functional API to represent the models. The range of model hyperparameters is provided in Section 4.5.

We note that:

- Eq. (1) expresses full-field oil-rate as a linear combination of the p-dimensional raw inputs.
- Eq. (2) as a linear combination of kernel-based (therefore parametrized and unsupervised) similarity features of dimensionality N (or the number of support vectors, for SVR).
- Eq. (3) as a linear combination of n-dimensional learned features extracted from input data in a modular, hierarchical and supervised manner using neural-based architectures leveraging inductive biases.
- The components of the linear combination become increasingly abstract as we progress from Eq. (1) to Eq. (3).
- Expressive power and model variance increases from Eq. (1) to Eq. (3).

We are interested in investigating whether hierarchical and modular feature learning represented by Eq. (3) and implemented with varying architectural choices as MLP, CNN + MLP and LSTM + MLP, as shown in Fig. 3, offer any advantage in terms of generalization over the shallow and less abstract ones in Eq. (1) and Eq. (2) in the context of full-field oil production forecast. As previously pointed out, this question has yet to be properly addressed in the oil production forecast literature. It also aligns with the Green AI initiative (Xu et al., 2021), which advocates in favor of simpler models. The hyperparameter space considered for each discussed model will be detailed in Section 4.5.

4.2. Model training and validation

Models are trained using a window-based approach and evaluated using walk-forward validation, discussed in Pardo (2012), Chan (2008), and Lopez de Prado (2018). Walk-forward validation is the time-series version of the k-fold cross-validation, as it preserves the temporal order in the dataset. It provides an estimate of the generalization error. Joseph (2022) remarks that the standard machine learning practice of splitting the dataset into a single train and validation (or test) set leads to a biased and non-

robust estimate of the generalization error in the context of nonstationary timeseries, such as the problem of short- and medium-term production forecasts. However, the use of a single split is commonly encountered in related works, such as Wang et al. (2020) and Sagheer and Kotb (2019), to name a few. In contrast, walk-forward validation repeatedly trains and evaluates model predictions on multiple out-of-sample folds. This results in a more robust estimate of the generalization error, as it sheds light on the model's ability to generalize across different regimes, instead of a single out-of-sample set (Cerqueira et al., 2019).

Window-based training and walk-forward validation are based on the following steps:

(1) An expanding/sliding window of size W is used to train the model.
(2) The trained model predicts $\hat{y}$ over a forecast horizon of length H.
(3) Using the metric of your choice, calculate the divergence between the H predicted $\hat{y}$ and known y target values.
(4) The expanding/rolling window absorbs the H samples of input and target values (X, y) into the training set.
(5) Return to step 1 or terminate the process if the end of sequence is reached.

Window-based training allows to recalibrate the model under nonstationary conditions (Huang and Wang, 2023). Shifts in the data-generating distribution in oil fields might be caused by changes in operational conditions, such as the sudden halt of water injection pumps. Likewise, changes in reservoir conditions associated with a rise in water cut or the onset of a new oil-recovery method (water injection, steam injection, unseen in the training set) may also cause a shift in the data-generating distribution (Kubota and Gioria, 2022). In the window-based regression framework, model training can be performed using either an expanding or sliding window, as shown in Fig. 4. In the expanding window configuration, the input dimensionality remains constant; only the total number of training samples increases over time.

LR is always trained with a sliding-window strategy, which makes it possible for LR to handle non-stationarities and express non-linearities by combining piecewise linear functions (Davtyan et al., 2020; Kubota and Reinert, 2019). Those two works also provide us with guidelines regarding the window size. In both, a window size of 24 or 36 months is reported to lead to good results for monthly forecast horizons. Deep-learning algorithms are all trained using the expanding window-approach, as they inevitably require larger datasets. With regards to KR and SVR, the best window type (sliding or expanding) is data-dependent and treated as a hyperparameter). For computational reasons, we begin inference after an initial time lapse longer than the size of the training window W. Aggarwal (2018) notes that data normalization can help with model training. For neural-based architectures, we apply min-max normalization (scaling to the 0–1 range using $\frac{x-x_{min}}{x_{max}-x_{min}}$). For kernel-based models, z-score normalization is applied (zero mean and unit standard deviation). Since model training and evaluation are performed multiple times, the walk-forward validation might be more computationally expensive. However, bear in mind the monthly datasets used in this work are relatively small. Therefore, in practice, the computational cost is relatively low even for neural-based architectures (a few minutes at most on a Google Colab cloud-based CPU).

4.3. Model evaluation metrics

We use two model evaluation metrics: the mean absolute percentage error (MAPE), shown in Eq. (4), and R-squared (R2), shown in Eq. (5). Both MAPE and R2 are computed using blind out-of-sample sets from walk-forward validation. The out-of-sample MAPE from walk-forward validation provides an estimate of the generalization error associated with an algorithm (Joseph, 2022). MAPE was used in the first editions of the well-known time series M-competition (Makridakis et al., 1982) and in the context of oil production forecast (Khanal et al., 2017; Yuan et al., 2021). MAPE can be regarded as a proxy for model bias and R2 as a proxy for model variance. The usage of multiple metrics is deemed as a good practice by Lones (2021), since different metrics convey different perspectives. For this reason, we avoid reporting a single metric in this work. In the context of time-series problems, the use of MAPE is a more intuitive metric than RMSE (Kim and Kim, 2016) used in several works devoted to production forecast, and sMAPE (Werneck et al., 2022). MAPE is better understood by stakeholders (Kim and Kim, 2016), which is an important criterion in the deployment of data-driven models (Géron, 2017).

$$MAPE_{out-of-sample} = \frac{1}{N}\sum_{i}\left|\frac{q_o[i] - \widehat{q}_o[i]}{q_o[i]}\right| \tag{4}$$

$$R^2_{out-of-sample} = 1 - \frac{\sum_i (q_o[i] - \widehat{q}_o[i])^2}{\sum_i (q_o[i] - \overline{q}_o[i])^2} \tag{5}$$

4.4. Model inputs

We propose a multivariate approach that can be used as a *low cost, full-field data-driven proxy model in scenario analysis to guide operational decisions and cash-flow evaluation of oil fields*. In the multivariate time-series (MTS) problem of oil production forecast, in addition to auto-regressive inputs $q_o[t-1], q_o[t-2], \ldots, q_o[t-k]$, which provide *context information* (the prediction of q_o at t might depend on what happened to q_o in $t-1$, $t-2$, …, $t-k$), we also consider three non-autoregressive input variables motivated by

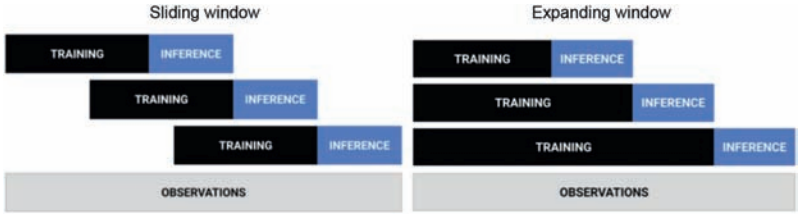


Fig. 4. Two configurations of model training in window-based regression. The model performance during inference is repeatedly evaluated using multiple out-of-sample sets from different unseen time periods resulting in a reliable estimate of its performance over time and a robust estimate of the generalization error.

domain knowledge, namely q_{wi} the full-field water rate injection, n_p the number of active producers and f_w the full-field water cut defined as $f_w = \frac{q_w}{q_w+q_o}$, where q_w is the full-field produced water flow rate. Latrach et al. (2023) argues that the use of physics and domain knowledge to guide the choice of input variables has a regularizing effect which helps avoid overfitting in our context of limited datasets. Compared to Capacitance-Resistance models (Holanda et al., 2018), which are valid for well-level analysis and do not consider well inventory n_p, our multivariate approach is valid for the full-field context. Its non-parametric, flexible and nonlinear nature (Liu, 2022) is not constrained by linearity assumptions present in Capacitance-Resistance models.

We pose full-field oil production forecasting as an MTS problem, as shown in Eq. (6). Function g maps from $X = (q_o[t-1], q_o[t-2], \ldots, q_o[t-k], q_{wi}, n_p, f_w) \in \mathbb{R}^{3+k}$ to $y = q_o \in \mathbb{R}$. The context window size k (also known as autoregression order) is treated as a hyperparameter. In this work, we consider $k = \{0, 3, 6\}$ months. As evidenced by Kubota and Reinert (2019) and Davtyan et al. (2020), full-field oil production time series has a short memory, so longer context windows were not considered. For $k = 0$, the model has no autoregressive features. The 2dsew method described in the next section is implemented whenever $k > 0$.

$$q_o[t] = g(q_o[t-1], q_o[t-2], \ldots, q_o[t-k], q_{wi}[t], n_p[t], f_w[t]) \quad (6)$$

Each non-autoregressive input accounts for factors of variation in full-field production. For example, q_{wi} and f_w account for material balance and waterflood/fractional flow effects (Dake, 1978), which models the fact that oil gain from boosting injected water rate at 10% water cut will be substantially different from the same increase at 80% water cut. The n_p feature accounts for full-field production boosts caused by an increase in well density due to infill-drilling or well reopening. The q_{wi} and n_p features help determine full-field liquid production, while f_w provides information on how to transform liquid production into full-field oil production.

We make some remarks regarding the non-autoregressive variables, namely q_{wi}, n_p and f_w .

- **Remark 1:** The main application for this model is in scenario analysis as a full-field data-driven proxy model to help determine the best course of action for some operational variables. As a result, field operators can make more informed decisions about how to optimize waterflooding schedules. We can build scenarios for q_{wi} and n_p spanning the next, say 12 months, and use the proxy-model to simulate oil production curves, as exemplified in Section 5.4. By changing scenarios, we can experiment with several operational schedules for q_{wi} and n_p and evaluate their effect on the production curve. Please note, a given scenario might be disrupted by unpredictable events such as operational delays, budget cuts, and others. That is why MTS models are best suited for prescriptive applications, rather than predictive ones, unless a given scenario is 100% guaranteed to take place.
- **Remark 2:** Please note that q_{wi} and n_p can be (at least in theory) imposed by the field operator. For instance, the injected water rate could be brought to zero at any moment. However, f_w has a different nature, as it is associated with the system's dynamics. The range of scenarios for f_w is more limited. If the current f_w is at 90%, we cannot create future scenarios where it is at 50% or 0%. During inference, we propose the adoption of the implementation in Kubota and Reinert (2019) and Kubota and Souto (2019), where a sliding-window linear regression model with respect to time is fitted to the last 12 samples of $f_w(t)$. Once the linear trend parameters are learned, an extrapolation is performed, during inference, over the forecast horizon. As shown and noted in Fig. 1, f_w behavior is monotonic, and it displays a well-defined upward trend. Therefore, it can be easily expressed as a combination of piecewise linear regressions, as noted in Kubota and Reinert (2019).
- **Remark 3:** Physically, it can be shown that f_w is partially related to the fractional flow curve, Dake (1978), which depends on rock and fluid properties. In Fig. 1, especially for F3, we can see that f_w follows a well-defined upward trend regardless of the behavior displayed by the other variables. While f_w contains information on q_o , *it is not possible to infer* q_o *solely from the knowledge of* f_w. For instance, a $f_w = 50\%$ could map to $q_o = 1000$ m^3/d or $q_o = 3000$ m^3/d. Additional information, present in the other input variables, is needed to properly untangle the value of q_o.

