

## Well production forecast in Volve field: Application of rigorous machine learning techniques and metaheuristic algorithm

Cuthbert Shang Wui Ng<sup>a,\*</sup>, Ashkan Jahanbani Ghahfarokhi<sup>a</sup>, Menad Nait Amar<sup>b</sup>

<sup>a</sup> Department of Geoscience and Petroleum, Norwegian University of Science and Technology, Trondheim, Norway

<sup>b</sup> Département Etudes Thermodynamiques, Division Laboratoires, Sonatrach, Boumerdes, Algeria



### ARTICLE INFO

#### Keywords:

Production prediction  
Data-driven techniques  
Machine learning  
Support vector regression  
Neural networks  
Particle swarm optimization

### ABSTRACT

Developing a model that can accurately predict the hydrocarbon production by only employing the conventional mathematical approaches can be very challenging. This is because these methods require some underlying assumptions or simplifications, which might cause the respective model to be unable to capture the actual physical behavior of fluid flow in the subsurface. However, data-driven methods have provided a solution to this challenge. With the aid of machine learning (ML) techniques, data-driven models can be established to help forecasting the hydrocarbon production within acceptable range of accuracy. In this paper, different ML techniques have been implemented to build the models that predict the oil production of a well in Volve field. These techniques comprise support vector regression (SVR), feedforward neural network (FNN), and recurrent neural network (RNN). Particle swarm optimization (PSO) has also been integrated in training the SVR and FNN. These developed models can practically estimate the oil production of a well in Volve field as a function of time and other parameters: on stream hours, average downhole pressure, average downhole temperature, average choke size percentage, average wellhead pressure, average wellhead temperature, daily gas production, and daily water production. All these models illustrate splendid training, validation, and testing results with correlation coefficients,  $R^2$  being greater than 0.98. Moreover, these models show good predictive performance with  $R^2$  exceeding 0.94. Comparative analysis is also done to evaluate the predictability of these models.

### 1. Introduction

Accurate prediction of hydrocarbon production is necessary to ensure that the petroleum engineers have useful information to perform economic evaluation and optimization routines. Nonetheless, achieving high accuracy in production prediction is very challenging due to the sophistication of the subsurface conditions. Furthermore, the non-linearity between hydrocarbon production and any relevant petrophysical parameter often adds complexity to the modeling of production forecasting. Despite having successfully modeled the relationship between hydrocarbon production and any of these petrophysical parameters, lack of these data in real life raises additional difficulty (Ma and Liu, 2018). Therefore, developing a reliable predictive model of hydrocarbon production based upon available data has been one of the research interests in petroleum domain for few decades. This is because with such models, petroleum engineers will have a more profound understanding of the reservoir performance to solve any reservoir management-related issue.

One of the classical approaches in forecasting the hydrocarbon production is the decline curve analysis (DCA). This method was first developed by Arps (1945) and its application has been extended in the oil and gas industry (Fanchi et al., 2013; Hong et al., 2019; Jochen and Spivey, 1996). Due to its simple implementation, it is widely used as only historical production data is required. However, this illustrates that decline curve model is not robust as other important data, such as bottomhole pressure, wellhead pressure, choke size, etc. that affect the production are not considered. Being empirical in nature, it is also insufficient to fully reflect the physics of the fluid flow in subsurface and might either underestimate or overestimate the production estimate (Mohaghegh, 2017, 2020). Apart from DCA, numerical reservoir simulation (NRS) is another alternative applied to forecast the hydrocarbon production. Nonetheless, the predictive performance of the NRS is highly dependent on how the history matching (HM), which is a laborious task, is done (Liu et al., 2019). Additionally, NRS requires different data, including geological data, fluid properties, location of wells, etc. As new data is available in real time, the simulation model needs to be

\* Corresponding author.

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

updated via HM to have a higher accuracy in production forecasting. Thus, the shortcomings of these methods are evident.

With the advancement of computing technology and data analytics, data-driven modeling has become another solution to hydrocarbon production forecasting. This method is not only simple to be implemented but can also capture the complex relationship between input and output of datasets provided. Data or measurement from real field is a representation of the “physics” that deciphers the “actual system” in the reservoir (Mohaghegh, 2017, 2020). Therefore, underlying assumption is not needed to simplify the physics in building a data-driven model that forecasts the production. In this context, the data-driven models learn the relationship between hydrocarbon production and other data obtained from real field through machine learning (ML) techniques: artificial neural network (ANN), support vector regression (SVR), etc. In recent years, the coupling of these ML methods with data analytics has achieved a great milestone in different domains of reservoir engineering, such as prediction of bottomhole pressure (Nait Amar et al., 2018; Nait Amar and Zeraibi, 2020), prediction of essential parameters needed in CO<sub>2</sub>-EOR (Nait Amar et al., 2020a; Nait Amar and Jahanbani Ghahfarokhi, 2020; Nait Amar and Zeraibi, 2018), optimization in water alternating CO<sub>2</sub>-EOR (Nait Amar et al., 2020b; Nait Amar and Zeraibi, 2019), waterflooding optimization (Ng et al., 2021a, 2021b), and forecast of hydrocarbon production (Aydin, 2015; Cao et al., 2016; Elmabrouk et al., 2014; Frausto-Solís et al., 2015; Zanjani et al., 2020).

Apart from these, coupling the application of metaheuristic algorithms with the ML techniques in data-driven modeling is another intriguing research domain. Metaheuristic algorithms are generally nature-inspired and derivative-free. Hence, their implementation is not only considered to be simplistic, but also powerful in terms of convergence to the global optimum (Ezugwu et al., 2020). Their employment in data-driven modeling has exhibited positive results as discussed by several literatures (Akande et al., 2017; Han and Bian, 2018; Nait Amar et al., 2018; Nait Amar and Zeraibi, 2020; Panja et al., 2018). On the other hand, a more advanced ANN technique: RNN, which Li et al. (2019) termed as deep learning, could also efficiently simulate the reservoir behaviors. Alakeely and Horne (2020) successfully implemented these deep learning methods to perform the estimation of bottomhole pressure. Moreover, Calvette et al. (2020) illustrated that RNN could be implemented to approximate the smart well production based upon a synthetic case study. The robustness of RNN was further demonstrated when it could also be coupled with ensemble Kalman filter (EnKF) to predict production of a waterflooded synthetic model (Bao et al., 2020). Besides, several literatures (Lee et al., 2019; Zhan et al., 2020) also highlighted the usefulness of RNN in forecasting the production from unconventional reservoirs. Thus, the use of ML in reservoir engineering shows a great potential.

Besides reservoir engineering, there are some contemporary works done on the employment of ML in the domains of production and drilling engineering. About production engineering, Mamudu et al. (2020) illustrated a dynamic risk analysis of petroleum production by developing ANN based on different geological realizations to help predicting the production. Bayesian network was also built to evaluate the risk of production. Moreover, Kondori et al. (2021) successfully established the connectionist models to evaluate the recovery performance of low water salinity injection. The connectionist models were developed with least squares support vector machine coupled with simulated annealing algorithm and adaptive network-based fuzzy inference system. Syed et al. (2020) also discussed how ML methods could be applied to optimize and conduct preventive maintenance on the artificial lift system. There are also other insightful literatures (Crnogorac et al., 2020; Khamis et al., 2020; Lin et al., 2020; Zhong et al., 2020) touching upon the implementation of ML in the production domain. For drilling engineering, Adedigba et al. (2018) conducted a risk assessment of offshore drilling operations with the help of data-driven model that is the Bayesian Tree Augmented Naïve Bayes algorithm. Fundamentally, this model could forecast the probability of kick that was updated in real time and utilized

to model the time dependent blowout risk. Additionally, Ozbayoglu et al. (2021) demonstrated the development of ANN by using the experimental data gathered and employed this ANN to optimize flow rate and speed of pipe rotation under effective cutting transport. Furthermore, there are other interesting contemporary literatures (Alali et al., 2021; Barbosa et al., 2019; Gan et al., 2020; Muojeke et al., 2020; Olukoga and Feng, 2021) about the application of ML in the drilling aspect.

This paper aims at applying different ML methods to develop data-driven models for the forecast of hydrocarbon production. Regarding the dataset, it is from a real-life well in Volve field (one of the latest databases released by Equinor (2020) to the public for research purposes) used to build the models. The details regarding the data will follow later. A portion of the data from the well is employed to develop the models whereas the remaining part of the data is used as the blind case to further verify the predictive performance of the models. About the ML methods, we first consider applying SVR and FNN. Also, we have employed particle swarm optimization (PSO) in the training of FNN and SVR models. Since hydrocarbon production is an example of time series data, RNN approach is also considered as it has been proven useful to forecast time series data (Alom et al., 2019; Connor et al., 1994; Zhang and Xiao, 2000). In terms of RNN modeling in this paper, three different types of RNNs: the simple RNN, Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU), are developed. In total, seven data-driven models, which comprise FNN with backpropagation algorithm (FNN-BP), FNN trained with PSO (FNN-PSO), SVR tuned with trial-and-error approach (SVR-TE), hybrid model of SVR and PSO (SVR-PSO), simple RNN, LSTM, and GRU, have been established for comparative analysis on their respective predictive capabilities.

The paper is followed by some brief explanations regarding the theory of SVR, FNN, PSO, RNN, LSTM, and GRU. The next section discusses the methodology involved and explains how the available data is pre-processed and utilized in developing these models. The procedures in the development of the models are also expounded. The results and discussion will then follow prior to proceeding to conclusions that summarize the main findings of this work.

