Predictive Model for Bottomhole Pressure based on Machine Learning

Pavel Spesivtsev[a,*], Konstantin Sinkov[a], Ivan Sofronov[a], Anna Zimina[a,b], Alexey Umnov[c], Ramil Yarullin[c], Dmitry Vetrov[c]

[a]*Schlumberger Moscow Research Center, 13 Pudovkina str., Moscow, 119285, Russian Federation*
[b]*Moscow Institute of Physics and Technology, 9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russian Federation*
[c]*National Research University Higher School of Economics, 20 Myasnitskaya str., Moscow, 101000, Russian Federation*

Abstract

The objective of this work is to develop a predictive model for multiphase wellbore flows using the machine learning approach. The artificial neural network is developed and then trained on the dataset generated using the numerical simulator of the full-scale transient wellbore flows. After the training is completed, the neural network is used to predict one of the key parameters of the wellbore flow, namely, the bottomhole pressure. The novelty of this work is related to the application of the neural network to analyze highly transient processes taking place in wellbores. In such processes, most of the parameters of interest can be represented by interdependent time series of variables linked through complex physical phenomena pertinent to the nature of multiphase flows. The proposed neural network with two hidden layers demonstrated the capability to predict the bottomhole pressure within 5% of the normalized root mean squared error for many complex wellbore configurations and flows. It is also shown that relatively higher prediction errors are mainly observed in the case of slug flows where the transient nature of flows is pronounced the most. Finally, the developed model is tested on data affected by noise. It is demonstrated that although the error of prediction slightly increases in contrast to the data without noise, the model captures essential features of the studied transient process. Description of the developed models, analysis of various test use cases, and possible future research directions are outlined.

Keywords: Multi-phase flow, machine learning, bottomhole, pressure, prediction

1. Introduction

The transient multiphase wellbore flow is a challenging physical problem. In the oil and gas industry, such flows can occur during the well startup operation when the well is open to flow at surface for the first time. In this process, the fluids used to drill and complete the well are being displaced by the hydrocarbons entering the well from the formation (Theuveny et al., 2013). The displacement of one fluid by another having different properties, mass exchange (e.g., release of gas from oil), manipulations with the surface choke, and other processes lead to changes in the distribution of pressure, volume fractions, and velocities in the wellbore in space and time. Hence, this process is always transient. Depending on the wellbore configuration and properties of fluids, the slug flow may be formed. In this case, the variations of wellbore flow parameters can be observed even for steady-state boundary conditions such as constant pressure at the wellbore outlet. Note that in this work, only the severe slugging process is investigated. The smaller-scale hydrodynamic slugging processes (see, for example, Barnea and Taitel, 1994) are beyond the scope of this work. The large-scale severe slugging is usually induced by the pipeline geometry (e.g., L-shaped) and by certain inflow rates that lead to the blocking of significant and extended gas volume fractions by liquid followed by the period of usually rapid outflow from the wellbore or pipeline. Such processes are often close to periodical ones.

The conventional approach to investigating the transient multiphase wellbore flows solidly established over last decades is to use a simulator based on conservation laws of fluid mechanics and numerical methods (e.g., Spesivtsev et al., 2017). In particular, the numerical simulator can be used to determine the unknown bottomhole pressure (BHP) for specific boundary conditions, such as wellhead pressure (WHP) or oil, water, and gas surface rates (often denoted as Q_o, Q_w, and Q_g, respectively). This problem is shown schematically in Fig. 1. In this work, an alternative approach based on machine learning techniques to study the aforementioned problem of wellbore flow with strong multiphase and transient effects is presented.