If auto-regressive inputs are considered, algorithms are implemented during the inference phase, using what we refer to as a two-step dynamic self-expanding window (2dsew). Our methodological contribution consists in the introduction of a mid-inference retraining step, which is shown to lessen the effect of error propagation in multi-step forecasting. Retraining is performed using a dataset containing original and model-generated values (predicted flowrates). The term dynamic relates to the method of recursively using, during inference, predicted values as inputs to forecast new ones, implemented in Sagheer and Kotb (2019) and Joseph (2022) in the context of time series and Vaswani et al. (2017) for natural language processing. We distinguish between two regimes: training and inference.

- During *training*, models learn to predict one month ahead.
- During *inference*, for a forecast horizon of H months, two steps are taken:

  - **Step one**: recursively predict the first $H/2$ months using predicted values as inputs.
  - **Step two**: *retrain* the model after expanding the training window with *model-generated data* (hence the name self-expanding, as it does not rely on ground-truth flowrate values), then recursively predict for the next $H/2$ months.

After H months have passed, the model-generated flowrates used to expand the training set are replaced with the ground-truth targets, which are now available (e.g., $\widehat{q}_o[t-1]$ is replaced with $q_o[t-1]$, and likewise for the other lags). The introduction of a retraining step leveraging generative data augmentation improves generalization and lessens the effect of error propagation (Davtyan et al., 2020; Zeng et al., 2022) present in the fully recursive method for multi-step forecasting. Fig. 5 shows two implementations of the KR model to predict full-field production of field F1. The implementation in the top chart recursively predicts the next 12 months, without any retraining. The implementation in the bottom chart uses the 2dsew methodology, which leads to noticeably superior results, as the effect of error propagation is reduced due to mid-inference model retraining using original and model-generated data.

The 2dsew method is graphically illustrated in Fig. 6 for a forecast horizon of 6 months and a context window of 3 months. The window sizes were chosen to make visualization easier. The inclusion of the retraining step using a dataset containing a mix of original and model-generated data leads to superior results relative to recursively predicting for the next H months without any retraining. As shown in Fig. 5, it lessens the effects of error propagation particularly for long forecast horizons (such as 12 and 24

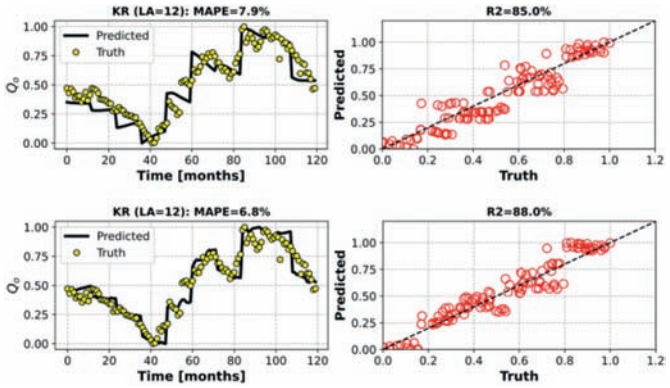


Fig. 5. The effect of 2dsew in lessening error propagation. The top chart uses the KR model, implemented without 2dsew, to recursively predict the oil flow rate over a forecast horizon of 12 months. The bottom chart implements the 2dsew procedure in which the first 6 months are recursively predicted, followed by a retraining step using a dataset containing original and model-generated data, and then recursive prediction for another 6 months. The implementation of 2dsew seems to improve generalization and lessens the effects of error propagation associated with the fully recursive method.

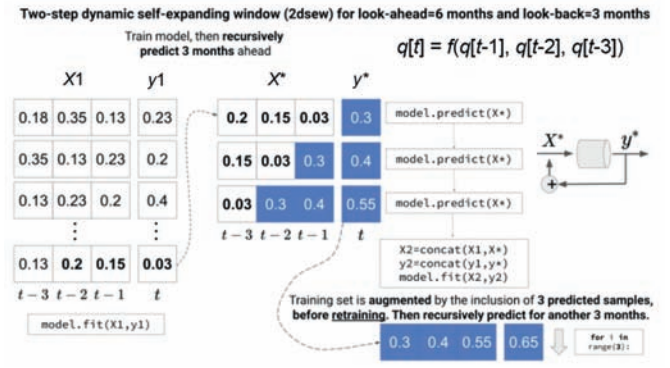


Fig. 6. Two-step dynamic self-expanding window exemplified for a forecast horizon of six months and a context window of three months (these horizons were chosen for the sake of ease of visualization). This implementation is used whenever auto-regressive inputs are considered. During training, the model learns to predict one month ahead. During inference, two steps are taken. First step: the trained model recursively predicts (using predicted values as inputs for subsequent predictions) for half the size of the forecast horizon ("model.predict" applied 3 times). The training set is augmented by including predicted values ("concat"), hence the qualifier "self-expanding". Second step: a retraining step using the augmented set is performed ("model.fit (X2, y2)") before the model recursively predicts over the second half of the forecast horizon ("for loop" 3 times). The inclusion of the mid-inference retraining step, instead of recursively predicting for the entire duration of forecast horizon, lessens the effects of error propagation associated with fully recursive predictions.

months) and improves generalization. The 2dsew method also avoids the increased variance associated with multi-output models (Zeng et al., 2022).

4.5. Model hyperparameters and model ensembling

We do not implement hyperparameter optimization. Challenges associated with hyperparameter optimization in nonstationary sequential data with small datasets are discussed in Lee et al. (2024) and Schneider et al. (2025). In the context of small datasets, hyperparameter optimization may lead to over tuning, a type of overfitting that hurts model generalization (Schneider et al., 2025). In the proposed workflow, we implement the ensemble averaging method (Hastie et al., 2009; Joseph, 2022). The members of the ensemble are different configurations of the same algorithm, as proposed in Lakshminarayanan et al. (2017) and Aggarwal (2018), where it is called the K-best approach. The final prediction is obtained by averaging the predictions from the K-best model configurations, which reduces variance and avoids overfitting (Koren, 2009). Instead of cherry-picking the single best hyperparameter configuration that minimizes the walk-forward validation MAPE for a given algorithm, we proceed in the following manner, inspired by Aggarwal (2018):

(1) Given an algorithm, such as CNN, define a list of values for each hyperparameter. For example, number of neurons = [32,64,128] and number of filters = [16,32,64]. The hyperparameter space is presented in Table 3.
(2) Perform walk-forward analysis for each combination of hyperparameters defined in the previous step to obtain MAPE and R2.
(3) Rank each model configuration from lowest MAPE, followed by highest R2.
(4) After ranking, return the K-best models. We adopt $K = 5$ to consider configurations inside the top quartile, given the cardinality of the hyperparameter space detailed in Table 3.
(5) Average the predictions from the K-best model configurations selected in step 4.

Combining uncorrelated forecasts, achieved by averaging predictions from different model configurations (each potentially reaching a distinct local minimum), reduces prediction variance and improves generalization, particularly for nonstationary time series data (Aggarwal, 2018; Galicia et al., 2019;

Table 3
Hyper-parameter space configuration.

Model	Hyper-parameter (hp) space	Number of configurations in hp-space
LR	**sliding window size** (5) = [12,24,36,48,60] **context window** (3) = [0,3,6]	15
KR	**kernel** (3) = [gaussian, linear, polynomial] **alpha (regularization strength)** (4) = [1.0,0.1,0.01,0.001] **gamma (visual field)** (4) = [1.0,0.1,0.01,0.001] **context window** (3) = [0,3,6] **use sliding window** (2) = ["Y", "N"]	216
SVR	**kernel** (3) = [gaussian, linear, polynomial] **C (regularization strength)** (4) = [1.0,10,100,1000] **gamma (visual field)** (4) = [1.0,0.1,0.01,0.001] **context window** (3) = [0,3,6] **use sliding window** (2) = ["Y", "N"]	216
MLP	**context window** (3) = [0,3,6] **# of neurons** (4) = [32,64,128,256] **epochs** (2) = [20,100] **activation** (1) = [relu]	24
CNN	**context window** (3) = [0,3,6] **# of filters = # of neurons** (4) = [32,64,128,256] **kernel size** (3) = [1,3,5] **epochs** (2) = [20,100] **activation** (1) = [relu]	72
LSTM	**context window** (3) = [0,3,6] **# of lstm units = # neurons** (4) = [32,64,128,256] **epochs** (2) = [20,100] **activation** (1) = [relu]	24

Lakshminarayanan et al., 2017). This approach mitigates the risk of overfitting in walk-forward validation, common in finance (Lopez de Prado, 2018) and avoids rewarding a single cherry-picked hyperparameter configuration obtained from cycles of back-testing and hyperparameter tuning. Furthermore, averaging predictions flattens the test loss surface, reducing the impact of sharp local minima that can trap individual models (Goodfellow et al., 2016; Prince, 2023).

Since the monthly-sampled datasets used in this work are relatively small, the runtime of the proposed implementation is of a few seconds for shallow architectures and a few minutes, at most, for deep learning algorithms in a cloud-based CPU (even faster in GPU) hosted on Google Colab. We take advantage of the relatively small datasets of monthly-sampled production data. Those methodological choices (walk-forward validation and ensemble averaging) would not have been feasible in datasets of thousands of samples, due to high computational cost, an obstacle to the K-best method noted in Aggarwal (2018). However, for monthly sampled data, most mature oil fields will have datasets of a few hundred samples at most.