## 2. Theory

### 2.1. Support vector regression (SVR)

SVR is a subset of support vector machine that is an advanced supervised machine learning method that uses data for regression analysis, which was proposed by Vapnik (1995). It develops a function that can estimate the relationship between the desired outputs  $y = \{y_1, y_2, \dots, y_k\}$  defined on  $\mathbb{R}$ , and inputs  $x = \{x_1, x_2, \dots, x_k\}$  in which  $x_j \in \mathbb{R}$  and  $k$  is the number of data points. The function can be formulated as shown below:

$$f(x) = w \cdot \Psi(x) + b \quad (1)$$

$\Psi(x)$  refers to the function that maps the input space vector  $x$  into a high dimensional feature space to enable the initial non-linear problem to be expressed and conveniently solved as a linear regression function.  $w$  denotes the weight vector whereas  $b$  is the bias term. To determine  $w$  and  $b$ , the minimization of the following regularized risk function should be done as recommended by Vapnik (1995):

$$E(C) = \frac{C}{k} \sum_{j=1}^k L(f(x_j) - y_j) + \frac{1}{2} \|w\|^2 \quad (2)$$

In equation (2), the first term indicates the empirical error, and the second term means the degree of flatness of the function. Pertaining to this, the constant  $C$  acts as the penalty parameter that governs the trade-off between the complexity of the model and the empirical error. To solve for the empirical error, Vapnik (1995) suggested to use  $\varepsilon$ -insensitive loss function which is represented below:

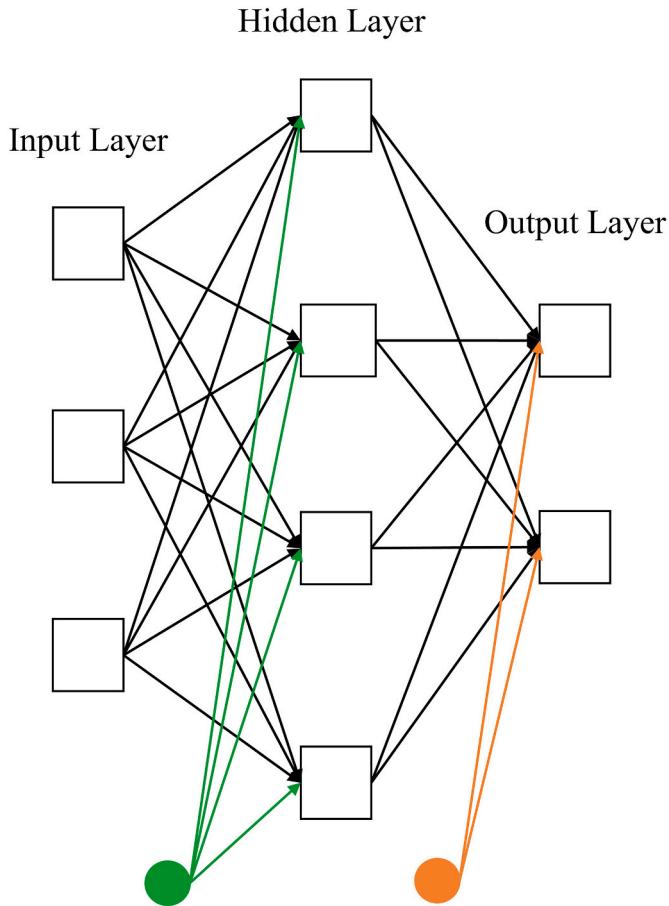


Fig. 1. The structure of an FNN model.

$$L(f(x) - y) = \begin{cases} 0, & \text{if } |f(x) - y| \leq \varepsilon \\ |f(x) - y| - \varepsilon, & \text{otherwise} \end{cases} \quad (3)$$

$\varepsilon$  is the error tolerance. Thereafter, the parameters can be optimized in the following equation through the formulation of the constrained optimization problem (Forrester et al., 2008):

$$\min C \sum_{j=1}^k (\xi_j^- + \xi_j^+) + \frac{1}{2} \|w\|^2$$

$$\text{subject to } \begin{cases} y_j - (w \cdot \Psi(x) + b) \leq \varepsilon + \xi_j^+ \\ (w \cdot \Psi(x) + b) - y_j \leq \varepsilon + \xi_j^- \\ \xi_j^-, \xi_j^+ \geq 0, j = 1, 2, \dots, k \end{cases} \quad (4)$$

$\xi_j^-$  and  $\xi_j^+$  are non-negative slack variables. To solve this constrained optimization problem, the optimization function can be transformed into dual space by using Lagrange multipliers (Shawe-Taylor and Cristianini, 2004). The obtained solution is shown below:

$$f(x) = \sum_{j=1}^k (\alpha_j - \alpha_j^*) K(x_j, x_m) + b \quad (5)$$

In equation (5),  $\alpha_j$  and  $\alpha_j^*$  are Lagrange multipliers which must fulfill the constraints of  $0 \leq j$  and  $j \leq C$  whereas the term  $K(x_j, x_m)$  denotes the kernel function. In the literature (Forrester et al., 2008), there are different kernel functions available, but the commonly used ones include, but not limited to, radial basis function (RBF), polynomial function, and Gaussian function as illustrated in several literatures

(Chiroma et al., 2014; Kavzoglu and Colkesen, 2009; Qu and Zhang, 2016). In this paper, RBF is used as the kernel function and defined as shown below:

$$K(x_j, x_m) = \exp(-\gamma \|x_j - x_m\|) \quad (6)$$

where  $\gamma$  is the kernel parameter. The performance and accuracy of SVR is heavily influenced by the combination of  $\gamma$ ,  $C$ , and  $\varepsilon$ . Therefore, implementing metaheuristic algorithms to optimize these parameters can be done to achieve an ideal performance of SVR. In addition, this can also overcome any inconvenience due to the use of traditional trial and error approach in tuning the parameters.

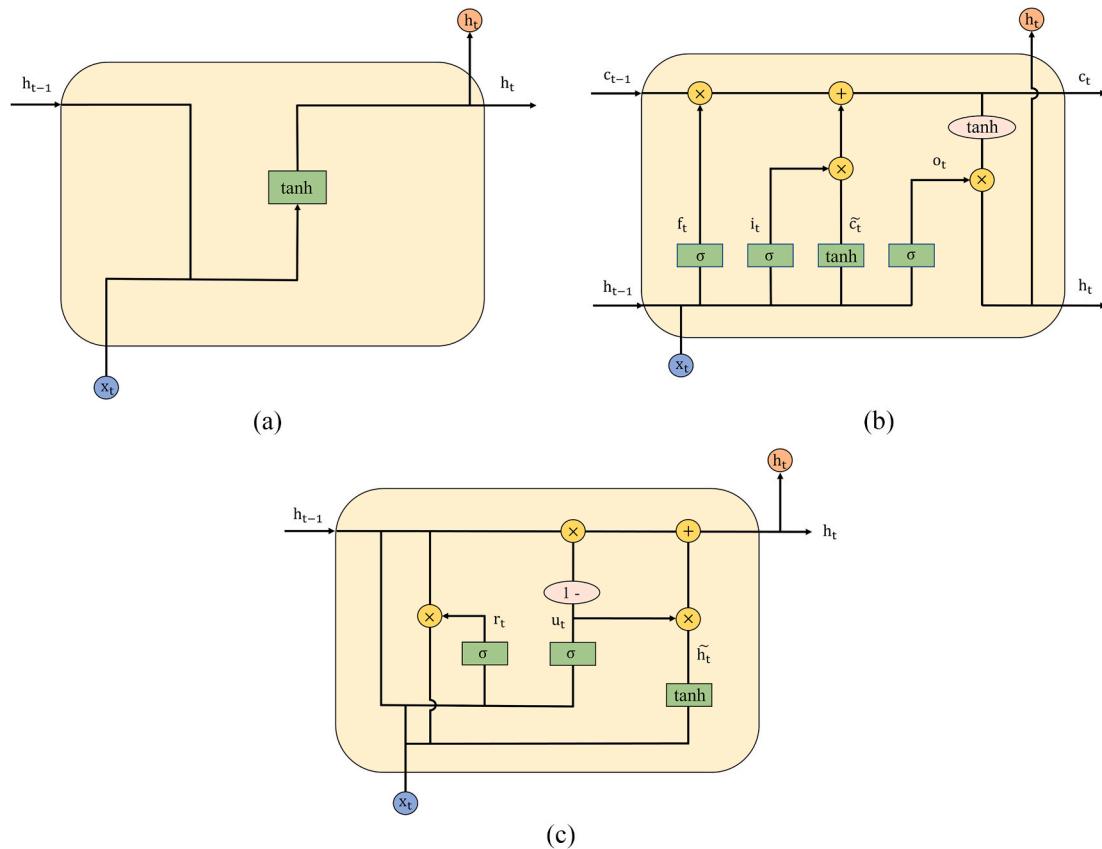
## 2.2. Feedforward neural network (FNN)

FNN is a ML algorithm that is formulated based on the functionalities of the biological neural networks. FNN comprises many calculating units which are known as artificial neurons or nodes. It has been demonstrated to be more successful in approximating the complex non-linear relationships between input and output vectors of a database than the conventional regression methods (Gharbi and Mansoori, 2005). There are different types of activation function used in FNN modeling, but the classical ones are the sigmoid function, the hyperbolic tangent, and the rectified linear unit (ReLU) function (Buduma and Locascio, 2017). In this paper, FNN, which is one of the most widely used ANNs as demonstrated in some literatures (Amini and Mohaghegh, 2019; Mohaghegh, 2011; Senthilkumar, 2010), is the chosen network with ReLU function as its activation function. It is also known as multilayer perceptron (MLP) and is made up of three layers, namely the input layer, the hidden layer, and the output layer. The topology of an arbitrary FNN is shown in Fig. 1. The green node is the bias node between the input and hidden layers whereas the orange node is the bias node between the hidden and output layers.

To ensure that the MLP learns the relationship between the input and output vectors of the database supplied, the MLP needs to undergo the training phase. Fundamentally, this training phase aims at optimizing the sets of weights and biases which minimize the pre-defined cost function, such as mean squared error (MSE). One of the classical methods of training is the backpropagation (BP) approach and it involves use of different algorithms, like steepest descent gradient, the Levenberg-Marquardt algorithm, the Powell-Beale conjugate gradient, Adam, and so on. In principle, after the forward propagation of the MLP, the resulting outputs will be compared with the targeted outputs. Errors are propagated back through the MLP in which the weights and biases are iteratively tuned and updated to achieve the optimum level. Apart from the conventional backpropagation algorithm, the metaheuristic algorithms can also be implemented to train the MLP. Therefore, in this paper, both backpropagation and metaheuristics algorithms are used to do the neural network training. Adam is the chosen backpropagation algorithm (Kingma and Ba, 2015) whereas Particle Swarm Optimization (PSO) is the metaheuristic algorithm used.