Machine learning approaches are extensively used in many applications (Jordan and Mitchell, 2015). The approach is found to be particularly useful in processing large amounts of data (Sagiroglu and Sinanc, 2013; Al-Jarrah

*Corresponding author
Email addresses: pspesivtsev@slb.com (Pavel Spesivtsev), ksinkov@slb.com (Konstantin Sinkov), isofronov@slb.com (Ivan Sofronov), azimina@slb.com (Anna Zimina), aumnov@hse.ru (Alexey Umnov), ramly@ya.ru (Ramil Yarullin), dvetrov@hse.ru (Dmitry Vetrov)

Nomenclature

BHP(t)	Bottomhole pressure, bar
MD(i)	Segment measured depth, m
MD$_s$(i)	Source measured depth, m
relu(x)	ReLU activation function
TVD(i)	Segment true vertical depth, m
WHP(t)	Wellhead pressure, bar
$\sigma(x)$	Sigmoid activation function
A	Randomly generated parameter of source flow rate, h
A_i	Randomly generated parameter of wellhead pressure, bar/h^2
B	Randomly generated parameter of source flow rate, h^2
B_i	Randomly generated parameter of wellhead pressure, bar/h
C	Randomly generated parameter of source flow rate, m^3/d
C_i	Randomly generated parameter of wellhead pressure, bar
D	Number of ANN inputs
$d(i)$	Pipe diameter, m
$f_a(t)$	Actual time series
$f_e(t)$	Estimated time series
$h(x)$	Activation function
i	Integer index
K	Number of ANN outputs
M	Number of ANN hidden units
o, g, w	Subscripts, related to oil, gas, and water
$Q(t)$	Surface flow rate, m^3/d
$q(t, i)$	Source flow rate, m^3/d
T	Time series length
t	Time, h
t_i	Randomly generated parameter of wellhead pressure, h
w_{ij}	ANN weights
x_j	ANN inputs
y_k	ANN outputs
z_i	ANN hidden units

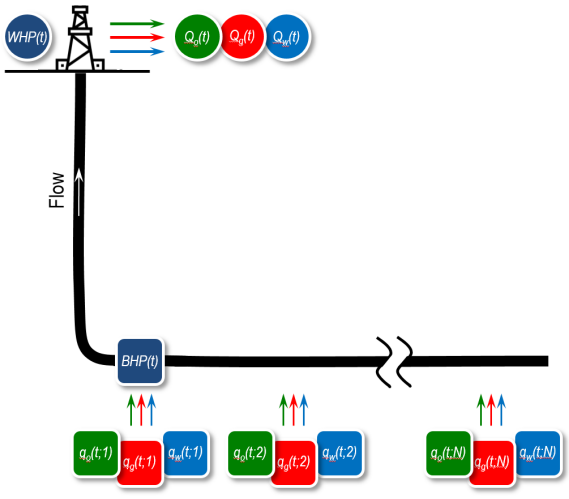


Figure 1: Schematics of the transient multiphase wellbore flow problem

et al., 2015) and addressing the problems with data that are similar to those in the reservoir modeling (single or multichannel real values over a time span). Also, there are also many recent advances in time series forecasting and analysis (Bontempi et al., 2013; Malhotra et al., 2015), speech recognition (Hinton et al., 2012) and medical data classification (Jin et al., 2009).

In research and development related to the oil and gas industry, data analytics and machine learning approaches receive increasing amount of attention. Recently, the approaches based on artificial neural networks (ANN) and least squares support vector machines (LS-SVM) are applied to solve a wide range of relevant problems. In particular, the data-driven models are proposed to describe the pseudo skin factor affecting the productivity of horizontal wells (see (Ahmadi et al., 2015) and references therein for more applications). Several works pose the problem of creating a surrogate model for reservoir modeling (Mohaghegh, 2011) and in other areas (Koziel and Leifsson, 2013; Ong et al., 2003). The surrogate model is developed using an existing numerical simulation model: the extensive dataset with simulation scenarios is created and the machine learning algorithm is trained on this dataset. The main need in such models is based on the fact that the numerical simulators usually require considerable amount of resources to carry out simulations; at the same time, the trained model can be less resource-consuming while giving reasonably good approximation results. Another problem is the reservoir behavior prediction using historical data (Mohaghegh, 2011; Zhao et al., 2006; Rabiei et al., 2010). The machine learning model is trained on the set of previous observations and can be used to make the predic-

tions or analytics for the future instances in the considered problem.