Table 3 shows the list of hyperparameters (and their cardinality) considered for each model used in this work. For ease of visualization, we also list the number of configurations in the hyperparameter space. The context window size applies only when auto-regressive inputs are considered. We may have $k = 0$, a configuration without auto-regressive inputs. The hyperparameters that are not featured in Table 3 are used in their default Keras configuration, version 2.12.0. All deep learning architectures use Adam as the optimizer with Keras default configurations. The optimized metric during training is the mean-squared error.

For deep learning algorithms, we report that the use of regularization strategies such as dropout layers and weight decay have a negative impact on prediction accuracy. Goodfellow et al. (2016) and Nakkiran et al. (2019) remark that the number of epochs can mimic the role of the regularization strength parameter. In this regard, more epochs imply less regularization. We consider the number of epochs as a hyperparameter for the set of deep learning algorithms. Configurations trained with fewer epochs can be deemed as regularized in the sense described in Goodfellow et al. (2016).

4.6. Workflow for constructing the full-field data-driven proxy model

The construction of the full-field data-driven proxy model requires the following steps:

(1) Prepare time series dataset of full-field historical production data (produced and injected flowrates, number of active producers, as in Table 1) for the oilfield of interest.
(2) Select a machine learning algorithm along with its hyperparameter configuration (Table 3).
(3) Evaluate MAPE and R2 for each configuration in hyperparameter space using walk-forward validation.

a. Consider that $X = (q_o[t-1], q_o[t-2], \ldots, q_o[t-k], q_{wi}, n_p, f_w) \in \mathbb{R}^{3+k}$, the context window k is treated as a hyperparameter and $y = q_o \in \mathbb{R}$.
b. If $k > 0$, use 2dsew method.

(4) Rank each model configuration from lowest MAPE, followed by highest R2.
(5) The ensemble of the five best model configurations (the K-Best method) can be used as the data-driven proxy model.
(6) Produce a scenario for q_{wi} and n_p extending over, for instance, 12 months.
(7) Combine, via simple averaging, the predictions from the configurations in step 5 considering the scenario in step 6.
(8) Output: multi-step full-field oil production curve extending over 12 months.

In Fig. 7, we provide a summary of the central elements of the proposed workflow. Section 5.4 exemplifies the application of the data-driven proxy model in oilfield management.

5. Results and discussions

5.1. Is deep learning all you need? Comparative analysis between traditional machine learning and deep learning

A multi-dimensional analysis conducted across 4 fields, 6 algorithms, and 3 forecast horizons suggests deep learning offers no benefit in terms of generalization in the context of short and medium-term full field oil production forecasts. In Figs. 8 and 9, oilfields (F1, F2, F3 and Volve) are organized row-wise, and forecast horizons (6, 12 and 24 months) are organized column-wise. In Fig. 8, each point in the MAPE-R2 plane is associated with a different hyperparameter configuration (using the hyperparameter space defined in Table 3). MAPE and R2 are computed by walking forward each individual hyperparameter configuration. The closer any given configuration lies to the top-left corner (high R2, low MAPE) the better it can generalize for out-of-sample sets. We cluster points into two classes: kernel-based (KR and SVR, represented with "o") and neural-based models (MLP, CNN, LSTM, represented with "*").

Fig. 8 suggests i) neural-based models do not outperform kernel-based architectures; ii) it seems possible to find configurations of kernel-based models that perform the same as neural-based ones with a much lower computational cost; iii) from a practical standpoint, the lower MAPE range is always below 10%. Rodriguez et al. (2020), Nazarenko and Zolotukhin (2021) and Yang et al. (2022) argue that forecast errors within 5%–10% range are deemed acceptable, particularly for oil fields with water cut above 10%. The lowest MAPEs are found in F3 and the highest MAPEs in Volve. We will elaborate further on this last observation in the next Section.

In Fig. 9, each model is represented by its out-of-sample (MAPE, R2) associated with the walk-forward analysis of the K-Best

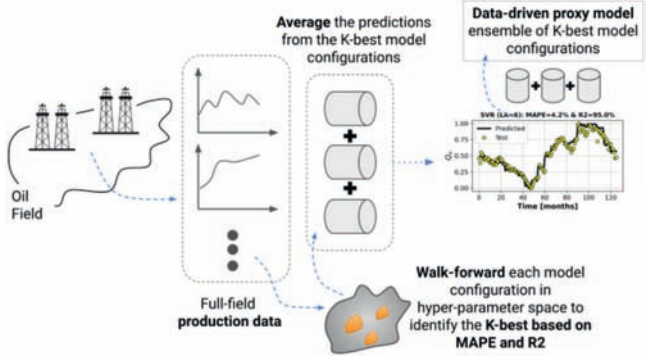


Fig. 7. Overview of the main elements used in the proxy model workflow. Historical full-field production data (full-field produced and injected flowrates, as well as the number of active producers) are used for training algorithms. Walk-forward analysis of different configurations of an algorithm is conducted to identify the K-best configurations. The data-driven proxy model results from the ensemble of the K-best model configurations.

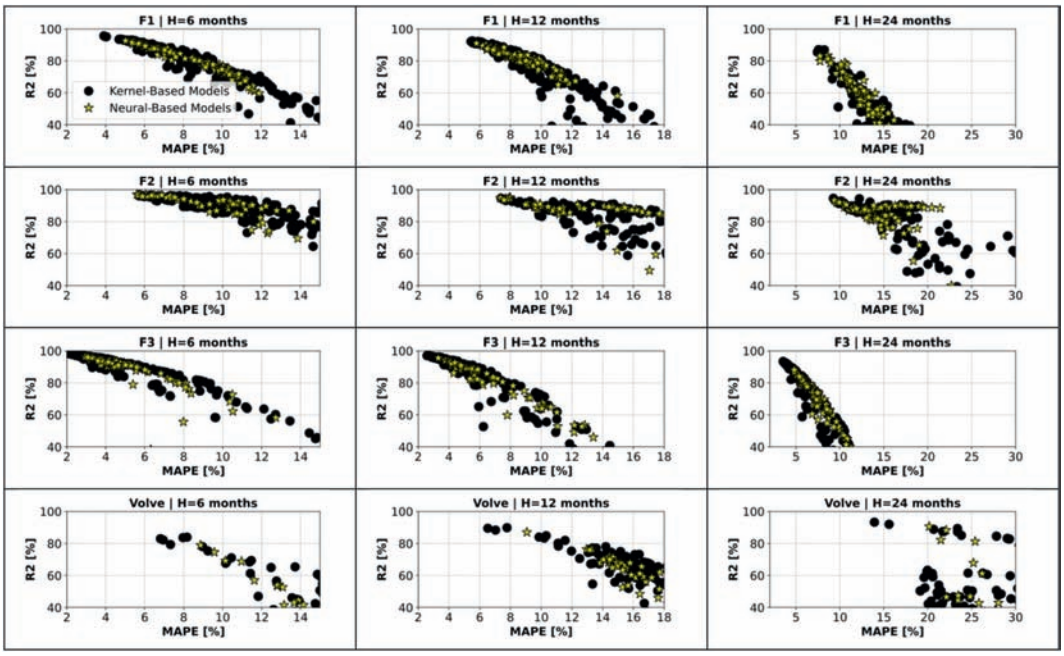


Fig. 8. Comparison between kernel-based (KR and SVR, represented with "o") and neural-based models (MLP, CNN and LSTM, represented with "*") across three forecast horizons (columns: 6, 12 and 24 months) and four oil fields (rows: F1, F2, F3 and Volve). Each point in the MAPE-R2 plane is associated with a different hyperparameter configuration. MAPE and R2 are computed from walk-forward validation.

ensemble. Traditional machine learning algorithms (LR, KR and SVR) are represented with "o" and deep learning algorithms (MLP, CNN and LSTM) are represented with "*". The best performing models are the ones closer to the top left corner in the MAPE-R2 plane (high R2 and low MAPE). The MAPE-R2 plane helps us visualize the relative performance of all six algorithms for each field and forecast horizon (e.g., for F1 and 6-month forecast horizon, SVR is the best model). It also provides an alternative view to Fig. 8 as it relates to the comparative analysis between traditional machine learning (the "o") and neural-based models (the "*") in terms of generalization. Please bear in mind, model comparison based on metrics alone leaves out qualitative aspects such as interpretability, which is an important criterion when machine learning is used to assist decision making (Rudin, 2018). In this regard LR is the most interpretable one, despite its poor-to-fair generalization performance.

We summarize the analysis of Figs. 8 and 9 into the following remarks for ease of reading:

- **Occam's razor.** Domingos (1999) argues on the role of Occam's razor in the field of machine learning: if two models have comparable *generalization error*, the simpler one should be favored. Across all 12 scenarios in Figs. 8 and 9, kernel-based models *either outperforms or matches the performance of neural-based ones. A multi-dimensional analysis conducted across fields, models, and time horizons suggests deep learning offers no benefits in terms of generalization, interpretability, and computational cost in the context of short and medium-term full field oil production forecasts.* This problem is a low-dimensional one, with relatively small and sparse datasets (Liu, 2022) and physics-based inputs. These characteristics combine to favor kernel-based learners over neural-based architectures, as quantified in Figs. 8 and 9. This is in line with Wilson (2025) which argues that deep learning is best-suited for high-dimensional datasets (such as image and natural language processing). On a broader context, our results contribute to including the problem of multi-step timeseries forecasting of full-field petroleum production to a series of studies such as Arik and Pfister (2021), Grinsztajn et al. (2022) and Zeng et al. (2022) calling into question the widespread, and *often unjustified*, *application of deep learning algorithms to low-dimensional structured data and time series problems.* The preference towards deep learning in petroleum production forecasting may be a manifestation of the bias towards new methods discussed in Herrmann et al. (2024) as one of the epistemic challenges associated with the widespread application of data-driven algorithms in applied research.
- **On the effectiveness of kernel-based methods.** Kernel-based models such as KR and SVR are consistently the top performers across all 12 scenarios. From a computational perspective, it follows that the full-field production forecast problem derives more benefits from internal representations leveraging *width* (dimensionality expansion), than *depth* (hierarchical compositionality). We make the following remarks on KR and SVR:
  - o They implement a linear combination of high-dimensional, kernel-based similarity features $K(x, x_i)$ learned in a shallow and unsupervised manner, and therefore *robust to output noise*, which helps avoid overfitting (Aggarwal, 2018). In contrast, the feature learning process in MLP, CNN and LSTM is a supervised one and, hence, less robust to output noise. In addition, the relatively small and low-dimensional datasets associated with monthly, full-field production forecasts, are well-handled by the less complex kernel-based models which can still express nonlinearities with a computationally cheap convex optimization training (Haykin, 2009). These factors shed light on the reasons why kernel-based models are consistently the top performer across all three horizons and all four fields.
  - o Most research papers dealing with oil production forecasts only consider SVRs (Davtyan et al., 2020; Ng et al., 2022; Liang et al., 2023; Atadeger et al., 2023) and fail to even mention KRs. As remarked in Aggarwal (2018) regarding the machine learning