## 2.3. Particle swarm optimization (PSO)

PSO is an example of the metaheuristic population-based optimization algorithms that was proposed by Kennedy and Eberhart (1995) according to the social behavior of flying birds. The fundamental idea regarding the mechanism of PSO is that each particle corresponds to a potential solution to an optimization problem. The status of the particle is determined based upon its position and velocity in a dimensional space that is equal to the number of unknown parameters being optimized. Thereafter, the fitness value of the particle is computed by using a cost function such as MSE. Through several iterations, each particle updates its position until it converges to the optimum position through the minimization of the fitness value. In this context, pbest and gbest are determined at every iteration step. pbest refers to the local best position



**Fig. 2.** Illustration of three types of RNN used in this study (a) simple RNN (b) LSTM (c) GRU.

**Table 1**  
Data provided for each well in the Volvo field.

Abbreviation from Database	Description
DATEPRD	Date of Record
ON_STREAM_HRS	On stream hours
AVG_DOWNHOLE_PRESSURE	Average Downhole Pressure
AVG_DOWNHOLE_TEMPERATURE	Average Downhole Temperature
AVG_DP_TUBING	Average Differential Pressure of Tubing
AVG_ANNULES_PRESS	Average Annular Pressure
AVG_CHOKE_SIZE_P	Average Choke Size Percentage
AVG_WHP_P	Average Wellhead Pressure
AVG_WHT_P	Average Wellhead Temperature
BORE_OIL_VOL	Oil Volume from Well
BORE_WAT_VOL	Water Volume from Well
BORE_GAS_VOL	Gas Volume from Well
BORE_WI_VOL	Water Volume Injected
FLOW_KIND	Type of Flow (production or injection)
WELL_TYPE	Type of Well (oil production or water injection)

or the best position of a particle in the dimensional space (the lowest fitness value in this case) whereas gbest indicates the global best position or the overall best position of a particle hitherto in the entire population. The algorithm starts by randomly initializing the position and velocity of each particle. Thereafter, the respective fitness of each particle is computed in which pbest and gbest are determined and recorded. The velocity at current iteration step is then updated based on equation (7). The position of a particle for the next iteration step is updated based on equation (8). In the subsequent steps, positions and velocities of particles are updated iteratively by the pbest and gbest.

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

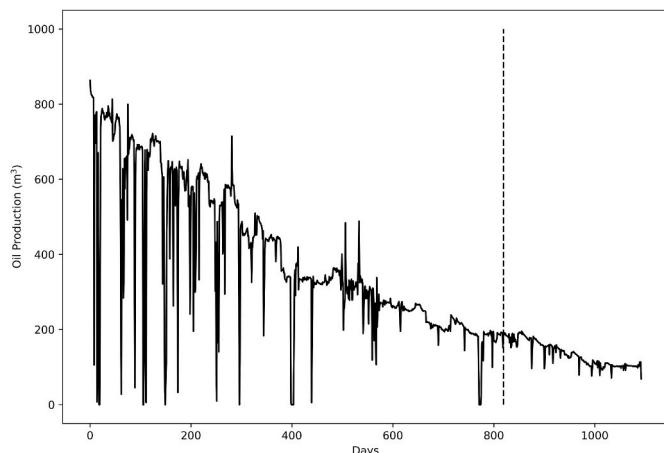
$$x_{jk,t+1} = x_{jk,t} + v_{jk,t+1} \quad (8)$$

**Table 2**  
Selected input and output data for data-driven modeling.

Parameters	Units
Input Data	
Time	Days
On stream hours	hours
Average Downhole Pressure	bar
Average Downhole Temperature	°C (degree Celsius)
Average Choke Size Percentage	%
Average Wellhead Pressure	bar
Average Wellhead Temperature	°C (degree Celsius)
Gas Volume from Well	m³ (daily)
Water Volume from Well	
Output Data	Units
Oil Volume from Well	m³ (daily)

**Table 3**  
Mean and standard deviation of input and output parameters of the production case considering all the data points.

Baseline Information		
Input and Output	Mean	Standard Deviation
Time	547	315.67
On stream hours	23.02	3.89
Average Downhole Pressure	261.01	15.54
Average Downhole Temperature	99.38	5.14
Average Choke Size Percentage	90.44	21.88
Average Wellhead Pressure	30.73	4.21
Average Wellhead Temperature	86.25	8.47
Gas Volume from Well	49,263.63	30,342.37
Water Volume from Well	3171.60	674.34
Oil Volume from Well	326.88	204.97



**Fig. 3.** Oil production of the well NO159-F-14H.

**Table 4**  
Parameters used in neural network training for both Adam and PSO.

Adam Parameters	Values
Number of iterations	2000
Learning rate	0.01
Exponential decay rates for the 1st moment estimates, $\beta_1$	0.9
Exponential decay rates for the 2nd moment estimates, $\beta_2$	0.999
Numerical stability constant, $\epsilon$	$10^{-7}$
PSO Parameters	Values
Number of iterations	2000
Number of particle swarms	100
Inertial Weight, $\omega$	0.8
Cognitive Learning Factor, $c_1$	1.05
Social Learning Factor, $c_2$	1.05

**Table 5**  
Optimized hyperparameters in SVR modeling.

Models	$\gamma$	C	$\epsilon$	$\kappa_1$	$\kappa_2$	$\kappa_3$
SVR-TE	0.5000	89.00	0.001000	–	–	–
SVR-PSO	0.4028	89.27	0.001802	0.4072	0.0171	0.5757

**Table 6**  
Performance metrics of the results estimated using the training, validation, and testing sets.

Datasets	Models	R <sup>2</sup>	RMSE
Training	SVR-TE	0.9951	13.88
	SVR-PSO	0.9944	14.68
	FNN-BP	0.9948	14.00
	FNN-PSO	0.9945	14.92
	Simple RNN	0.9945	14.46
	LSTM	0.9962	12.03
	GRU	0.9962	12.17
Validation	SVR-TE	0.9880	21.37
	SVR-PSO	0.9889	20.79
	FNN-BP	0.9911	19.13
	FNN-PSO	0.9923	15.75
	Simple RNN	0.9921	18.27
	LSTM	0.9910	19.51
	GRU	0.9940	15.75
Testing	SVR-TE	0.9764	30.83
	SVR-PSO	0.9936	16.61
	FNN-BP	0.9936	16.44
	FNN-PSO	0.9898	19.91
	Simple RNN	0.9941	15.37
	LSTM	0.9922	17.64
	GRU	0.9915	18.24

In equation (7),  $v_{jk,t}$  refers to the velocity of the  $j^{\text{th}}$  particle at iteration  $t$  in  $k^{\text{th}}$  dimension whereas  $x_{jk,t}$  represents its corresponding position.  $c_1$  and  $c_2$  respectively refer to the cognitive and social learning factors which govern the local and global search of the best position. They are determined by trial-and-error approach.  $r_1$  and  $r_2$  are random numbers retrieved from uniform (0, 1).  $\omega$  is inertial weight that was recommended by [Shi and Eberhart \(1998\)](#) to enhance the convergence performance.

#### 2.4. Recurrent neural network (RNN)

RNN is a subset of ANN, which is established to handle the input data that has sequential characteristics ([Alakeely and Horne, 2020](#); [Alom et al., 2019](#)). Examples of these sequential inputs include sets of words or sentences, document texts, stock price, etc. Fundamentally, RNN can preserve any previous information to the current task and such ability widens its application in different aspects, including speech recognition ([Amberkar et al., 2018](#); [Graves et al., 2013](#)) and language processing ([Guan et al., 2019](#); [Sutskever et al., 2014](#)). The fundamental mechanism of a basic RNN is that information can be preserved and sent from the current to the successive step ([Alom et al., 2019](#)) as illustrated by its architecture as shown in [Fig. 2a](#). Apart from this simple RNN, there are also other representations of RNN, such as Hopfield network, Echo state, Bi-directional, LSTM, GRU, and so forth. In this paper, we applied three examples of RNNs, including the simple RNN, LSTM, and GRU, to perform the well production forecast. The details regarding LSTM and GRU will be expounded later. The simple RNN used in this study consists of one hidden layer and one output layer and the respective mathematical formulation is presented below:

$$h_t = \gamma(W_h x_t + U_h h_{t-1} + b_h) \quad (9)$$

$$y_t = \gamma(W_y h_t + b_y) \quad (10)$$

where  $h_t$  is known as the vector of hidden-state or hidden layer. It is computed as shown in equation (9) by summing up three terms and placing the summation into the activation function that is represented as  $\gamma$ . In this work, the activation function used is the hyperbolic tangent. Also,  $y_t$  is the output vector that is determined by adding two terms into the activation function as shown in equation (10). For the other terms,  $x_t$  is the input vectors,  $W$  and  $U$  represent the weights, and  $b$  is the bias term. It is important to know that the subscripts  $t$  and  $t-1$  correspondingly refer to the current and previous timesteps. The subscript  $h$  indicates the properties of the hidden layer whereas the subscript  $y$  represents those of the output layer. The use of these notations also applies to the mathematical formulations of LSTM and GRU in the following sections. For LSTM, the subscripts  $f, i, c$ , and  $o$  correspondingly denote the relevant properties of forget gate, input gate, cell state and output gate. For GRU, the subscripts  $u$  and  $r$  respectively mean the properties of update gate and reset gate. The pertinent details will follow later.