Use of data analytics and machine learning to study the processes taking place in wellbores is a relatively new area. In Tian and Horne (2015), the machine learning techniques are applied to interpret the pressure, temperature, and flow rates recorded by the downhole gauges. In Song et al. (2016) a backpropagation ANN was applied to analyze the velocity field during wellbore cleanout operations. However, there is not much work that considers the transient problem of large-scale multiphase wellbore flows (including slug flows) using data analytics and machine learning techniques. In Osman et al. (2005), Jahanandish et al. (2011), Ashena and Moghadasi (2011), and Ebrahimi and Khamehchi (2015) ANNs were used to predict BHP in wells operated under conditions of multiphase flow. In Osman et al. (2005) an ANN was applied to estimate the pressure drop in vertical wells. Performance of the model trained on real field data was compared against empirical correlations available in the literature. It was demonstrated that for the data studied the ANN produces the lowest errors. Trend analysis of the model showed continuous and predictable effect of the input variables on the bottomhole pressure estimations. These observations were later confirmed using different network architectures, optimization algorithms, and field datasets (Jahanandish et al., 2011; Ebrahimi and Khamehchi, 2015). An ANN-based model predicting the pressure drop in the annulus was developed in Ashena and Moghadasi (2011); several algorithms were utilized to adjust weights and optimize architecture of the ANN.

The steady-state problem of BHP evaluation using the machine learning techniques was addressed in (Osman et al., 2005; Jahanandish et al., 2011; Ashena and Moghadasi, 2011; Ebrahimi and Khamehchi, 2015). The authors assumed that the BHP is a single-valued characteristic of a particular well that depends on WHP and surface flow rates that are, in turn, also single-valued characteristics of this particular well. It was assumed that the well parameters are steady-state. However, the BHP is substantially affected by physical processes taking place along the wellbore from wellhead to perforations. Phase transitions and flow pattern changes occurring on as hydrocarbons flow from the formation to the surface define local pressure drops and, consequently, the BHP. The instant value of the BHP is a function of the entire wellbore state, including the distribution of phases that may vary with depth. Because we focus on the essentially transient well startup process, the distribution of phases changes with time. This distribution is affected not only by the current values of the inflow from the reservoir and the WHP, but by the states of wellbore at the previous moments of time. Hence, the instant value of BHP is history dependent. Moreover, the flow rates observed at the surface may significantly differ from the rates entering the well from the reservoir at the same time instant due to accumulation and redistribution of phases within wellbore. In other words, measuring the WHP, surface flow rates, and other parameters at the current moment only is not sufficient to predict the BHP at this same moment of time, and this problem cannot be simply reduced to the sequence of predictions of the BHP for different steady states of the well. In contrast to previous studies, we address specifically the transient problem of predicting the time-dependent BHP. To predict this characteristic, we must use the history of WHP and surface rates, which are also represented as the time-dependent variables. As it is demonstrated below, one sample of training set for the machine learning algorithm is represented as a set of time series, not just a set of single-valued numbers.

2. Problem formulation

In the oil and gas industry, deep wells are drilled to reach the hydrocarbon-bearing formations. The wells can be as deep as several thousands of meters and more. At such depth, the pressures in the formation and in the wellbore can be as high as several hundreds of bars. In some of the wells, downhole gauges are deployed to record the data. Because deploying gauges is an expensive procedure, in a significant number of wells these data is not available. However, knowledge of these parameters is important to manage the well startup after completion and production effectively and safely. Hence, determining parameters at the well bottom is a challenging and important task. In this work, it is proposed to use the machine learning approach and construct a neural network that can be trained on the available data; the network then will be used to provide predictions of the transient BHP.