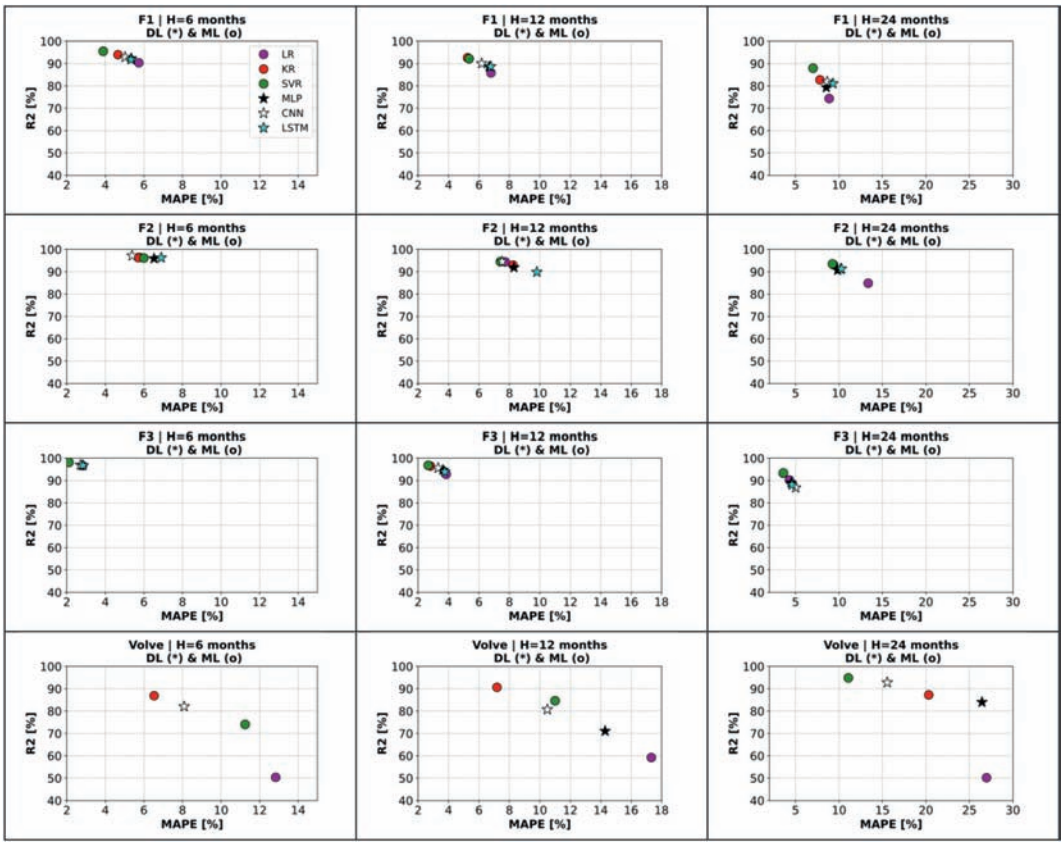


Fig. 9. Comparison between traditional machine learning algorithms (represented with "o") and deep learning algorithms (represented with "*") across three forecast horizons (columns: 6, 12 and 24 months) and four oil fields (rows: F1, F2, F3 and Volve). The performance of each algorithm is represented by the out-of-sample MAPE from walk-forward validation (a measure of the generalization error) and R2. These values become a coordinate on the MAPE-R2 plane. The best performing models are the ones closer to the top left corner in the MAPE-R2 plane. This visualization helps us see the relative performance of all six algorithms for each field and forecast horizon. It also helps confront traditional machine learning (the "o") against deep learning (the "*"). Kernel-based models, such as KR and SVR (instances of traditional machine learning models), are invariably the top performers across all fields and horizons.

literature in general (and we include here the oil production forecast, in particular), KRs full potential remains unexplored, as it was eclipsed by both SVRs and neural networks. A trend that has been recently reversed, as findings in machine learning research suggest kernel methods can approximate neural networks under certain conditions (Chen et al., 2020). Such a claim can be visualized in the clouds of neural-based and kernel-based models of Fig. 8. The results obtained in this work highlight the performance of KR in dealing with full-field production forecasts with relatively small and low-dimensional datasets.

- **On depth versus width.** As pointed out earlier, the full-field production forecast problem seems to benefit more from internal representations leveraging width (dimensionality expansion), than depth (hierarchical compositionality). The automatic and hierarchical feature learning implemented by neural-based architectures at the cost of increased model complexity and the subsequent need for larger datasets does not seem to provide substantial benefits in terms of improved generalization in our context. Moreover, the introduction of specialized CNN or LSTM layers before the fully connected one does not seem to improve generalization. Prince (2023) argues that depth must be justified by the existence of compositionality in the data. Schmidt-Hieber (2020) establishes that in neural networks with ReLU activation, depth should scale logarithmically with dataset size. Hence, for small dataset problems, architectural depth is not encouraged. *Our results suggest full-field oil production data does not exhibit deep compositionality, but rather a shallow piecewise compositionality*, such as the one expressed by LR with sliding windows (Kubota and Reinert, 2019).
- **On the effectiveness of LR.** LR models, *implemented with sliding windows*, perform reasonably well (although not the top performer) for F1, F2 and F3 across all three horizons, as shown in Fig. 9. For those fields, MAPEs from LR are mostly below the 10% range even for the 24-month forecast horizon. This result suggests it could serve as a *good compromise between interpretability, ease of implementation, very low computational cost, and generalization ability*. In a few cases, it even outperforms more complex models, which endorses Green AI (Schwartz et al., 2020). For a forecast horizon of 12 months in F1, the LR model reveals a walk-forward MAPE of 7%, which is acceptable in practice (although not the best). LR model is low-variance, interpretable and can be easily implemented by practicing engineers lacking programming skills, but well-versed in spreadsheets, which allows the replication of the methodology proposed in this work by a much larger audience. Xu et al. (2021) argues that Green AI initiatives help improve AI inclusiveness. We illustrate the application of the LR model as a prescriptive tool in scenario analysis in Section 5.4. Rudin (2018) warns against the use of black box algorithms to assist in decision-making and argues in favor of interpretable models. The effectiveness of the LR model in full-field oil production

forecast is another piece of evidence that suggests traditional and relatively low-variance machine learning algorithms might be well-suited for this task: *full-field production data seems to be reasonably represented by the shallow composition of piecewise linear regions.* LR imposes a piecewise linear composition where regions have a fixed size.

We can also remark on two additional points from Fig. 9: i) we can consistently rank the generalization error for all 3 horizons in the following order, as measured by MAPE: MAPE (F3) < F1<F2<Volve, and ii) the MAPE gap between traditional machine learning models and deep learning also obeys the following order: Volve > F1,F2>F3. These two observations are the subject of Section 5.2.

5.2. The Swiss-cheese effect

Building upon the findings of the preceding section, we introduce and provide evidence for the "Swiss-cheese effect." This concept describes the impact of the spatial sampling density of production data on the generalization error of algorithms. While the existing literature on full-field production forecasts has largely emphasized the role of dataset size in generalization, it has overlooked the influence of spatial sampling density. Our proposition aligns with recent advancements in computer science research that highlight the importance of dataset properties in determining generalization performance (Konz and Mazurowski, 2024).

We argue that $\overline{n_p}$, the historical average of n_p, can be used as a proxy for spatial sampling density in an oil field. We can relate the receptive field of an oil producer with its drainage area, which depends, among other things, on the local permeability. As a result, we should only compare oil fields featuring permeabilities of the same order of magnitude. This is the case in the fields selected in our study. For example, a low permeability field with many wells might still have sparse spatial sampling, as each well has a very limited receptive field (such as in unconventional oil fields). Compartmentalization is another factor that should be considered, when comparing different fields. Perhaps a better proxy for spatial sampling density would be the ratio between $\overline{n_p}$ and the oil-bearing reservoir surface area. However, this information is quite often confidential and would be difficult to come by. Moreover, oil fields are not always drilled following a uniform spatial pattern. Considering these circumstances, our choice of $\overline{n_p}$ as a proxy for density of spatial sampling may be the best option available for the purposes of this analysis.

A manifestation of the Swiss-cheese effect is the observation of an *inverse relationship between the average number of producers, denoted as $\overline{n_p}$, and the generalization error measured by the out-of-sample MAPE from walk-forward validation*. In Fig. 10, we consider five oil fields, namely F1 ($\overline{n_p} = 21$), F2 ($\overline{n_p} = 24$), F3 ($\overline{n_p} = 675$), Volve ($\overline{n_p} = 3$), and an additional oil field referred to as field X ($\overline{n_p} = 930$). Field X, a sandstone conglomerate, was included in this analysis to verify whether it would follow the inverse relationship between $\overline{n_p}$ and MAPE. Field X permeability is of the same order of magnitude as the other fields. The out-of-sample MAPE values for SVR and three forecast horizons are shown in Fig. 10 on the y-axis. We observe a lower generalization error in oil fields featuring more wells, regardless of the forecast horizon. Please note, in F3 and X (oilfields with the longest production history), SVR was trained with *sliding-windows of fixed size, so the observed results are not associated with the training set size*.