#### 2.5. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)

Albeit the simple RNN can be practically robust, it still has a limitation, namely having the problem of vanishing gradient ([Alom et al., 2019](#); [Hochreiter and Schmidhuber, 1997](#); [Li et al., 2019](#)). This limitation circumvents the simple RNN from exploiting the long-term information ([Alom et al., 2019](#); [Hochreiter and Schmidhuber, 1997](#); [Li et al., 2019](#)). This implies that it is unable to store large amount of information from previous iterations for a more accurate prediction of the outputs. Therefore, more complicated versions of RNN, which are LSTM and GRU, have been utilized. LSTM was first developed by [Hochreiter and Schmidhuber \(1997\)](#) to ensure the long-term dependencies on the previous information. The architecture of the LSTM employed in this study is portrayed in [Fig. 2b](#). The respective formulas are expressed below:

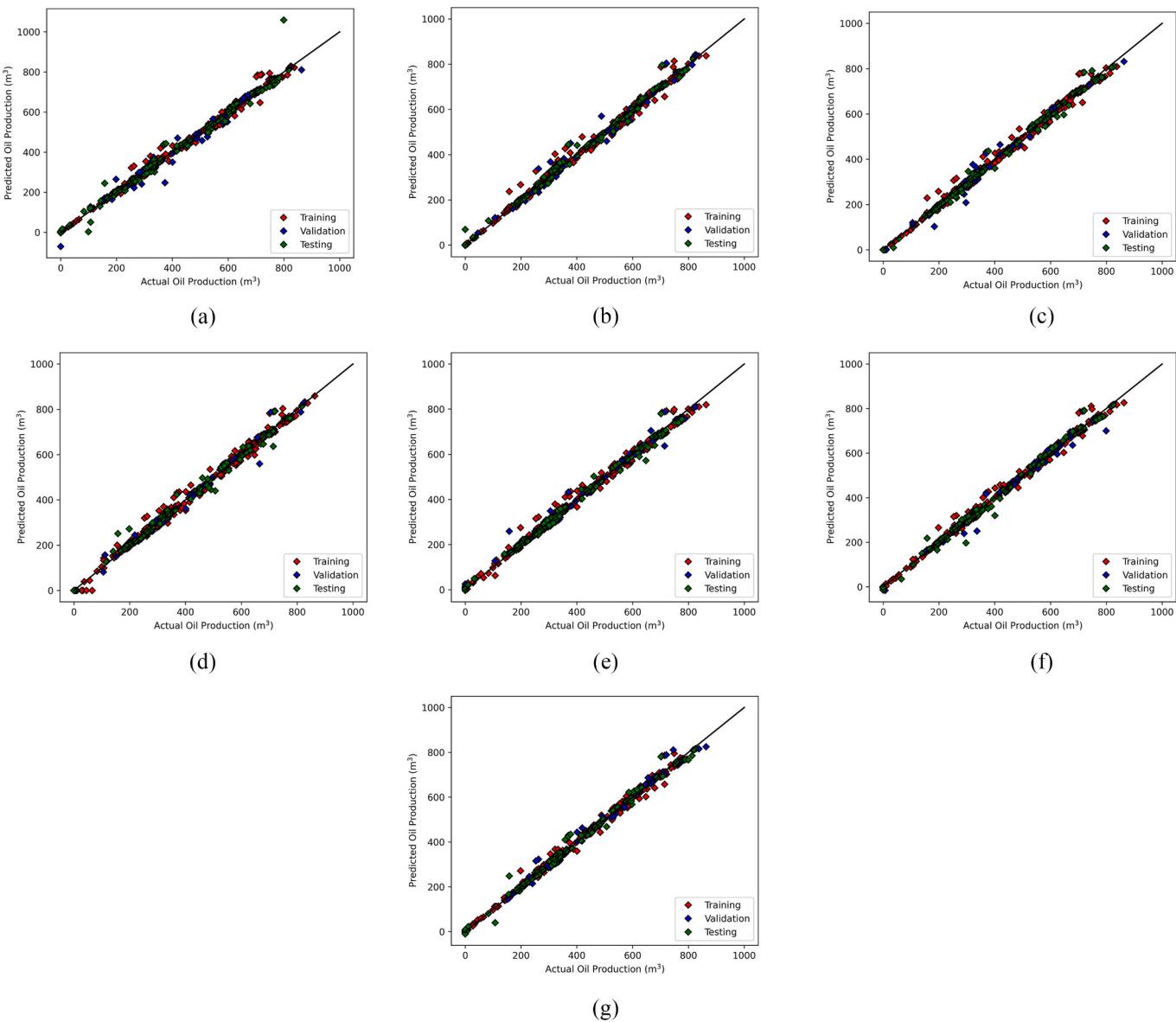


Fig. 4. Cross plot of the actual and predicted oil production (a) SVR-TE (b) SVR-PSO (c) FNN-BP (d) FNN-PSO (e) simple RNN (f) LSTM (g) GRU.

**Table 7**  
Performance metrics of the results estimated by using the blind case.

Datasets	Models	R <sup>2</sup>	RMSE
Blind Validation	SVR-TE	0.9476	7.34
	SVR-PSO	0.9644	6.04
	FNN-BP	0.9538	6.89
	FNN-PSO	0.9574	6.61
	Simple RNN	0.9665	5.87
	LSTM	0.9712	5.45
	GRU	0.9700	5.56

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (11)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (12)$$

$$\tilde{c}_t = \gamma(W_c x_t + U_c h_{t-1} + b_c) \quad (13)$$

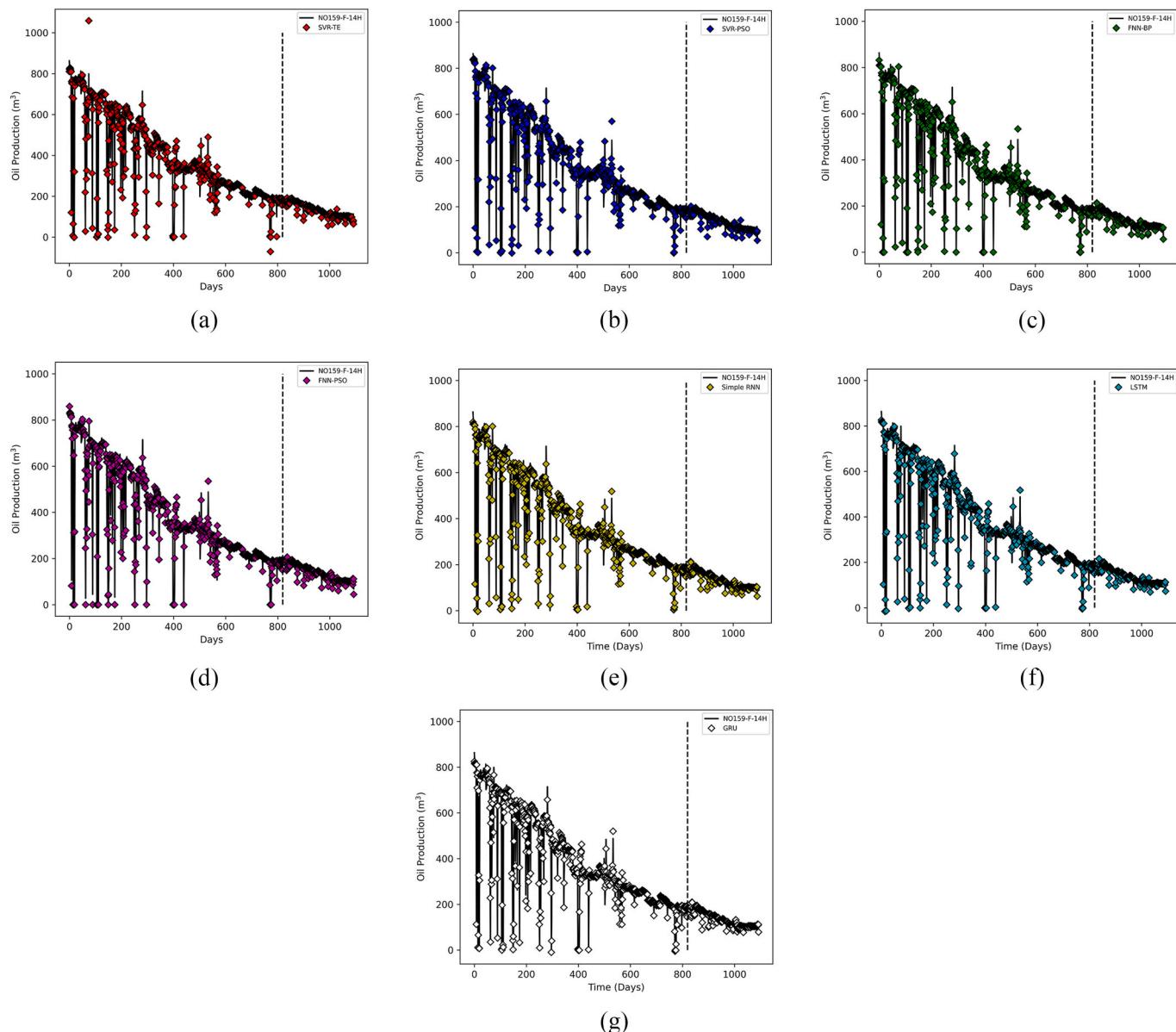
$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \quad (14)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (15)$$

$$h_t = o_t \times \gamma(c_t) \quad (16)$$

The fundamental idea of LSTM revolves around a cell state  $c_t$  (shown as the horizontal top line in Fig. 2b) in which the addition or removal of any information is conducted through three gates, namely forget gate  $f_t$ , input gate  $i_t$ , and output gate  $o_t$  (Alom et al., 2019). These gates make assessments as if the sequential input data is valuable or not to be kept (Alom et al., 2019; Li et al., 2019). By doing so, relevant information can be preserved to the downstream. First, the forget gate plays a pivotal role to decide if information should be kept or omitted based upon equation (11). In this aspect, the information in the form of input and hidden state will be discarded (retained) if  $f_t$  approximates zero (one) (Li et al., 2019). Pertaining to the input gate, it is computed to update the cell state and through this update, the importance of the input being sent to the next cell is assessed. Moreover, about the output gate, it determines the output for the hidden states as shown in equation (16). It can be noticed that the recurrent activation function used in LSTM is a sigmoid function that is denoted as  $\sigma$ .

GRU is another development of RNN, which was initiated by Cho et al. (2014), that is employed in this paper. As compared to LSTM, GRU



**Fig. 5.** Oil production profile (a) SVR-TE (b) SVR-PSO (c) FNN-BP (d) FNN-PSO (e) simple RNN (f) LSTM (g) GRU.

**Table 8**

Performance metrics of all seven models considering all data points.