To generate the dataset that could be used for training and for validation, a transient multiphase wellbore flow simulator is used (Spesivtsev et al., 2017). The simulator captures the following major hydrodynamic phenomena affecting the transient behavior of the BHP during well startup and initial production:

- Dynamics of volume fraction distributions of phase in the wellbore
- Velocity difference between phases (also referred to as slip)
- Compressibility of phases
- Release of gas in the wellbore as pressure drops below the bubblepoint value
- Friction

In the cases presented in this work, all these effects are simulated. As described below, the simulator input parameters are chosen to be representative of real transient field data.

2.1. Wellbore flow description

The behavior of a well is defined by the following set of principal parameters (see also Fig. 1).

- **Well geometry.** The well is represented by a set of segments. Each segment i is described by three principal numbers: measured depth $\mathrm{MD}(i)$, true vertical depth $\mathrm{TVD}(i)$, and diameter $d(i)$.
- **Inflow.** Several volumetric source terms (also referred to as sources) are placed along the well at different points. The source term is a simplified representation of a perforation where the fluids from the underground formation enter the wellbore. Each source i is described by its position expressed in terms of measured depth $\mathrm{MD}_s(i)$ and oil flow rate time series $q_o(t;i)$, water flow rate time series $q_w(t;i)$, and gas flow rate time series $q_g(t;i)$. The sources are sorted by their depth, so the source with $i = 1$ is the closest to the surface.
- **Wellbore control.** The wellbore is controlled using the wellhead pressure time series $\mathrm{WHP}(t)$.

Other parameters are not listed here because they are either kept constant during all the experiments or are not relevant to the considered problem formulation (e.g., roughness of pipe walls). Using this list of parameters, the simulator models the transient multiphase wellbore flow over a given duration and provides the results in terms of the following time series:

- $\mathrm{BHP}(t)$ — BHP (measured at the position of the source closest to the surface)
- $Q_o(t)$ — surface oil volumetric flow rate at standard conditions
- $Q_w(t)$ — surface water volumetric flow rate at standard conditions
- $Q_g(t)$ — surface gas volumetric flow rate at standard conditions

Often, in the real field situation, the available data are the parameters known before the well is started to flow and the time series that can be measured at the surface during the flow. Therefore, we take the following list of input parameters for our machine learning model: the $\mathrm{WHP}(t)$ and rates ($Q_o(t)$, $Q_w(t)$, $Q_g(t)$), the well geometry ($\mathrm{MD}(i)$, $\mathrm{TVD}(i)$, $d(i)$), and the measured depth of the source closest to the surface that usually coincides with the position of the topmost perforation ($\mathrm{MD}_s(1)$). The output of the model is the $\mathrm{BHP}(t)$, important and often unknown wellbore flow characteristic.

2.2. Dataset generation

Massive simulations with different samples are carried out to form data suitable for training the machine learning algorithm. The input wellbore flow parameters are varied in a unique (random) way for each sample. The variation is carried out as follows:

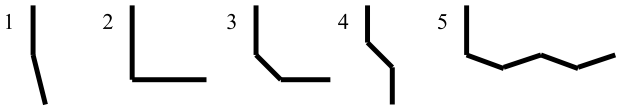


Figure 2: Schematic representation of the five types of geometry.

- Ten different geometry parameters are used; for each sample, the geometry is picked randomly. There are five types of well trajectories (see Fig. 2) with different diameter parameters; see the full geometries description in Table A.1.
- There are six sources with positions generated randomly for each sample. To avoid positioning of sources too close to the surface, the sources are allowed to be located starting from the second wellbore segment. There is no limitation imposed on spacing between sources.
- For each source, the $q_w(t)$ and $q_o(t)$ are given as

$$
\begin{aligned}
q_w(t) &= C_w e^{-\frac{(t-A)^2}{B}}, \\
q_o(t) &= \begin{cases} 0, & \text{when } t < A; \\ C_o\left(1 - e^{-\frac{(t-A)^2}{B}}\right), & \text{when } t \geqslant A. \end{cases}
\end{aligned}
$$