Another manifestation of the Swiss-cheese effect is the wider performance gap between kernel-based models (SVR and KR, represented by "o") and neural-based ones (MLP, CNN, LSTM, represented by "*") in fields with few wells. In Fig. 11, we plot on the y-axis the average MAPE. The horizontal axis features the average number of producers, denoted as $\overline{n_p}$. *The use of sparsely sampled spatial production data seems to hurt the generalization performance of neural-based models and widen the performance gap between kernel-based and neural-based models. The combination of small datasets and sparsely sampled spatial information (due to the reduced well inventory) seems to be extremely negative for the generalization ability of neural-based architectures in full-field oil production forecast*, as evidenced by the case of the Volve field. We refer the reader back to the bottom panel of Figs. 8 and 9, which shows the underperformance of neural-based learners for the Volve field. In fact, Fig. 11 conveys three points: i) for all five oil fields, kernel-based learners outperforms deep learning, ii) well inventory has an inverse relationship with the generalization error for both "o" and "*" (the Swiss-cheese effect), and iii) low well inventory increases the MAPE gap between kernel-based and neural-based learners. Liu (2022) argues that subsurface data is often sampled in a sparse and non-representative manner, which introduces a sampling bias. *Our results suggest this sampling bias is reduced in oilfields with hundreds or thousands of wells, where spatial sampling of production information is denser than in fields with few wells.*

5.3. Predictions vs ground truth

The plots featured in Fig. 12 juxtapose the concatenation of predicted and *unseen ground-truth target values* for field F1, F2, F3 and Volve. They serve as evidence that the proposed workflow results in an accurate full-field data-driven proxy model for oil production forecast. We choose to show only the 12-month forecast horizon, as it is the most frequently used forecast horizon for short-term operational planning. For each field, we select the algorithm in Fig. 9 closest to the top-left corner (low MAPE and high R2). Some readers may inquire about the hyperparameter configuration for this algorithm. However, bear in mind we implement the K-best method (discussed in Section 4.5), which simply averages the predictions from the top-ranked K configurations of an algorithm as measured by MAPE and R2 from walk-forward validation. This choice avoids the bad practice of hyperparameter cherry-picking, which is prone to overfitting (Lopez de Prado, 2018). As a result, the final predictions are not associated with a single set of hyperparameters, instead they come from the 5 best hyperparameter configurations (K-best method), presented in Table 4. By using the walk-forward analysis, we are evaluating the predictions from the ensemble of K-best model configurations across the (non-stationary) production history, as opposed to using a single holdout set and a single set of hyperparameters. We emphasize the reported MAPE originates from walk-forward validation *using only out-of-sample sets* and it is a robust estimate of the *generalization error* (Joseph, 2022).

Analysis of model configurations in Table 4 reveals that, except for F1, the best configurations make extensive use of sliding windows. This suggests that full-field oil production is *mostly influenced by recent history* because of the underlying non-stationarity. Shifts in data-generating distribution weaken the relevance of earlier production data (Huang and Wang, 2023). Interestingly, full-field production forecast accuracy seems to be more influenced by spatial sampling density (Swiss-cheese effect) of production data, than by dataset size, as evidenced by the prevalence of sliding windows. In the case of F1, it is worth noting that KR model complexity is associated with the training set size (Murphy, 2012). So, it is fitting that expanding windows were selected for computational reasons. We also remark on the prevalence of Gaussian kernels, due to the *smoothness* of the underlying functions.

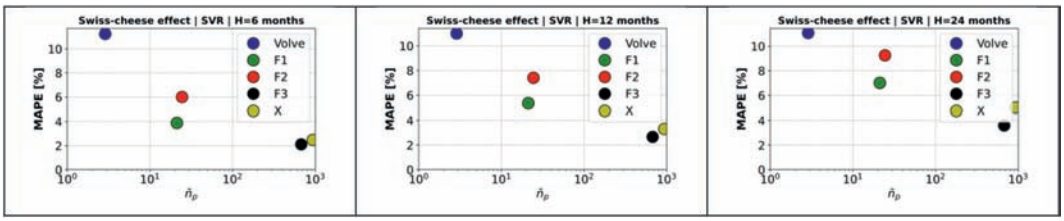


Fig. 10. Manifestation of the Swiss-cheese effect. We observe an inverse relationship between the average number of producers, denoted as $\overline{n_p}$, and the generalization error measured by the out-of-sample MAPE from walk-forward validation. We consider five oil fields (F1, F2, F3, Volve, and an extra one, named X) featuring well inventory of different orders of magnitude. The out-of-sample MAPE for SVR and three forecast horizons is displayed in the y-axis. A low generalization error is observed in heavily drilled oil fields, regardless of the forecast horizon.

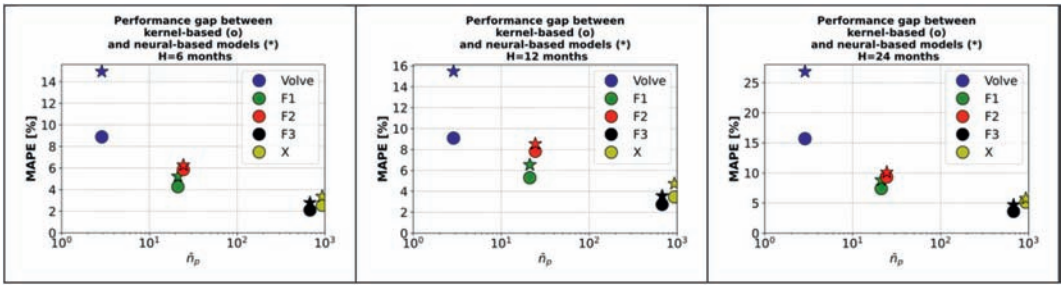


Fig. 11. Two manifestations of the Swiss-cheese effect: 1) The MAPE gap between kernel-based learners (represented by "o") and neural-based models (represented by "*") increases in oilfields with few wells (Volve), 2) The inverse relationship between number of wells and generalization error observed for kernel-based and neural-based learners across all three forecast horizons. In addition, the charts also show that neural-based models do not outperform kernel-based ones.

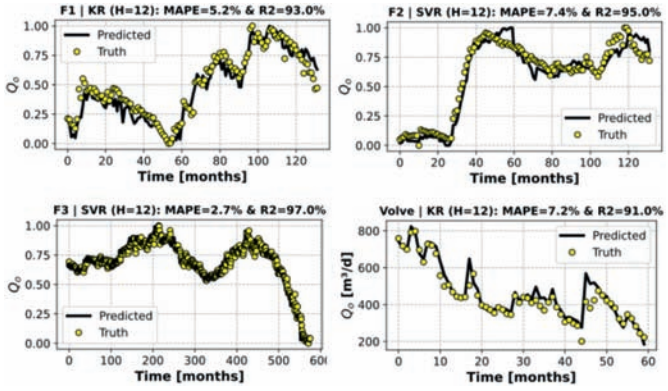


Fig. 12. Juxtaposing the concatenation of predicted and blind ground-truth target values for F1, F2, F3 and Volve (denoted as V) using the best algorithm for each forecast horizon. Predictions were obtained from averaging the K-best model configurations. The final predictions are not associated with a single set of (cherry-picked) hyperparameters, which is prone to overfitting.

Fig. 13 shows that generalization error measured by MAPE diminishes with the stretching of the forecast horizon. This is expected due to the nonstationary nature of full-field production data. For each field, we select the algorithm in Fig. 9 closest to the top-left corner (low MAPE and high R2). The maximum forecast horizon is a function of the desirable generalization error. For practical purposes, a MAPE within 5%–10% range is deemed acceptable, particularly for oil fields with water cut above 10% as detailed in Rodriguez et al. (2020), Nazarenko and Zolotukhin (2021) and Yang et al. (2022). We show in Fig. 13 that such a level of accuracy can be obtained for forecast horizons as long as 24 months. The accuracy degradation does not seem to be data- or model-dependent, judging by our examples. Regardless of the

Table 4
Hyper-parameter configurations resulting from the use of the K-best method for a 12-month forecast horizon.

Field	Algorithm	K-best configurations ($K = 5$)				
		Expanding (E) or sliding (S) window	Kernel	Regularization strength: *alpha* in KR and *C* in SVR	Visual field: *gamma*	Context window size
F1	KR	E	polynomial	1.0	1.0	0
		E	polynomial	0.01	0.1	0
		E	gaussian	0.001	0.1	0
		E	gaussian	0.01	0.1	0
		E	polynomial	0.001	0.1	0
F2	SVR	S	gaussian	1000	0.01	0
		E	gaussian	1000	0.01	0
		S	gaussian	100	0.1	0
		E	gaussian	100	0.1	0
		S	gaussian	1000	0.1	0
F3	SVR	S	gaussian	1	1	6
		S	gaussian	1	1	3
		S	gaussian	10	1	3
		S	gaussian	1	0.1	3
		S	gaussian	1	0.1	6
V	KR	S	polynomial	0.001	0.01	0
		S	gaussian	0.001	0.01	0
		S	polynomial	0.01	0.1	0
		S	gaussian	0.001	0.1	0
		S	gaussian	0.01	0.1	0

underlying function complexity, roughly the same deterioration pattern is observed for F1, F2 and F3. In F1, F2 and F3, the 24-month MAPE is less than 10%. For F3, we quote a MAPE of 3.6% for 24-month forecast horizons over the course of about 50 years. Therefore, despite the degradation seen in a 24-month forecast horizon, the generalization error is still acceptable for most

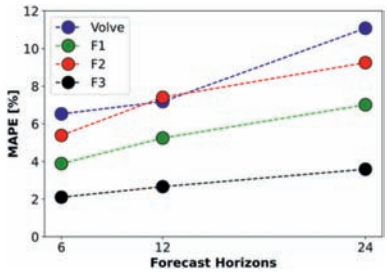


Fig. 13. Generalization error measured by out-of-sample MAPE diminishes with the stretching of the forecast horizon. We used the best algorithm for each forecast horizon. The observed generalization error for all time horizons is acceptable for most practical applications. In particular for F3, we register a MAPE of 3.6% for a 24-month forecast horizon over the course of about 50 years.

practical applications (Rodriguez et al., 2020; Nazarenko and Zolotukhin, 2021).

We remark F1, F2 and F3 behave in a similar manner in Fig. 13, unlike Volve. We believe this might be caused by the fact Volve production profile, displayed in Fig. 1, is monotonic: the flow rate declines with the increase of production time. The other fields show a non-monotonic production profile, which causes more variations in the flowrate behavior between 6 and 12-month forecast horizons. This explains why, in Volve, we see little difference between the 6-month and 12-month MAPE. As a result, the MAPE deterioration, in Volve, between 12 and 24 months seem steeper.

5.4. Application of a data-driven proxy model in oil field management: scenario analysis

This simple example illustrates the application of the full-field data-driven proxy model in scenario analysis in the context of operational oilfield management. Its role as a prescriptive tool for operational decision-making is emphasized. We show how it can be used to quantify production loss associated with the introduction of delays in the operational schedule. Over the last few years of its production history, both the water injection rate and the number of active producers severely declined in F3, as seen in Fig. 1. As a result, a steep decline in oil production ensued. Due to budgetary constraints caused by low Brent prices, producers and injectors undergoing equipment or completion issues had to be permanently shut down. Let us now suppose we have managed to secure the financial resources needed. We are tasked with scheduling well-intervention campaigns with a view to fixing broken pumps, cleaning up plugged wellbores, tubing and pipes, and repairing damaged equipment which will increase the number of active producers. We plan to reopen 120 producers and increase water injection by 2400 m^3/d over the course of the next 12 months. For the sake of simplicity (as our goal is to illustrate the use of the data-driven proxy model), let us assume that over the next 12 months equipment and completion issues leading to wells being shut down will not take place, therefore, if the number of active producers changes by 3 units, it is all due to the re-opening of 3 producers.