Datasets	Models	R <sup>2</sup>	RMSE
All	SVR-TE	0.9935	16.52
	SVR-PSO	0.9952	14.21
	FNN-BP	0.9956	13.65
	FNN-PSO	0.9952	14.15
	Simple RNN	0.9957	13.51
	LSTM	0.9961	12.69
	GRU	0.9964	12.28

only consists of two gates, which are the reset gate  $r_t$  and the update gate  $u_t$ . The function of the reset gate is to evaluate as if new information should be passed, which is like those of forget and input gates (Li et al., 2019). Thereafter, the reset gate decides on how extensively the previous information should be forgotten. According to the formulas of GRU shown below, it can be inferred that its simpler framework enables it to be more computationally favorable as compared to LSTM (Alom et al., 2019).

$$u_t = \sigma(W_u x_t + U_u h_{t-1} + b_u) \quad (17)$$

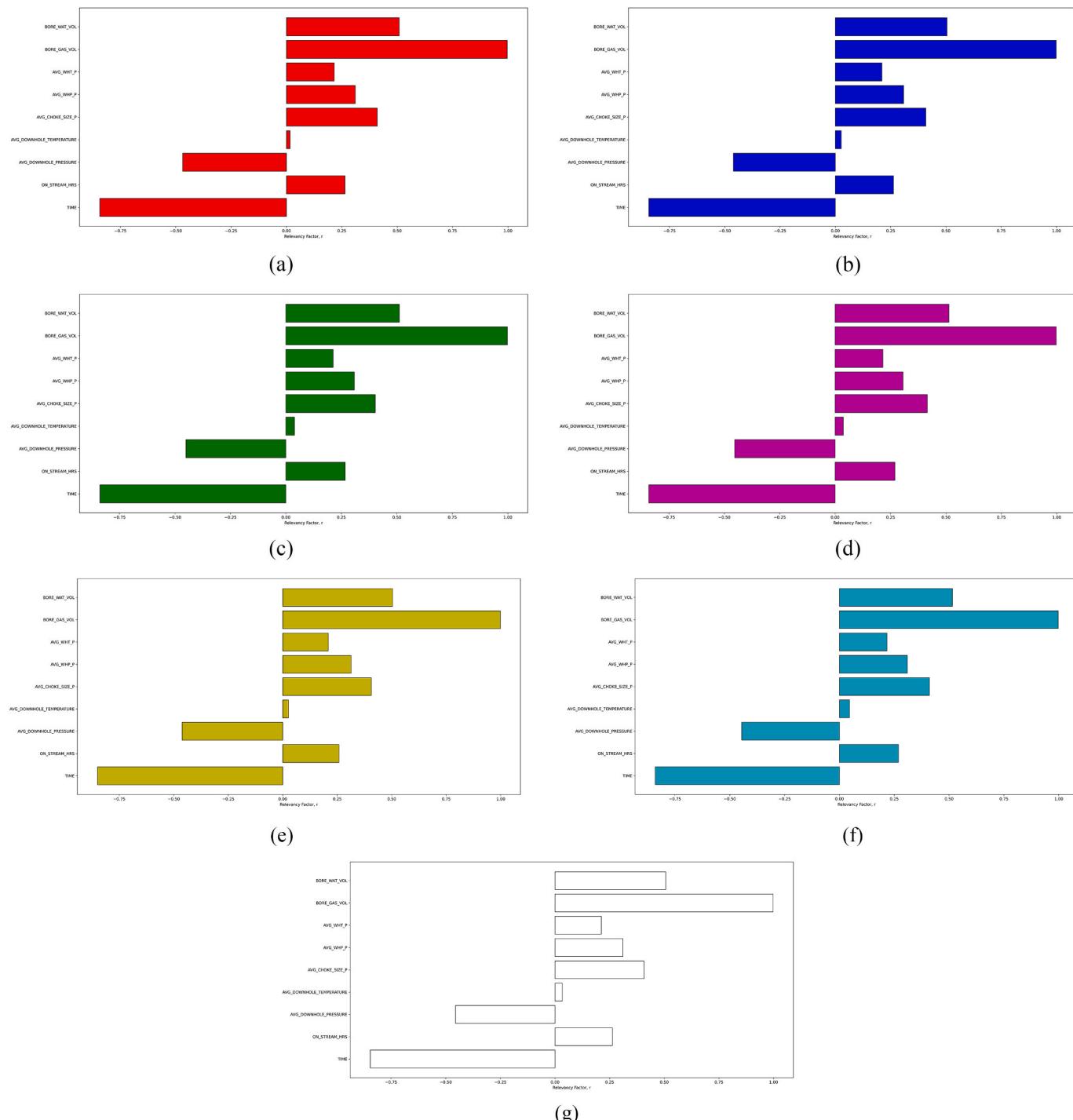
$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (18)$$

$$\tilde{h}_t = \gamma(W_h x_t + U_h [r_t \times h_{t-1}] + b_h) \quad (19)$$

$$h_t = (1 - u_t) \times h_{t-1} + u_t \times \tilde{h}_t \quad (20)$$

### 3. Methodology

Having a good model that helps predicting hydrocarbon production is crucial in reservoir management. As mentioned previously, we have developed seven models in this work: FNN-BP, FNN-PSO, SVR-TE, SVR-PSO, simple RNN, LSTM, and GRU. To build these data-driven models, we need to first know the source of data because it is the main building blocks of these models. The details regarding the data will follow.



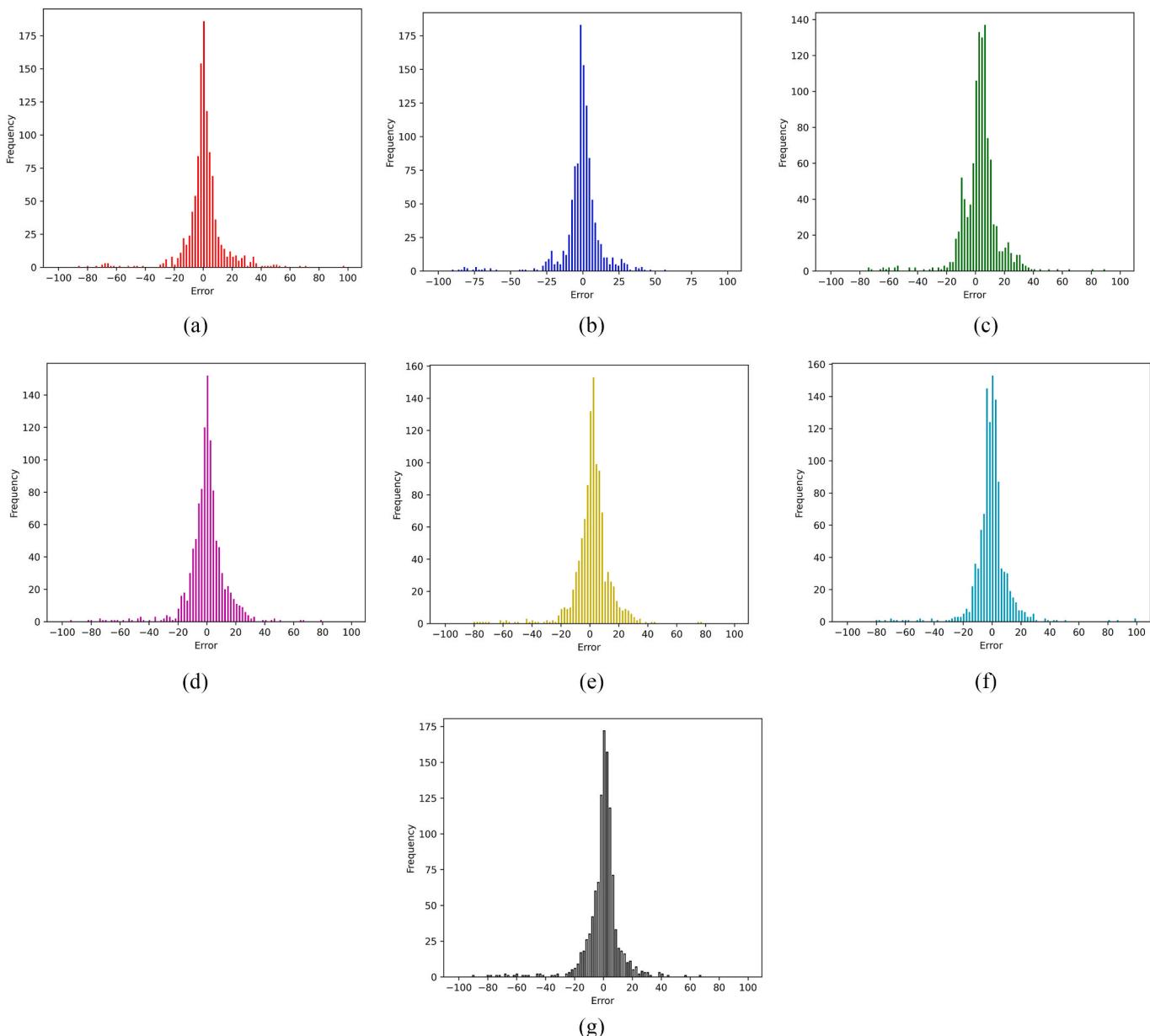
**Fig. 6.** Relative effect of each input parameter on the output predicted by each model considering all data points (a) SVR-TE (b) SVR-PSO (c) FNN-BP (d) FNN-PSO (e) simple RNN (f) LSTM (g) GRU.

### 3.1. Field data

In this work, the data from Volve field (Equinor, 2018) on the Norwegian Continental Shelf was utilized. According to the field development plan report retrieved from Equinor (2020), Volve field is a 2 km by 3 km oil-bearing reservoir and is located at a depth between 2750 m and 3210 m below sea level. It comprises sandstone and has average properties with permeability of about 1000 mD (from well testing), porosity of 0.21, and net-to-gross ratio of 0.93. The water saturation of oil-bearing zone is on average 0.2. At the depth of 3060 m, the reservoir

pressure and temperature are 340 bar and 110 °C, respectively. Pertaining to the characteristics of crude oil from Volve field, according to ExxonMobil (2018), the API gravity is 29.1°, the specific gravity is 0.881, and the viscosity at 20 °C is 22.5 cSt. For more details, kindly peruse the crude oil assay released by ExxonMobil (2018).