  The parameters A and B of these functions are shared between water and oil sources; the amplitudes of water and oil sources C_w and C_o are chosen independently. All parameters are generated randomly (affecting amplitude, time of maximum water flow rate, and time of oil rate buildup). An example of the sources functions is shown in Fig. 3.
- The $q_g(t)$ is fixed to zero in the current study. However, oil contains dissolved gas so that the resulting gas rate at the outlet $Q_g(t)$ is non-zero.
- The $\mathrm{WHP}(t)$ is generated as a random piecewise constant/linear/quadratic functions with parameters t_i, A_i, B_i, and C_i, $i = \overline{0,k}$ defined as

$$
\mathrm{WHP}(t) = \begin{cases} A_0 t^2 + B_0 t + C_0, & \text{when } t < t_1; \\ A_1 t^2 + B_1 t + C_1, & \text{when } t_1 \leqslant t < t_2; \\ \dots, & \\ A_k t^2 + B_k t + C_k, & \text{when } t_k \leqslant t. \end{cases}
$$

  In the majority of cases, the parameters were chosen so that piecewise constant and linear functions are frequently generated. Also, no specific conditions are imposed on the continuity and monotonicity of these functions.

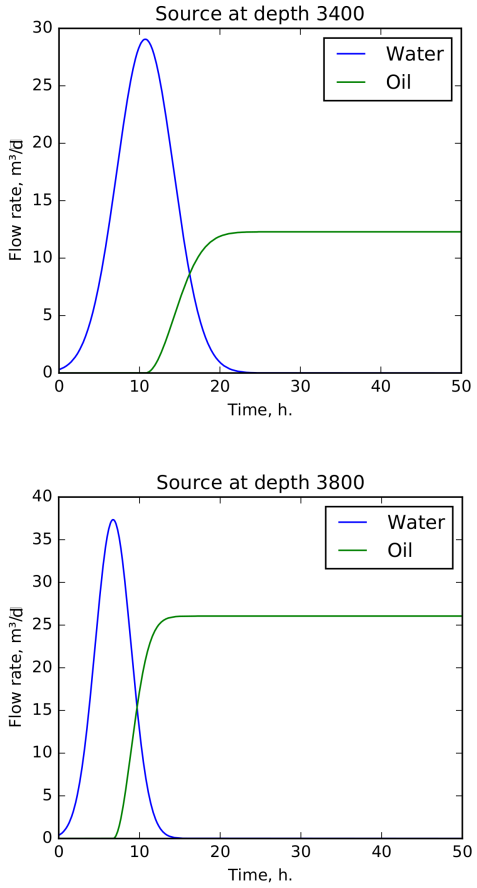


Figure 3: Examples of the source functions for a selected scenario. Only two of six sources are shown for brevity.

An example of one sample with graphs of WHP(t) and BHP(t) functions and surface flow rates $Q_o(t)$, $Q_w(t)$, and $Q_g(t)$ is shown in Fig. 4. Note that in this figure negative values of surface flow rates can be observed. This is due to the backflow across the outlet boundary of the computation domain of the numerical simulation that can be caused by rapid changes of WHP or may be due to the formation of a slug flow regime in the wellbore. Usually, the periods of backflow at surface are small compared to the overall simulation time and have no significant influence on the overall system behavior. In reality, backflow across the oil and gas well outlet may also exist. However, this backflow is not measured and not recorded because all the flow metering devices (e.g., separators) are installed downstream of the wellhead, at significant distance.

All the time series functions are given in points 0, 0.1, 0.2, . . . , 50 h (total 501 points over 50 h simulated for each sample). The total number of 3500 samples were generated using the simulator consisting of 2000 cases with low inflow rates of sources (and, therefore, pronounced slugging behavior) and 1500 cases with higher inflow rates (hence, almost without slugging behavior).