The monthly increase in water injection rate, $dq_{wi}[t]$, and the monthly increase in the number of producers, $dn_p[t]$, are limited by the range $100\ m^3/d \leq dq_{wi}[t] \leq 300\ m^3/d$ and $6 \leq dn_p[t] \leq 12$. In practice, these constraints might be caused by operational limitations in the water injection pumps, limitation in the availability of workover crews and/or budgetary constraints. In addition, we note that $\sum_{t=1}^{12} dq_{wi}[t] = 2,400 m^3/d$ and $\sum_{t=1}^{12} dn_p[t] = 120$. Six scenarios, detailed in Table 5 and depicted in the top panels of Fig. 14, are proposed.

With the help of a data-driven proxy model, we would like to quantify the following:

- The gap in cumulative oil production, defined as $\sum_{t=1}^{12} q_o[t]$*30, between each scenario and the Freeze scenario, where the number of producers and the full-field water injection remain unchanged over the next 12 months. This helps us quantify the oil gain from each strategy relative to the choice of keeping the number of active producers and full-field water injection unchanged.
- The gap in cumulative oil production between the ideal scenario, Plan A, against the other scenarios. This allows us to quantify the production loss in case Plan A falls through and we have to rely on the remaining scenarios. We can rank the remaining scenarios based on the production loss with respect to Plan A. That way we can quantify whether Plan B and Plan C are good back-up plans.
- The cumulative oil production loss caused by introducing delays in the well intervention schedule. Which one is less desirable: an early or mid-schedule delay?

The flowrate $q_o[t]$ shown in the bottom panel of Fig. 14 is obtained using the K-best method and the data-driven methodology proposed in this work. In this example, we use the multivariate LR model, which has a low generalization error for F3 and can be easily implemented in an Excel spreadsheet for the sake of reproducibility. Moreover, it is an interpretable model, which is a desirable trait for algorithms used in decision making (Rudin, 2018). We combine 5 LR models with the following configurations (where k stands for the context window size): i) $k = 0$,

Table 5
Six scenarios considered for the re-opening of 120 producers and the increase of water injection by 2400 m^3/d.

Scenario	Strategy	Mathematical Description
Plan A	Operate at maximum capacity (300 m^3/d and 12 wells) until the quota is met.	$dq_{wi}[t] = \begin{cases} 300\ m^3/d, 1 \leq t \leq 8 \\ 0\ m^3/d, 9 \leq t \leq 12 \end{cases}$ $dn_p[t] = \begin{cases} 12, 1 \leq t \leq 10 \\ 0, 11 \leq t \leq 12 \end{cases}$
Plan B	Operate at maximum capacity (300 m^3/d and 12 wells) then at minimum capacity (100 m^3/d and 6 wells) to meet the quota.	$dq_{wi}[t] = \begin{cases} 300\ m^3/d, 1 \leq t \leq 6 \\ 100\ m^3/d, 6 \leq t \leq 12 \end{cases}$ $dn_p[t] = \begin{cases} 12, 1 \leq t \leq 8 \\ 6, 9 \leq t \leq 12 \end{cases}$
Plan C	Operate at the same and intermediate capacity throughout (200 m^3/d and 10 wells) to meet the quota.	$dq_{wi}[t] = 200\ m^3/d$, $1 \leq t \leq 12$ $dn_p[t] = 10, 1 \leq t \leq 12$
Freeze	No re-openings and no increase in water injection.	$dq_{wi}[t] = 0\ m^3/d$, $1 \leq t \leq 12$ $dn_p[t] = 0, 1 \leq t \leq 12$
Delay Start	Plan A with a delay at the beginning. Quota is met, despite the delay.	$dq_{wi}[t] = \begin{cases} 0\ m^3/d, 1 \leq t \leq 4 \\ 300\ m^3/d, 5 \leq t \leq 12 \end{cases}$ $dn_p[t] = \begin{cases} 0, 1 \leq t \leq 2 \\ 12, 3 \leq t \leq 12 \end{cases}$
Delay Middle	Plan A with a delay somewhere after the start. Quota is met, despite the delay.	$dq_{wi}[t] = \begin{cases} 300\ m^3/d, 1 \leq t \leq 4 \\ 0\ m^3/d, 5 \leq t \leq 8 \\ 300\ m^3/d, 9 \leq t \leq 12 \end{cases}$ $dn_p[t] = \begin{cases} 0, 1 \leq t \leq 2 \\ 12, 3 \leq t \leq 12 \end{cases}$

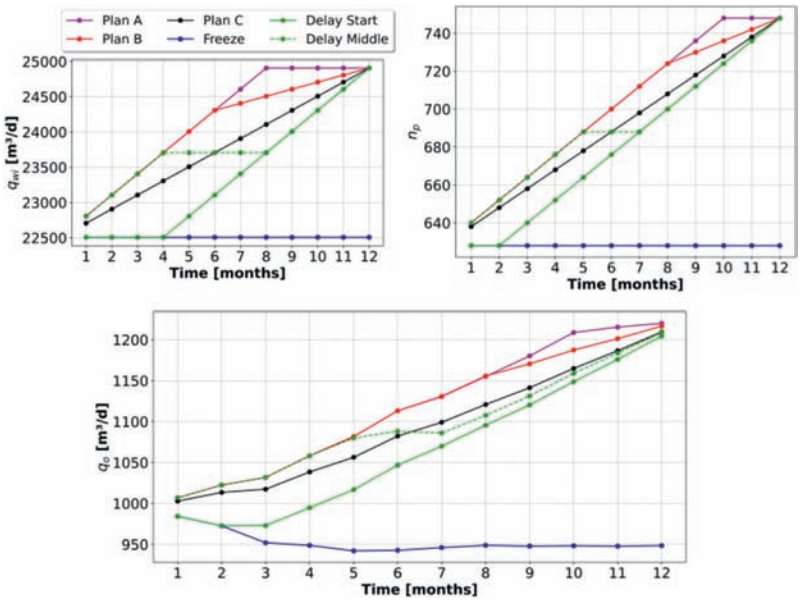


Fig. 14. (top-left) Full-field water injection schedule for each scenario described in Table 5. (top-right) Number of active producers for each scenario. (bottom) Resulting oil flowrate obtained from using the full-field data-driven proxy model to simulate each scenario.

training window = 48 months (a sliding window), ii) $k = 0$, training window = 36 months, iii) $k = 3$, training window = 48 months, iv) $k = 3$, training window = 36 months, and v) $k = 6$, training window = 48 months. Those hyperparameter configurations were obtained using the K-best method outlined in Section 4.5. We linearly extrapolate the prevailing trend in f_w over the last 12 months to predict its next values, as proposed in Kubota and Reinert (2019) and explained in this work in Section 4.4. For $k > 0$, the 2dsew method described in Section 4.4.1 is implemented.

Fig. 15 shows the gap in cumulative oil production between each scenario and Plan A. It quantifies the production loss in case Plan A fails to materialize. In this case, Plan B could be a good replacement with a relatively low production loss. Plan C is not a good back-up plan, as evidenced by its substantial production loss. Fig. 15 also highlights the detrimental role played by delaying the start of the intervention. Table 6 shows the gap in cumulative oil production at the end of 12 months of each scenario against the option to keep the full-field water injection and the number of active producers unchanged (the Freeze scenario). An economic analysis (which is outside the scope of this work) will determine whether the extra oil revenue could offset the higher operational costs. We remark that the predicted oil rate obtained from the data-driven proxy model will serve as input for subsequent cash-flow evaluations.

The following practical takeaways can be derived from the scenario analysis:

- **Inject and re-open as much as possible, as early as possible.** Operating at the limit of the operational capacity seems to be a good choice as evidenced by results in Fig. 15. Plan C, which has no downtime but operates at intermediate capacity, leads to a higher production loss than Plan A with a delay happening in the middle of the campaign, as evidenced by the comparison of curves “Plan A-Plan C” and “Plan A-Delay Middle” in Fig. 15

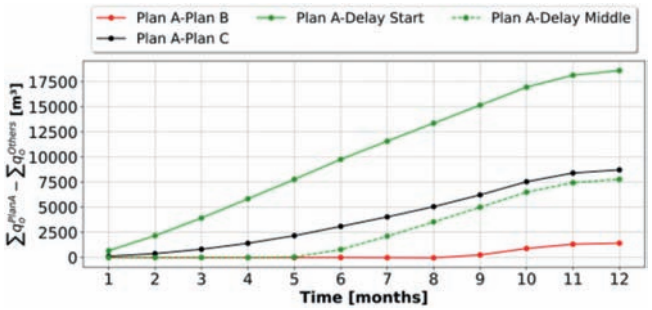


Fig. 15. Production loss between each scenario (except for Freeze) and Plan A. These results highlight the detrimental effect of introducing delays early on during the intervention campaign (“Plan A-Delay Start” curve). Plan B could be a good alternative to Plan A, given the relatively low production loss (“Plan A-Plan B” curve). Plan C, on the other hand, leads to about the same production loss as Plan A with a delay taking place during the intervention campaign (“Plan A-Plan C” and “Plan A-Delay Middle” curves).

Table 6
Gain in cumulative oil production between each scenario and Freeze by the end of 12 months.

Scenario	Cumulative oil gain over “Freeze” [$\times 1000\ m^3$]
Plan A	59
Plan B	58
Plan C	51
Delay Start	41
Delay Middle	52

- **Avoid delaying the start of the intervention.** The biggest production loss comes from delaying the start of Plan A. The production loss associated with falling back on Plan B, Plan C or introducing downtime in the middle of the campaign is less

than 50% of the production loss associated with delaying the start of Plan A, as shown in Fig. 15

5.5. Limitations and future works

The main application for the full-field data-driven proxy model is in scenario analysis as a prescriptive tool to help guide the best course of action for some operational variables. Field operators can make more informed decisions by simulating different schedules of water injection and well re-opening. However, operational schedules can be disrupted by unpredictable events. *Therefore, the proxy model prediction must be conditioned on the effective realization of a given scenario*. A limitation that will be addressed in future work is the inclusion of prediction intervals. Future work will also investigate the efficacy of transfer learning for full-field oil production forecasting. Given the relatively limited datasets inherent in this domain, it is worth exploring how both traditional and deep learning models can benefit from pre-trained knowledge to improve predictive accuracy. Can a database of historical production data from similar oil fields be used to pre-train the weights of earlier layers, or to extract similarity features via kernels, whereas top layers would be fine-tuned to the target field? And how do we define similarity among oil fields in terms of historical production data instead of commonly used static data (petrophysical and geological variables) to preserve the reproducibility and scalability of the method to other fields?