Equinor (2018) has released this database to public in May 2018 for the purpose of research and development. In this aspect, there are different types of data in the database, including seismic data, well log data, reservoir simulation model, etc. However, only the real-field production data is used in this study. Regarding the production data,



**Fig. 7.** Distribution of errors (a) SVR-TE (b) SVR-PSO (c) FNN-BP (d) FNN-PSO (e) simple RNN (f) LSTM (g) GRU.

it consists of the data of 7 wells, namely NO15/9-F-1 C, NO15/9-F-11H, NO15/9-F-12H, NO159-F-14H, NO15/9-F-15D, NO15/9-F-4AH, and NO15/9-F-5AH. Each well consists of the data as shown in [Table 1](#):

The production data was recorded daily. For illustrative purpose, only the well NO159-F-14H is used in this study. For this well, the production period lasts from February 2008 to September 2016. However, for practical purpose, only the data between July 2013 and July 2016, which lasts for 1093 days, is used. In addition to this, not all the data provided will be used and the selected data used for data-driven modeling is presented in [Table 2](#). The selection of input and output data was done based upon knowledge of reservoir and production engineering, but it can be conveniently done by using feature selection method ([Zanjani et al., 2020](#)). To further facilitate the readers' understanding of the production scenario, the mean and standard deviation of each parameter are determined and presented in [Table 3](#). In addition, the oil production profile of the well NO159-F-14H between July 2013 and July 2016 is plotted in [Fig. 3](#). The dashed vertical line in [Fig. 3](#) will be explained later.

### 3.2. Model development

The data needs to be pre-processed before it is used to build the models. As explained earlier, there are 10 types of data being utilized and each type contributes to 1093 data points. Hence, this sums up to 10,930 data points. Each data point is then normalized as follows:

$$x_{i,\text{normalized}} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \quad (21)$$

In equation (21),  $x_{i,\text{normalized}}$  is the normalized value of  $x_i$  that is any data point out of the 1093 data points under each type of data as shown in [Table 2](#).  $x_{\max}$  and  $x_{\min}$  denote the maximum and minimum values under each data type in [Table 2](#). Thereafter, the normalized data points are divided into two different sets, namely the modeling set and the prediction set, based on a ratio of 7.5:2.5. This implies that the first 8190 data points out of 10,930 data points will be employed to develop the data-driven models whereas the remaining 2740 data points are used as the blind case to evaluate the predictive performance of the models. It is essential to divulge that the division for modeling and prediction sets is

done arbitrarily for practical purposes. It relies upon the consideration of the modeler about the size of the dataset preserved for prediction. For a more vivid illustration, the modeling set corresponds to the data points on the left of the dashed vertical line in Fig. 3 whereas the prediction set corresponds to the right of the line. Besides, 70% of the data points from the modeling set is used as the training set and the remaining 30% is equally divided into the validation and testing sets. In this context, only the training set is utilized to develop and train the models. The validation set is employed to prevent the overfitting of the models whereas the testing set ensures that the models have a good predictive performance prior to being verified by the data from the blind case (Mohaghegh, 2017). The performance of the models is determined by using two different metrics, which are the correlation coefficient  $R^2$  and the root mean squared error (RMSE). The formulas of the performance metrics are presented as follows:

$$R^2 = 1 - \frac{\sum_{j=1}^N (q_j^{exp} - q_j^{cal})^2}{\sum_{j=1}^N (q_j^{cal} - \bar{q})^2} \quad (22)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^N (q_j^{exp} - q_j^{cal})^2} \quad (23)$$

where  $N$  means the total number of data,  $q_j^{exp}$  is the actual oil production at timestep  $j$ ,  $q_j^{cal}$  is the oil production estimated by the models at timestep  $j$ , and  $\bar{q}$  is the mean actual oil production. For the development of FNN-BP, FNN-PSO, and the three RNNs, the data from the training set is fed into the neural network to enable the network to learn the relationship between input and output data. Pertaining to this, the pre-defined cost function implemented in the neural network training is the MSE. Therefore, during the training phase, the weights and biases will be iteratively adjusted as explained to minimize the cost function.

Pertaining to the specifics of the data-driven models, the architectures of both FNNs are the same, which include one input layer with 9 nodes, one hidden layer with 30 nodes, and one output layer with only one node. For the three RNNs, each of them also comprises only one hidden layer and one output layer. Besides that, each of the three RNN representations also has 30 hidden nodes and 1 output node. The number of hidden nodes and layers for both FNNs and RNNs is determined by using the trial-and-error approach. The relevant parameters used to conduct this neural network training phase are presented in Table 4. From Table 4, it is better to reiterate that Adam has only been implemented to train all the RNNs and FNN-BP. For FNN-PSO, since each of the weights (biases) is represented as one particle, the number of particle swarms is the number of sets of particles employed in the training phase.

Regarding the development of SVR and SVR-PSO, it is important to achieve the optimum values of the hyperparameters  $\gamma$ ,  $C$ , and  $\epsilon$  to develop models with good performance. For SVR-PSO, the hyperparameters are tuned such that the objective function will be minimized. The objective function consists of the corresponding MSE of the training, validation, and testing sets, and it is expressed as shown in equation (24). For SVR-PSO, there are three additional parameters to be adjusted, namely the weighting factors  $\kappa_1$ ,  $\kappa_2$ , and  $\kappa_3$  for each MSE. To conduct the tuning with PSO, 200 iterations and 20 particle swarms are used. Furthermore, the inertial weight used here is 0.40 while both learning factors are 1.05. Refer to Table 5 for the values of these optimized hyperparameters.

$$MSE_{SVR-PSO} = \kappa_1 \times MSE_{Training} + \kappa_2 \times MSE_{Validation} + \kappa_3 \times MSE_{Testing} \quad (24)$$

To generate the initial population of the swarm particles, we used the distribution of uniform (0.01, 1.5) for  $\gamma$ , uniform (12, 90) for  $C$ , uniform (0.0001, 0.1) for  $\epsilon$ , uniform (0, 0.5) for  $\kappa_1$  and uniform (0, 0.5) for  $\kappa_2$ . Without determining the initial  $\kappa_3$ , we optimized it by subtracting the sum of optimized  $\kappa_1$  and  $\kappa_2$  at each iteration from 1.

#### 4. Results and discussion

We have established seven data-driven models to predict the daily oil production of a real-field well. To determine if these models will exhibit excellent predictive performance, their corresponding training performance need to be evaluated first. Pertaining to this, the models with excellent training results will generally be able to produce predictions within a good level of accuracy. In this work, the training performance of each of the seven models is presented in Table 6. In addition to the performance metrics computed using the training data, those calculated using the validation and testing data are also shown. During the development stage, if the models demonstrate good performance with the validation data, it implies that the overfitting issue may be eluded. Thereafter, the predictability of the models can be evaluated using the testing data. It is important to understand that only the training data is employed to build the models. The other data is utilized to provide useful insights regarding the training process.

From Table 6, it is inferred all the seven models demonstrate excellent results of training, validation, and testing with  $R^2$  exceeding 0.99 and RMSE being at most 30.83. To be more precise, LSTM has the best training performance in terms of  $R^2$  and RMSE compared to the other models. However, when the models are fed with the validation data, GRU exhibits the best results. In addition to this, as the models are verified with the testing data, simple RNN performs the best. Therefore, RNN-based models generally illustrate better outcomes than both SVR-based and FNN-based models in terms of training, validation, and testing. Despite these better results exhibited by these RNN-based models, the performances of SVR-based and FNN-based models are deemed to be practically excellent. Nevertheless, the performance metrics shows that all models have undergone an excellent development phase. We need to be cognizant that having satisfactory modeling outcome does not necessarily imply that the models can directly be used. They still must be evaluated by the data from the blind case to further verify their robustness.

The cross-plots of the actual and the predicted oil production are presented for SVR-TE in Fig. 4a, SVR-PSO in Fig. 4b, FNN-BP in Fig. 4c, FNN-PSO in Fig. 4d, simple RNN in Fig. 4e, LSTM in Fig. 4f, and GRU in Fig. 4g. In general, most of the data points lie on the 45° line which indicates high accuracy. Nevertheless, Fig. 4a exhibits that there is an outlier of the validation data being less than zero and another outlier of the testing data being highly overestimated. This implies that the overall training performance can still be improved albeit the performance metrics suggest otherwise. Moreover, Fig. 4d shows that there are some outliers from the training data that are underestimated by FNN-PSO. These outliers do not greatly affect the overall training performance of the model but contribute to the relatively less satisfying training performance compared with FNN-BP. For the RNN-based models, these plots generally add more confidence that the overall training performance of each of the three models is practically excellent. Additionally, there is no obvious outlier being detected in the plots, which are produced by using these models.

After the modeling phase is completed, we need to provide data from the blind case to justify if the models are ready to be employed. As explained, the data from the blind case is retrieved from the data points of the remaining 274 days. When these data are supplied into the built models, their performance metrics are calculated and recorded in Table 7. For a more vivid illustration, all the data points (1093 data points of oil production) are plotted alongside the prediction yielded by all the seven models in Fig. 5. For clarification, the statistics provided in Table 7 only consider the data points on the right side of the vertical dashed line in the figures. Based on Table 7, it can be observed that the use of PSO improves the predictive performance of the models in this work. For SVR, using PSO to tune the hyperparameters improves the  $R^2$  by 1.77% and the RMSE by 17.7%. Therefore, using a metaheuristic algorithm to tune the hyperparameters does not only reduce the computational effort, but also helps to attain a higher accuracy of

prediction. For FNN modeling, when PSO is utilized to conduct the training, the  $R^2$  and RMSE are respectively enhanced by 0.38% and 4.06%. Albeit the improvement is not significant, it provides useful insight that the application of metaheuristic algorithm is viable in modeling FNN and can have a good predictive performance.

Moreover, it is deduced that LSTM has the best performance with  $R^2$  being greater than 0.97 and RMSE being about 5.4. However, it is also important to observe that in this study, the performance of LSTM is slightly better than those of GRU and simple RNN. With respect to simple RNN, LSTM correspondingly improves  $R^2$  and RMSE by 0.49% and 7.2% whereas the enhancements induced by GRU are respectively 0.36% and 5.3%. In other words, the improvement of prediction accuracy is not very significant by applying more complicated representation of RNN. Therefore, from Fig. 5, the robustness of ML techniques in capturing the fluctuating trend of the data is clearly portrayed. In this context, the conventional DCA approach is only able to perform the "curve fitting" and reflect the general declining trend of the data. In addition to this, for the purpose of more comprehensive comparison, the performance metrics considering all the 1093 data points are calculated and tabulated in Table 8 for each model. As the result shows, GRU outperforms the other models. In general, all the models can capture the overall trend of the data points. Nonetheless, for SVR-TE, it can be noted that there are both overestimation and underestimation of values in two of the data points. This corresponds to the outliers mentioned earlier. Despite this, SVR-TE still performs reasonably well in estimating the output of the data from the blind case.