After the dataset is generated, the samples are randomly mixed to avoid formation of biased patterns. After this, the dataset is split into a set of objects and the correct answers on them. This subset is often referred to as the training set. In our analysis, the training set comprised 80% of samples. The remaining part of samples constitutes the test dataset. Similarly to the training set, the testing dataset consists of the set of objects and the correct answers on them; however, they are not given to the learning algorithm to train on. This set is used to evaluate the quality of the machine learning algorithm. The problem of machine learning on the generated dataset is formulated as follows. Given the time series data at the surface, geometry parameters, and measured depth of the source closest to the surface, predict the BHP(t) time series for the new case that was not a part of the training dataset. The model is assumed to have access to all the input time series values to generate prediction for any output point. In the current formulation, the real-time prediction is not given full consideration and is left for the future work.

2.3. Quality metrics

Several quality metrics are used to compare the actual and the estimated time series. Denote the estimated time series (predicted using the machine learning method) as $f_e(t)$ and the actual time series (obtained with simulator) as $f_a(t)$, where $t = 1, \ldots, T$. The following metrics are introduced (Hyndman and Koehler, 2006): the root mean square error (RMSE)

$$\mathrm{RMSE} = \left(\frac{1}{T} \sum_{t=1}^{T} (f_e(t) - f_a(t))^2 \right)^{\frac{1}{2}},$$

the mean absolute error (MAE)

$$\mathrm{MAE} = \frac{1}{T} \sum_{t=1}^{T} |f_e(t) - f_a(t)|,$$

the maximum (supremum) absolute error (SAE)

$$\mathrm{SAE} = \max_{t=1,\ldots,T} |f_e(t) - f_a(t)|,$$

the mean absolute percentage error (MAPE)

$$\mathrm{MAPE} = \frac{1}{T} \sum_{t=1}^{T} \frac{|f_e(t) - f_a(t)|}{|f_a(t)|},$$

the maximum (supremum) absolute percentage error (SAPE)

$$\mathrm{SAPE} = \max_{t=1,\ldots,T} \frac{|f_e(t) - f_a(t)|}{|f_a(t)|},$$

and the normalized root mean squared error (NRMSE)

$$\mathrm{NRMSE} = \left(\sum_{t=1}^{T} (f_e(t) - f_a(t))^2 \right)^{\frac{1}{2}} \Big/ \left(\sum_{t=1}^{T} (f_a(t))^2 \right)^{\frac{1}{2}}.$$