We emphasize that this study is set in the context of an oilfield under waterflood. For a different oil recovery strategy (polymer, chemical or steam injection), the proposed methodological choices for model training, validation and ensembling can still be applied. However, a new problem-specific input space might have to be defined, and appropriate historical data (at least 36 months under the new recovery strategy) must be collected for model training.

6. Conclusions

We propose a new data-driven method for forecasting full-field oil production over short and medium-term horizons. The workflow can be used as a prescriptive tool to guide operational and business decisions, such as designing waterflooding schedules or performing cash-flow evaluations in the context of oilfield management. The data-driven full-field proxy model, a technological contribution, enables engineers to quickly simulate full-field oil production under different schedules of water injection, infill drilling and well reopening. We present an example of the use of a full-field proxy model to quantify the production loss associated with delays in the operational schedule and the resulting oil gain from different scenarios.

At a fundamental level, our findings suggest traditional machine learning algorithms, such as kernel-based models, consistently outperform neural-based architectures in the context of full-field production forecasts. We do not claim that deep learning models are unsuitable for this problem. In fact, we show they are. However, a systematic comparison over four oil fields and three forecast horizons reveals that a combination of low dimensionality, limited and sparse datasets, and the use of physics to guide the selection of input variables seem to favor the faster, shallow, and more interpretable traditional machine learning models over deep learning, despite the unjustified and widespread preference of related works for the latter. This suggests the Green AI paradigm should be considered by engineers and researchers applying machine learning to petroleum engineering problems. In addition, we provide evidence for an inverse relationship between the generalization error (measured by out-of-sample MAPE from walk-forward validation) and spatial-sampling density of production data (measured by the number of wells) in the context of full-field predictions, even when controlled for training set size. We call it the Swiss-cheese effect. This relationship amplifies the generalization error gap between traditional machine learning and deep learning models in oil fields with few active wells. This finding suggests dataset spatial-sampling density seems to be an important factor in algorithmic generalization in the context of full-field production forecasts, more so than dataset size and choice of algorithm.

CRediT authorship contribution statement

Leonardo Kubota: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Rafael dos Santos Gioria:** Writing – review & editing, Validation, Supervision, Methodology, Investigation.

Declaration of competing interest

The authors declare that there is no conflict of interest.

Acknowledgments

The authors are grateful to fellow engineers Mr. R. Araujo Frota and Mr. C. Keidel for their invaluable comments and suggestions.

References

Aggarwal, C.C., 2018. Neural Networks and Deep Learning. Springer.
Alali, Z.A.A.H., Horne, R., 2023. Probabilistic well production forecasting in volve field using temporal fusion transformer deep learning models. In: SPE Gas & Oil Technology Showcase and Conference. https://doi.org/10.2118/214133-MS.
Alimohammadi, H., Rahmanifard, H., Chen, N., 2020. Multivariate time series modelling approach for production forecasting in unconventional resources. In: SPE Annual Technical Conference and Exhibition. https://doi.org/10.2118/201571-ms. SPE-201571-MS.
Aliyuda, K., Howell, J., Humphrey, E., 2020. Impact of geological variables in controlling oil-reservoir performance: an insight from a machine-learning technique. SPE Reservoir Eval. Eng. 23, 1314–1327. https://doi.org/10.2118/201196-pa.
Arik, S.Ö., Pfister, T., 2021. TabNet: attentive interpretable tabular learning. Proc. AAAI Conf. Artif. Intell. 35 (8), 6679–6687. https://doi.org/10.1609/aaai.v35i8.16826.
Arps, J., 1945. Analysis of decline curves. Trans. AIME 160 (1), 228–247.
Atadeger, A., Onur, M., Sheth, S., Banerjee, R., 2023. Deep learning-based and kernel-based proxy models for nonlinearly constrained life-cycle production optimization. In: SPE Reservoir Characterisation and Simulation Conference and Exhibition. https://doi.org/10.2118/212690-MS. Abu Dhabi, UAE.
Awad, M., Khanna, R., 2015. Support vector regression. In: Awad, M., Khanna, R. (Eds.), Efficient Learning Machines. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4302-5990-9_4.
Bahaloo, S., Mehrizadeh, M., Najafi-Marghmaleki, A., 2023. Review of application of artificial intelligence techniques in petroleum operations. Petrol. Res. 8 (2), 167–182. https://doi.org/10.1016/j.ptlrs.2022.07.002.
Bengio, Y., 2011. Deep learning of representations for unsupervised and transfer learning. In: Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27 (UTLW'11). JMLR.org, pp. 17–37.
Cao, C., Jia, P., Cheng, L., Jin, Q., Qi, S., 2022. A review on application of data-driven models in hydrocarbon production forecast. J. Petrol. Sci. Eng. 212. https://doi.org/10.1016/j.petrol.2022.110296.
Cerqueira, V., Torgo, L., Mozetič, I., 2019. Evaluating time series forecasting models: an empirical study on performance estimation methods. Mach. Learn. 109, 1997–2028. https://doi.org/10.1007/s10994-020-05910-7.
Chan, E.P., 2008. Quantitative Trading: How to Build your Own Algorithmic Trading Business, first ed. Wiley Publishing.
Chen, Z., Cao, Y., Gu, Q., Zhang, T., 2020. A generalized neural tangent kernel analysis for two-layer neural networks. Adv. Neural Inf. Process. Syst. 33, 13363–13373.
Chollet, F., 2017. Deep Learning with Python. First Edition, Manning.
Cruz, E., Pyrcz, M.J., 2024. Multi-horizon well performance forecasting with temporal fusion transformers. Results Eng. https://doi.org/10.2139/ssrn.4403939.
Dake, L.P., 1978. Fundamentals of Reservoir Engineering, first ed. Elsevier.
Davtyan, A., Rodin, A., Muchnik, I.B., Romashkin, A., 2020. Oil production forecast models based on sliding window regression. J. Petrol. Sci. Eng. 195, 107916. https://doi.org/10.1016/j.petrol.2020.107916.