Furthermore, the relevancy factor ( $r$ ) has been implemented to evaluate the relative importance of these input variables on the predicted output by the models. In this case, higher absolute value of  $r$  indicates more significant relative effect on the output (Chen et al., 2014; Nait Amar, 2020; Nait Amar et al., 2021). The relevancy factor can be mathematically expressed as follows:

$$r(I_k, q) = \frac{\sum_{j=1}^N (I_{k,j} - \bar{I}_j)(q_j - \bar{q})}{\sqrt{\sum_{j=1}^N (I_{k,j} - \bar{I}_j)^2 \sum_{j=1}^N (q_j - \bar{q})^2}} \quad (25)$$

In equation (25), the data point index (or timestep in this case) is indicated as  $j$ ,  $I_k$  denotes the  $k$ th input parameter, and  $\bar{I}_k$  means the respective average value. Besides that,  $q$  and  $\bar{q}$  correspondingly represent the predicted output value and its average. The relevancy factor of each input parameter is depicted in Fig. 6. As shown, gas volume from well (or gas production) has the most influential impact on the output, which is oil volume from well (oil production). Distribution of the errors corresponding to the predictions (of all data points) performed by all the seven models are also demonstrated as histogram in Fig. 7. It can be observed that all seven models display a normal distribution that has a center being close to errors with zero values. Such distribution provides extra confidence to the integrity and robustness of the models developed in this paper.

## 5. Conclusions

In this work, SVR-TE, SVR-PSO, FNN-BP, FNN-PSO, simple RNN, LSTM, and GRU models have been developed to predict the oil production of a well in Volve field. These models have been trained, validated, and tested to ensure that they have learnt the relationship between input and output models before being blind validated.

Generally, RNN-based models outperformed the SVR-based and FNN-based models in terms of training and prediction. To be more specific, LSTM outperformed the other six models in the case of training. Besides that, GRU performed the best in the validation phase whereas simple RNN yielded the best outcome in the testing phase. However, the training performance and predictability of SVR-based and FNN-based models are still practically excellent. Apart from these, we can infer

that PSO contributes to the enhancement of SVR modeling in terms of training, but not in the case of FNN modeling due to the existence of several outliers. Nevertheless, we illustrated that the application of PSO in data-driven modeling could induce improvements although such improvements might not be significant for FNN modeling. Additionally, during the prediction phase, LSTM produced the most accurate results. Also, when considering all the data points, the performance metrics computed by using the results estimated by GRU were the best. Finally, the resemblance of the error distribution produced by each predictive model to a normal distribution with center close to zero further displayed the reliability of the models built in this work.

## Authors' contributions

Cuthbert Shang Wui Ng: Methodology, Data Curation, Analysis and Investigation, Modeling, Software, Writing, Editing, Revising. Ashkan Jahanbani Ghahfarokhi: Supervision, Methodology, Writing, Reviewing, Revising, and Editing. Menad Nait Amar: Methodology, Writing, Reviewing, Editing and Revising.

## Declaration of competing interest

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

## Acknowledgement

This research is a part of BRU21 – NTNU Research and Innovation Program on Digital Automation Solutions for the Oil and Gas Industry ([www.ntnu.edu/bru21](http://www.ntnu.edu/bru21)).

## References

- Adedigba, S.A., Oloruntobi, O., Khan, F., Butt, S., 2018. Data-driven dynamic risk analysis of offshore drilling operations. *J. Petrol. Sci. Eng.* 165 <https://doi.org/10.1016/j.petrol.2018.02.049>.
- Akande, K.O., Owolabi, T.O., Olatunji, S.O., AbdulRaheem, A.A., 2017. A hybrid particle swarm optimization and support vector regression model for modelling permeability prediction of hydrocarbon reservoir. *J. Petrol. Sci. Eng.* 150 <https://doi.org/10.1016/j.petrol.2016.11.033>.
- Alakeely, A., Horne, R.N., 2020. Simulating the behavior of reservoirs with convolutional and recurrent neural networks. In: SPE Reservoir Evaluation and Engineering. <https://doi.org/10.2118/201193-PA>.
- Alali, A.M., Abughaban, M.F., Aman, B.M., Ravela, S., 2021. Hybrid data driven drilling and rate of penetration optimization. *J. Petrol. Sci. Eng.* 200 <https://doi.org/10.1016/j.petrol.2020.108075>.
- Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidiqe, P., Nasrin, M.S., Hasan, M., Van Essen, B.C., Awwal, A.A.S., Asari, V.K., 2019. A state-of-the-art survey on deep learning theory and architectures. *Electron.* <https://doi.org/10.3390/electronics8030292>.
- Amberkar, A., Awasarmol, P., Deshmukh, G., Dave, P., 2018. Speech Recognition using Recurrent Neural Networks. In: Proceedings of the 2018 International Conference on Current Trends towards Converging Technologies. <https://doi.org/10.1109/ICCTCT.2018.8551185>.
- Amini, S., Mohaghegh, S., 2019. Application of machine learning and artificial intelligence in proxy modeling for fluid flow in porous media. *Fluid.* <https://doi.org/10.3390/fluids4030126>.
- Arps, J.J., 1945. Analysis of decline curves. *Trans. AIME* 160. <https://doi.org/10.2118/945228-g>.
- Aydin, G., 2015. Regression models for forecasting global oil production. *Petrol. Sci. Technol.* 33 <https://doi.org/10.1080/10916466.2015.1101474>.
- Bao, A., Gildin, E., Huang, J., Coutinho, E.J.R., 2020. Data-driven end-to-end production prediction of oil reservoirs by EnKF-enhanced recurrent neural networks. In: SPE Latin American and Caribbean Petroleum Engineering Conference Proceedings. <https://doi.org/10.2118/199005-ms>.
- Barbosa, L.F.F.M., Nascimento, A., Matthias, M.H., de Carvalho, J.A., 2019. Machine learning methods applied to drilling rate of penetration prediction and optimization - a review. *J. Petrol. Sci. Eng.* 183, 106332. <https://doi.org/10.1016/j.petrol.2019.106332>.
- Buduma, N., Locascio, N., 2017. *Fundamentals of Deep Learning : Designing Next-Generation Machine Intelligence Algorithms, Designing Next-Generation Machine Intelligence Algorithms*.
- Calvette, T., Gurwicz, A., Abreu, A.C., Pacheco, M.A.C., 2020. Forecasting smart well production via deep learning and data driven optimization. In: Offshore Technology Conference Brasil 2019. <https://doi.org/10.4043/29861-ms>. OTCB 2019.