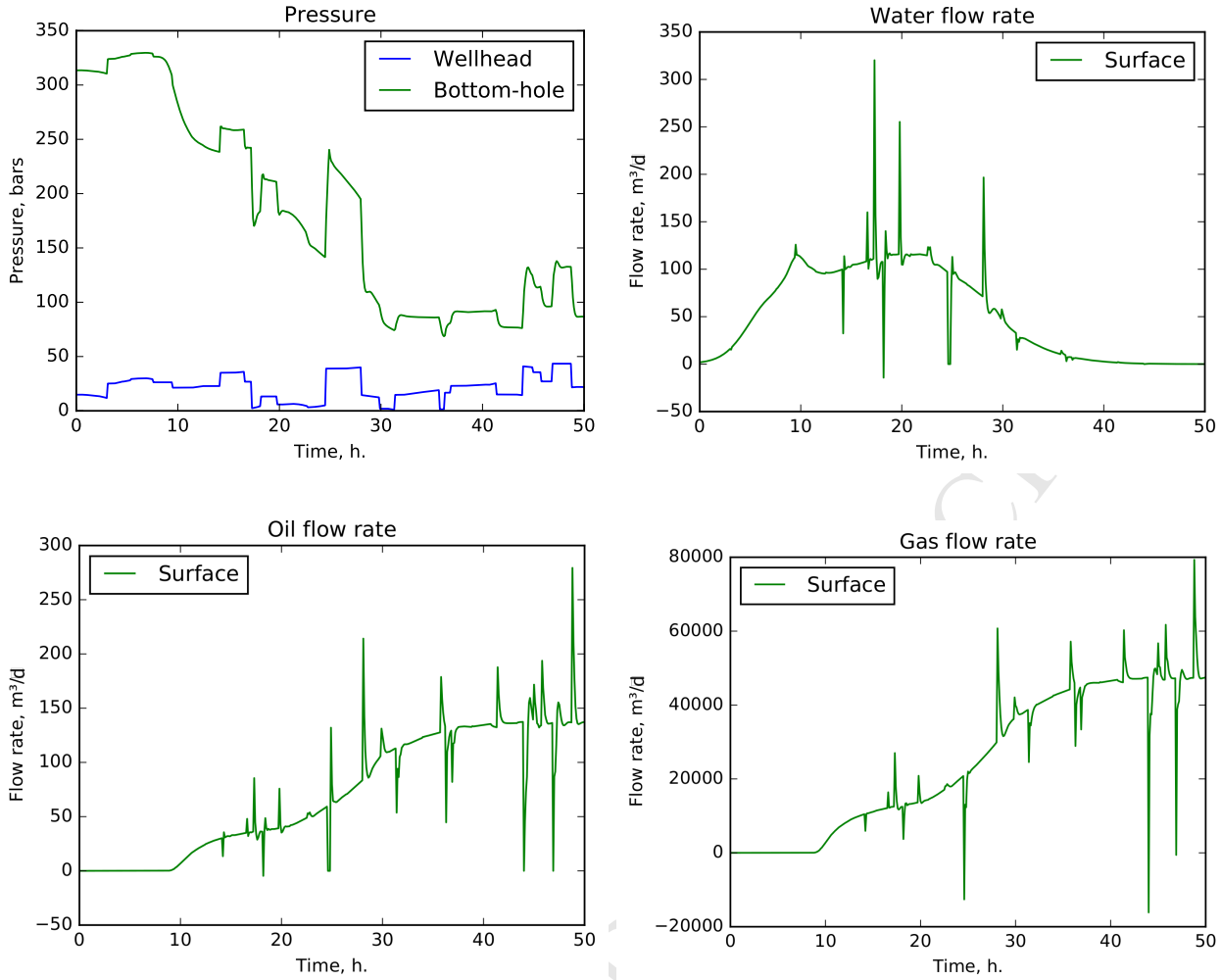


Figure 4: Example of a sample with graphs of WHP and BHP functions and the surface flow rates $Q_o(t)$, $Q_w(t)$, and $Q_g(t)$.

In this work, the NRMSE is considered as the main quality metric and values of this metric below 5% are considered acceptable. Depending on particular operational requirements (e.g., balance between accuracy, complexity of data model architecture, and training time), this threshold can be revised. Both underestimation and overestimation of the BHP in the decision-making process might have unfavorable consequences for the future well production performance. This largely depends on the physical phenomena that are critical for the considered wellbore. That is why in this work no distinction is made with this respect and only the absolute value of error matters. Other metrics values are used for better understanding of the results.

These metrics are aggregated by calculating the average value on the dataset and maximum value on the dataset. The latter is an unstable characteristic, so in the most complex cases, the quantiles may be used instead (for example, 95%-quantile).

3. Development of a predictive algorithm

In our work, we decided to use ANNs (Bishop, 2007, ch. 5) as a predictive model. Artificial neural networks have received a lot of development attention recently and are used in many applications (Jordan and Mitchell, 2015). We chose this model because it is able to predict complex dependencies and is also very flexible, which may be useful in future research.

3.1. Artificial neural networks

The ANN (Bishop, 2007, ch. 5) is defined as an acyclic graph of nodes, where each node transforms its inputs by some differentiable function. For example let us consider a dense network with D inputs $x_1, \ldots, x_D$, M hidden units $z_1, \ldots, z_M$, and K outputs $y_1, \ldots, y_k$. This network is schematically shown in Fig. 5, where $x_0 = 1, z_0 = 1$ are introduced for the shift transformation. The transforma-