Domingos, P., 1999. The role of Occam's razor in knowledge discovery. Data Min. Knowl. Discov. 3, 409–425. https://doi.org/10.1023/A:1009868929893.
Equinor, 2018. Disclosing all Volve Data [WWW document], 6.28.21. https://www.equinor.com/en/news/14jun2018-disclosing-volve-data.html.
Galicia, A., Talavera-Llames, R., Troncoso, A., Koprinska, I., Martínez-Álvarez, F., 2019. Multi-step forecasting for big data time series based on ensemble learning. Knowl. Base Syst. https://doi.org/10.1016/j.knosys.2018.10.009.
Géron, A., 2017. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, first ed. O'Reilly Publishing.
Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, Cambridge.
Grinsztajn, L., Oyallon, E., Varoquaux, G., 2022. Why do tree-based models still outperform deep learning on tabular data? ArXiv, abs/2207.08815. https://doi.org/10.48550/arXiv.2207.08815.
Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning, second ed. Springer.
Haykin, S., 2009. Neural Networks and Learning Machines. Prentice Hall.
Herrmann, M., Lange, F.J.D., Eggensperger, K., Casalicchio, G., Wever, M., Feurer, M., Rugamer, D., Hullermeier, E., Boulesteix, A.L., Bischl, B., 2024. Rethinking Empirical Research in Machine Learning: Addressing Epistemic and Methodological Challenges of Experimentation. Proceedings of the 41 St International Conference on Machine Learning. PMLR, Vienna, Austria, p. 235.
Holanda, R.W.D., Gildin, E., Jensen, J.L., Lake, L.W., Kabir, C.S., 2018. A state-of-the art literature review on capacitance resistance models for reservoir characterization and performance forecasting. Energies 11 (12), 3368. https://doi.org/10.3390/en11123368.
Huang, C., Wang, K., 2023. A stability principle for learning under non-stationarity. ArXiv, abs/2310.18304. https://doi.org/10.48550/arXiv.2310.18304.
Hunyinbo, S., Azom, P.N., Ben-Zvi, A., Leung, J.Y., 2022. A machine learning approach to real-time uncertainty assessment of SAGD forecasts. SPE J. https://doi.org/10.2118/208962-pa.
James, G.M., Witten, D.M., Hastie, T.J., Tibshirani, R., 2021. An Introduction to Statistical Learning. Springer Texts in Statistics.
Joseph, M., 2022. Modern Time Series Forecasting with Python. Packt Publishing.
Kelleher, J.D., 2019. Deep Learning, first ed. MIT Press.
Khanal, A., Khoshghadam, M., Lee, W.J., Nikolaou, M., 2017. New forecasting method for liquid rich shale gas condensate reservoirs with data driven approach using principal component analysis. J. Nat. Gas Sci. Eng. 38, 621–637. https://doi.org/10.1016/J.JNGSE.2017.01.014.
Kim, S., Kim, H., 2016. A new metric of absolute percentage error for intermittent demand forecasts. Int. J. Forecast. 32 (3), 669–679. https://doi.org/10.1016/j.ijforecast.2015.12.003.
Konz, N., Mazurowski, M.A., 2024. The effect of intrinsic dataset properties on generalization: unraveling learning differences between natural and medical images. arXiv:2401, 08865v2. https://doi.org/10.48550/arXiv.2401.08865.
Koren, Y., 2009. The bellkor solution to the netflix grand prize. Netflix Prize Document. 81, 1–10.
Kubota, L.K., Gioria, R.S., 2022. Data-driven technique estimates skin factor and average pressure during oil-flowing periods. J. Petrol. Sci. Eng. https://doi.org/10.1016/j.petrol.2022.111061.
Kubota, L.K., 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. https://doi.org/10.2118/196152-ms.
Kubota, L.K., Souto, F., 2019. Application of machine learning to oil production forecast under uncertainties-the linear model. In: Offshore Technology Conference Brasil. https://doi.org/10.4043/29883-ms.
Kumar, I., Tripathi, B.K., Singh, A., 2023. Attention-based LSTM network-assisted time series forecasting models for petroleum production. Eng. Appl. Artif. Intell. https://doi.org/10.1016/j.engappai.2023.106440.
Lakshminarayanan, B., Pritzel, A., Blundell, C., 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems, pp. 6402–6413. https://arxiv.org/pdf/1612.01474.pdf.
Lara-Benítez, P., Carranza-García, M., Santos, J.C., 2021. An experimental review on deep learning architectures for time series forecasting. Int. J. Neural Syst., 2130001 https://doi.org/10.1142/S0129065721300011.
Latrach, A., Malki, M.L., Morales, M., Mehana, M., Rabiei, M., 2023. A critical review of physics-informed machine learning applications in subsurface energy systems. ArXiv, abs/2308.04457. https://arxiv.org/pdf/2308.04457.pdf.
Lee, T.L., Hellan, S.P., Ericsson, L., Crowley, E.J., Storkey, A.J., 2024. Hyperparameter selection in continual learning. arXiv preprint arXiv: 2404.06466. https://arxiv.org/pdf/2404.06466.pdf.
Liang, Y., Zhao, P., 2019. A machine learning analysis based on big data for eagle ford shale formation. In: SPE Annual Technical Conference and Exhibition. https://doi.org/10.2118/196158-MS. SPE-196158-MS.
Liang, B., Liu, J., You, J., Jia, J., Pan, Y., Jeong, H., 2023. Hydrocarbon production dynamics forecasting using machine learning: a state-of-the-art review. Fuel. https://doi.org/10.1016/j.fuel.2022.127067.
Liao, L., Zeng, Y., Liang, Y., Zhang, H., 2020. Data mining: a novel strategy for production forecast in tight hydrocarbon resource in Canada by random forest analysis. In: International Petroleum Technology Conference, IPTC-20344-MS. https://doi.org/10.2523/iptc-20344-ms.
Lim, B., Zohren, S., 2021. Time-series forecasting with deep learning: a survey. Phil. Trans. R. Soc. A. https://doi.org/10.1098/rsta.2020.0209.
Liu, W., 2022. Integration of Spatial Data Context into Machine Learning Models (Doctoral Dissertation). The University of Texas at Austin.
Liu, W., Chen, Z., Hu, Y., Xu, L., 2022. A systematic machine learning method for reservoir identification and production prediction. Pet. Sci. https://doi.org/10.1016/j.petsci.2022.09.002.
Lolon, E., Hamidieh, K., Weijers, L., Mayerhofer, M., Melcher, H., Oduba, O., 2016. Evaluating the relationship between well parameters and production using multivariate statistical models: a middle bakken and three forks case history. In: SPE Hydraulic Fracturing Technology Conference and Exhibition, pp. 9–11. https://doi.org/10.2118/179171-MS. SPE-179171-MS.
Lones, M.A., 2021. How to avoid machine learning pitfalls: a guide for academic researchers. ArXiv, abs/2108.02497. https://doi.org/10.48550/arXiv.2108.02497.
Lopez de Prado, M., 2018. Advances in Financial Machine Learning, 1st. Wiley Publishing.
Makhotin, I., Orlov, D.M., Koroteev, D.A., 2022. Machine learning to rate and predict the efficiency of waterflooding for oil production. Energies. https://doi.org/10.3390/en15031199.
Makridakis, S., Anderson, R., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, H.J., Parzen, E., Winkler, R., 1982. The accuracy of extrapolation (time series) methods: results of a forecasting competition. J. Forecast. 1, 111–153. https://doi.org/10.1002/FOR.3980010202.
Murphy, K.P., 2012. Machine Learning - a Probabilistic Perspective. Adaptive Computation and Machine Learning Series. MIT Press.
Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., Sutskever, I., 2019. Deep double descent: where bigger models and more data hurt. J. Stat. Mech. Theor. Exp. https://doi.org/10.1088/1742-5468%2Fac3a74.
Nazarenko, M.Y., Zolotukhin, A.B., 2021. Application of machine learning algorithms and integrated production modelling to improve accuracy of liquid production rate measurements using multiphase flow meters. In: Abu Dhabi International Petroleum Exhibition & Conference. https://doi.org/10.2118/207674-ms.
Ng, C.S., Jahanbani Ghahfarokhi, A., Nait Amar, M., 2022. Well production forecast in volve field: application of rigorous machine learning techniques and metaheuristic algorithm. J. Petrol. Sci. Eng. https://doi.org/10.1016/j.petrol.2021.109468.
Ning, Y., Kazemi, H., Tahmasebi, P., 2022. A comparative machine learning study for time series oil production forecasting: ARIMA, LSTM, and prophet. Comput. Geosci. 164. https://doi.org/10.1016/j.cageo.2022.105126.
Noshi, C.I., 2019. An intelligent data driven approach for production prediction. In: Offshore Technology Conference. OTC-29243-MS. https://doi.org/10.2118/1019-0062-jpt.
Pardo, R., 2012. The Evaluation and Optimization of Trading Strategies, second ed. Wiley Publishing. https://doi.org/10.1002/9781119196969.
Prince, S.J.D., 2023. Understanding Deep Learning. MIT Press.
Rodriguez, M., Huter, A.C., Berrios, V., Bitopoulus, C., Pitrelli, G., 2020. Nonlinear regression algorithm for uncertainty analysis in production data measurements, Estancia Cholita, Santa Cruz, Argentina. In: SPE Latin American and Caribbean Petroleum Engineering Conference. https://doi.org/10.2118/199113-MS.
Rudin, C., 2018. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat. Mach. Intell. 1, 206–215. https://doi.org/10.1038/S42256-019-0048-X.
Sagheer, A.E., Kotb, M., 2019. Time series forecasting of petroleum production using deep LSTM recurrent networks. Neurocomputing 323, 203–213. https://doi.org/10.1016/j.neucom.2018.09.082.
Sapankevych, N.I., Sankar, R., 2009. Time series prediction using support vector machines: a survey. IEEE Comput. Intell. Mag. 4 (2), 24–38. https://doi.org/10.1109/MCI.2009.932254.
Schmidt-Hieber, J., 2020. Nonparametric regression using deep neural networks with ReLU activation function. Ann. Stat. 48 (4), 1875–1897. https://doi.org/10.1214/19-AOS1875.
Schneider, L., Bischl, B., Feurer, M., 2025. Overtuning in Hyperparameter Optimization. https://doi.org/10.48550/arXiv.2506.19540. AUTOML 2025.
Schwartz, R., Dodge, J., Smith, N.A., Etzioni, O., 2020. Green AI. Commun. ACM 63 (12), 54–63. https://doi.org/10.1145/3381831.
Sheikhoushaghi, A., Gharaei, N.Y., Nikoofard, A., 2021. Application of rough neural network to forecast oil production rate of an oil field in a comparative study. J. Petrol. Sci. Eng. https://doi.org/10.1016/j.petrol.2021.109935.
Sircar, A., Yadav, K., Rayavarapu, K., Bist, N., Oza, H., 2021. Application of machine learning and artificial intelligence in oil and gas industry. Petrol. Res. 6 (4), 379–391. https://doi.org/10.1016/j.ptlrs.2021.05.009.
Tewari, S., Dwivedi, U.D., Gandhi, R., Shiblee, M., 2019. Assessment of big data analytics based ensemble estimator module for the real-time prediction of reservoir recovery factor. In: SPE Middle East Oil and Gas Show and Conference. SPE-194996-MS. https://doi.org/10.2118/194996-MS.
Vaswani, A., et al., 2017. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, pp. 6000–6010. https://doi.org/10.48550/arXiv.1706.03762.
Wang, H., Chen, S., 2023. Insights into the application of machine learning in reservoir engineering: current developments and future trends. Energies 16 (3), 1392. https://doi.org/10.3390/en16031392.
Wang, H., Mu, L., Shi, F., Dou, H., 2020. Production prediction at ultra-high water cut stage via recurrent neural network. Petrol. Explor. Dev. 47, 1084–1090. https://doi.org/10.1016/s1876-3804%2820%2960119-7.
Werneck, R.O., Prates, R., Moura, R., Gonçalves, M.M., Castro, M., Soriano-Vargas, A., Junior, P.R.M., Hossain, M.M., Zampieri, M.F., Ferreira, A., Davolio, A.,

Schiozer, D., Rocha, A., 2022. Data-driven deep-learning forecasting for oil production and pressure. J. Petrol. Sci. Eng. 210. https://doi.org/10.1016/j.petrol.2021.109937.
Wilson, A.G., 2025. Deep learning is not So mysterious or different. arXiv preprint arXiv:2503.02113. https://doi.org/10.48550/arXiv.2503.02113.
Wolberg, J.R., 2000. Expert Trading Systems: Modeling Financial Markets with Kernel Regression, 1st. ed. John Wiley & Sons, Inc., USA.
Xu, J., Zhou, W., Fu, Z., Zhou, H., Li, L., 2021. A survey on green deep learning. arXiv preprint arXiv:2111.05193. https://doi.org/10.48550/arXiv.2111.05193.
Xue, L., Liu, Y., Xiong, Y., Liu, Y., Cui, X., Lei, G., 2021. A data-driven shale gas production forecasting method based on the multi-objective random forest regression. J. Petrol. Sci. Eng. 196. https://doi.org/10.1016/j.petrol.2020.107801.
Yang, Y., Ha, W., Zhang, C., Liu, M., Zhang, X., Wang, D., 2022. Measurement of high-water-content oil-water two-phase flow by electromagnetic flowmeter and differential pressure based on phase-isolation. Flow Meas. Instrum. 84. https://doi.org/10.1016/j.flowmeasinst.2022.102142.
Yuan, Z., Huang, H., Jiang, Y., Li, J., 2021. Hybrid deep neural networks for reservoir production prediction. J. Petrol. Sci. Eng. 197. https://doi.org/10.1016/j.petrol.2020.108111.
Zainab, H., Horne, R., 2023. Meta learning using deep N-BEATS model for production forecasting with limited history. In: Gas & Oil Technology Showcase and Conference. UAE, Dubai. https://doi.org/10.2118/214214-MS.
Zeng, A., Chen, M., Zhang, L., Xu, Q., 2022. Are transformers effective for time series forecasting? arXiv preprint arXiv:2205.13504. https://doi.org/10.48550/arXiv.2205.13504.
Zubarev, D.I., 2009. Pros and cons of applying proxy-models as a substitute for full reservoir simulations. In: SPE Annual Technical Conference and Exhibition. SPE-124815-MS, New Orleans, Louisiana, USA. https://doi.org/10.2118/124815-MS.