- Cao, Q., Banerjee, R., Gupta, S., Li, J., Zhou, W., Jeyachandra, B., 2016. Data driven production forecasting using machine learning. In: Society of Petroleum Engineers - SPE Argentina Exploration and Production of Unconventional Resources Symposium. <https://doi.org/10.2118/180984-ms>.
- Chen, G., Fu, K., Liang, Z., Sema, T., Li, C., Tontiwachwuthikul, P., Idem, R., 2014. The genetic algorithm based back propagation neural network for MMP prediction in CO2-EOR process. Fuel. <https://doi.org/10.1016/j.fuel.2014.02.034>.
- Chiroma, H., Abdulkareem, S., Abubakar, A.I., Herawan, T., 2014. Kernel functions for the support vector machine: comparing performances on crude oil price data. In: Advances in Intelligent Systems and Computing. [https://doi.org/10.1007/978-3-319-07692-8\\_26](https://doi.org/10.1007/978-3-319-07692-8_26).
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. <https://doi.org/10.3115/v1/d14-1179>.
- Connor, J.T., Martin, R.D., Atlas, L.E., 1994. Recurrent neural networks and robust time series prediction. IEEE Trans. Neural Netw. 5 <https://doi.org/10.1109/72.279188>.
- Crnogorac, M., Tanašević, M., Danilović, D., Marićić, V.K., Leković, B., 2020. Selection of artificial lift methods: a brief review and new model based on fuzzy logic. Energies 13. <https://doi.org/10.3390/en13071758>.
- Elmabrouk, S., Shirif, E., Mayorga, R., 2014. Artificial neural network modeling for the prediction of oil production. Petrol. Sci. Technol. 32 <https://doi.org/10.1080/10916466.2011.605093>.
- Equinor, 2020. Volvo Data Village. Azure Storage Explorer.
- Equinor, 2018. Disclosing all Volvo data [WWW document]. 6.28.21. <https://www.equinor.com/en/news/14jun2018-disclosing-volve-data.html>.
- ExxonMobil, 2018. Crude Oil Assay Volvo.
- Ezugwu, A.E., Adeleke, O.J., Akinyelu, A.A., Viriri, S., 2020. A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems. Neural Comput. Appl. 32 <https://doi.org/10.1007/s00521-019-04132-w>.
- Fanchi, J.R., Cooksey, M.J., Lehman, K.M., Smith, A., Fanchi, A.C., Fanchi, C.J., 2013. Probabilistic decline curve analysis of barnett, fayetteville, haynesville, and woodford gas shales. J. Petrol. Sci. Eng. 109 <https://doi.org/10.1016/j.petrol.2013.08.002>.
- Forrester, A.J., Sobester, A., Keane, A.J., 2008. Engineering Design via Surrogate Modelling : A Practical Guide. J. Wiley.
- Frausto-Sólís, J., Chi-Chim, M., Sheremetov, L., 2015. Forecasting oil production time series with a population-based simulated annealing method. Arabian J. Sci. Eng. 40 <https://doi.org/10.1007/s13369-015-1587-z>.
- Gan, C., Cao, W.H., Liu, K.Z., Wu, M., Wang, F.W., Zhang, S.B., 2020. A new hybrid bat algorithm and its application to the ROP optimization in drilling processes. IEEE Trans. Ind. Informatics 16. <https://doi.org/10.1109/TII.2019.2943165>.
- Gharbi, R.B.C., Mansoori, G.A., 2005. An introduction to artificial intelligence applications in petroleum exploration and production. J. Petrol. Sci. Eng. <https://doi.org/10.1016/j.petrol.2005.09.001>.
- Graves, A., Mohamed, A.R., Hinton, G., 2013. Speech recognition with deep recurrent neural networks. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. <https://doi.org/10.1109/ICASSP.2013.6638947>.
- Guan, M., Cho, S., Petro, R., Zhang, W., Pasche, B., Topaloglu, U., 2019. Natural language processing and recurrent network models for identifying genomic mutation-associated cancer treatment change from patient progress notes. JAMIA Open 2. <https://doi.org/10.1093/jamiaopen/ooy061>.
- Han, B., Bian, X., 2018. A hybrid PSO-SVM-based model for determination of oil recovery factor in the low-permeability reservoir. Petroleum. <https://doi.org/10.1016/j.petlm.2017.06.001>.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780.
- Hong, A., Bratvold, R.B., Lake, L.W., Ruiz Maraggi, L.M., 2019. Integrating model uncertainty in probabilistic decline-curve analysis for unconventional-oil-production forecasting. In: SPE Reservoir Evaluation and Engineering. <https://doi.org/10.2118/194503-PA>.
- Jochen, V.A., Spivey, J.P., 1996. Probabilistic reserves estimation using decline curve analysis with the bootstrap method. In: Society of Petroleum Engineers - SPE Annual Technical Conference and Exhibition. <https://doi.org/10.2523/36633-ms>.
- Kavzoglu, T., Colkesen, I., 2009. A kernel functions analysis for support vector machines for land cover classification. Int. J. Appl. Earth Obs. Geoinf. 11, 352–359. <https://doi.org/10.1016/J.JAG.2009.06.002>.
- Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: IEEE International Conference on Neural Networks - Conference Proceedings. <https://doi.org/10.4018/ijmfp.2015011004>.
- Khamis, M., Elhaj, M., Abdulraheem, A., 2020. Optimization of choke size for two-phase flow using artificial intelligence. J. Pet. Explor. Prod. Technol. 10 <https://doi.org/10.1007/s13202-019-0734-6>.
- Kingma, D.P., Ba, J.L., 2015. Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.
- Kondori, J., Miah, M.I., Zendehboudi, S., Khan, F., Heagle, D., 2021. Hybrid connectionist models to assess recovery performance of low salinity water injection. J. Petrol. Sci. Eng. 197 <https://doi.org/10.1016/j.petrol.2020.107833>.
- Lee, K., Lim, J., Yoon, D., Jung, H., 2019. Prediction of shale-gas production at duvernay formation using deep-learning algorithm. SPE J. 24 <https://doi.org/10.2118/195698-PA>.
- Li, Y., Sun, R., Horne, R., 2019. Deep learning for well data history analysis. In: Proceedings - SPE Annual Technical Conference and Exhibition. <https://doi.org/10.2118/196011-ms>.
- Lin, Z., Liu, X., Lao, L., Liu, H., 2020. Prediction of two-phase flow patterns in upward inclined pipes via deep learning. Energy 210, 118541. <https://doi.org/10.1016/J.ENERGY.2020.118541>.
- Liu, W., Liu, W.D., Gu, J., 2019. Petroleum production forecasting based on machine learning. In: ACM International Conference Proceeding Series. <https://doi.org/10.1145/3373419.3373421>.
- Ma, X., Liu, Z., 2018. Predicting the oil production using the novel multivariate nonlinear model based on Arps decline model and kernel method. Neural Comput. Appl. 29 <https://doi.org/10.1007/s00521-016-2721-x>.
- Mamudu, A., Khan, F., Zendehboudi, S., Adedigba, S., 2020. Dynamic risk assessment of reservoir production using data-driven probabilistic approach. J. Petrol. Sci. Eng. 184 <https://doi.org/10.1016/j.petrol.2019.106486>.
- Mohaghegh, S.D., 2020. Subsurface analytics: contribution of artificial intelligence and machine learning to reservoir engineering, reservoir modeling, and reservoir management. Petrol. Explor. Dev. [https://doi.org/10.1016/S1876-3804\(20\)60041-6](https://doi.org/10.1016/S1876-3804(20)60041-6).
- Mohaghegh, S.D., 2017. Data-Driven Reservoir Modeling. Society of Petroleum Engineers.
- Mohaghegh, S.D., 2011. Reservoir simulation and modeling based on artificial intelligence and data mining (AI&DM). J. Nat. Gas Sci. Eng. <https://doi.org/10.1016/j.jngse.2011.08.003>.
- Muojeke, S., Venkatesan, R., Khan, F., 2020. Supervised data-driven approach to early kick detection during drilling operation. J. Petrol. Sci. Eng. 192 <https://doi.org/10.1016/j.petrol.2020.107324>.
- Nait Amar, M., 2020. Modeling solubility of sulfur in pure hydrogen sulfide and sour gas mixtures using rigorous machine learning methods. Int. J. Hydrogen Energy 45, 33274–33287. <https://doi.org/10.1016/j.ijhydene.2020.09.145>.
- Nait Amar, M., Jahanbani Ghahfarokhi, A., 2020. Prediction of CO2 diffusivity in brine using white-box machine learning. J. Petrol. Sci. Eng. <https://doi.org/10.1016/j.petrol.2020.107037>.
- Nait Amar, M., Jahanbani Ghahfarokhi, A., Ng, C.S.W., Zeraibi, N., 2021. Optimization of WAG in real geological field using rigorous soft computing techniques and nature-inspired algorithms. J. Petrol. Sci. Eng. 109038. <https://doi.org/10.1016/j.petrol.2021.109038>.
- Nait Amar, M., Jahanbani Ghahfarokhi, A., Zeraibi, N., 2020a. Predicting thermal conductivity of carbon dioxide using group of data-driven models. J. Taiwan Inst. Chem. Eng. <https://doi.org/10.1016/j.jtice.2020.08.001>.
- Nait Amar, M., Zeraibi, N., 2020. A combined support vector regression with firefly algorithm for prediction of bottom hole pressure. SN Appl. Sci. 2 <https://doi.org/10.1007/s42452-019-1835-z>.
- Nait Amar, M., Zeraibi, N., 2019. An efficient methodology for multi-objective optimization of water alternating CO2 EOR process. J. Taiwan Inst. Chem. Eng. <https://doi.org/10.1016/j.jtice.2019.03.016>.
- Nait Amar, M., Zeraibi, N., 2018. Application of hybrid support vector regression artificial bee colony for prediction of MMP in CO2-EOR process. Petroleum. <https://doi.org/10.1016/j.petlm.2018.08.001>.
- Nait Amar, M., Zeraibi, N., Jahanbani Ghahfarokhi, A., 2020b. Applying hybrid support vector regression and genetic algorithm to water alternating CO2 gas EOR. Greenh. Gases Sci. Technol. <https://doi.org/10.1002/ghg.1982>.
- Nait Amar, M., Zeraibi, N., Redouane, K., 2018. Bottom hole pressure estimation using hybridization neural networks and grey wolves optimization. Petroleum. <https://doi.org/10.1016/j.petlm.2018.03.013>.
- Ng, C.S.W., Jahanbani Ghahfarokhi, A., Nait Amar, M., 2021a. Application of nature-inspired algorithms and artificial neural network in waterflooding well control optimization. J. Pet. Explor. Prod. Technol. <https://doi.org/10.1007/s13202-021-01199-x>.
- Ng, C.S.W., Jahanbani Ghahfarokhi, A., Nait Amar, M., Torsæter, O., 2021b. Smart proxy modeling of a fractured reservoir model for production optimization: implementation of metaheuristic algorithm and probabilistic application. Nat. Resour. Res. 30, 2431–2462. <https://doi.org/10.1007/s11053-021-09844-2>.
- Olukoga, T.A., Feng, Y., 2021. Practical machine-learning applications in well-drilling operations. SPE Drill. Complet. <https://doi.org/10.2118/205480-pa>.
- Ozbayoglu, E., Ozbayoglu, M., Ozdilli, B.G., Ergo, O., 2021. Optimization of flow rate and pipe rotation speed considering effective cuttings transport using data-driven models. Energies 14. <https://doi.org/10.3390/en14051484>.
- Panja, P., Velasco, R., Pathak, M., Deo, M., 2018. Application of artificial intelligence to forecast hydrocarbon production from shales. Petroleum. <https://doi.org/10.1016/j.petlm.2017.11.003>.
- Qu, H., Zhang, Y., 2016. A new kernel of support vector regression for forecasting high-frequency stock returns, 2016 Math. Probl. Eng. <https://doi.org/10.1155/2016/4907654>.
- Senthilkumar, M., 2010. Use of artificial neural networks (ANNs) in colour measurement. In: Colour Measurement: Principles, Advances and Industrial Applications. <https://doi.org/10.1533/9780857090195.1.125>.
- Shawe-Taylor, J., Cristianini, N., 2004. Kernel Methods for Pattern Analysis, Kernel Methods for Pattern Analysis. <https://doi.org/10.1017/cbo9780511809682>.
- Shi, Y., Eberhart, R., 1998. Modified particle swarm optimizer. In: Proceedings of the IEEE Conference on Evolutionary Computation. ICEC. <https://doi.org/10.1109/icec.1998.699146>.
- Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems.

- Syed, F.I., Alshamsi, M., Dahaghi, A.K., Neghabhan, S., 2020. Artificial lift system optimization using machine learning applications. *Petroleum*. <https://doi.org/10.1016/J.PETLM.2020.08.003>.
- Vapnik, V.N., 1995. The Nature of Statistical Learning Theory, the Nature of Statistical Learning Theory. <https://doi.org/10.1007/978-1-4757-2440-0>.
- Zanjani, M.S., Salam, M.A., Kandara, O., 2020. Data-driven hydrocarbon production forecasting using machine learning techniques. *Int. J. Comput. Sci. Inf. Secur.* **18**.
- Zhan, C., Sankaran, S., LeMoine, V., Graybill, J., Mey, D.O.S., 2020. Application of machine learning for production forecasting for unconventional resources. In: SPE/AAPG/SEG Unconventional Resources Technology Conference 2020. <https://doi.org/10.15530/urtec-2019-47>. URTeC 2020.
- Zhang, J.S., Xiao, X.C., 2000. Predicting chaotic time series using recurrent neural network. *Chin. Phys. Lett.* **17** <https://doi.org/10.1088/0256-307X/17/2/004>.
- Zhong, Z., Sun, A.Y., Wang, Y., Ren, B., 2020. Predicting field production rates for waterflooding using a machine learning-based proxy model. *J. Petrol. Sci. Eng.* **194**, 107574. <https://doi.org/10.1016/j.petrol.2020.107574>.