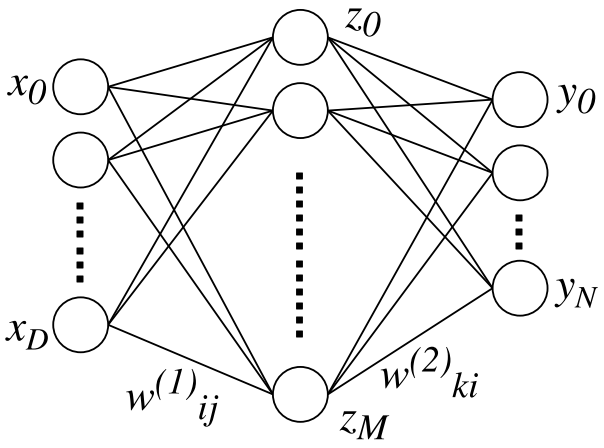


Figure 5: Two-layer dense network.

tions between layers are given by the following formulas:

$$z_i = h\left(\sum_{j=0}^{D} x_j w_{ij}^{(1)}\right), \quad i = 1, \ldots, M;$$

$$y_k = \left(\sum_{i=0}^{M} z_i w_{ki}^{(2)}\right), \quad i = 1, \ldots, K;$$

where $w_{ij}^{(1)}$ and $w_{ki}^{(2)}$ are trainable weights and $h(x)$ is some (smooth) nonlinear function, for example, sigmoid function σ:

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

The network answer is computed by applying all the transformations starting with the input until the output nodes' values are calculated (the feed-forward process). The weights of the network are trained by minimizing a loss function on the training dataset.

One of the commonly used algorithms for the minimization is the stochastic gradient descent (SGD), which approximates the true gradient using small subsets of examples (*mini-batches*). This method is currently considered to be the state-of-the-art optimization algorithm (Goodfellow et al., 2016). The common problem when training the network with SGD is related to the learning rate. The learning rate is the parameter used in ANN learning algorithms to affect the speed of learning (see Bishop, 2007, ch. 3). In SGD, the learning rate is typically hard to tune, and it also needs to be adjusted over training time. In this work, we used the adaptive modification of the SGD — AdaDelta (Zeiler, 2012). The optimization is improved by the following approaches: 1) the learning rate automatically decreases over time; 2) the gradient is corrected using second-order approximations (the true second-order optimization is usually impossible due to high computational costs); 3) the gradient accumulation (momentum) is used to accelerate the convergence. We also found that using the rectified linear $\mathrm{relu}(x)$ activation function helps produce better results than the sigmoid function $\sigma(x)$ (see Nair and Hinton (2010)):

$$\mathrm{relu}(x) = \begin{cases} x, & \text{if } x > 0; \\ 0 & \text{otherwise.} \end{cases}$$

In contrast to other activation functions, $\mathrm{relu}(x)$ is free from the nondimensionalization issues.

To improve the network convergence, the batch normalization have been used (Ioffe and Szegedy (2015)). The additional transformation is added between the linear transformation and the activation function that standardizes the input of the activation function by subtracting the mean value from it and dividing it by the variance. Mean and variance values are estimated using all samples in the mini-batch. Note that the similar scaling can be applied to the input features, which improves the convergence, too (however, the mean and the variance can be calculated more reliably on the full dataset).

3.2. Features and network architecture

The performance of the neural network depends significantly on the proper choice of the input features. In particular, it is important to include the most relevant features in the analysis. Hence, in complex problems, it can be reasonable to provide the neural network with a wider range of possible features to avoid the risk of losing important input information. To characterize the wellbore configuration, the information about the geometry and source depth were selected as features as follows:

1. MD and TVD of the source closest to the surface. For example, in Fig. 1 this is the coordinates of the source $q_o(t; 1)$
2. MD and TVD for the segments closest to the source. Let i be the index of the segment in which the source is located, then the added values are the following: $\mathrm{MD}(i)$, $\mathrm{TVD}(i)$, $\mathrm{MD}(i-1)$, $\mathrm{TVD}(i-1)$
3. MD and TVD for the first segment and last segment and maximums of these values over all segments
4. Diameters for the first and second segments (note that the diameters for other segments are always equal to the diameter of the second)

These features were designed so that they extract as much information as possible from the raw geometry descriptions, and some features may not be particularly useful (for example item 3). However, the neural networks usually can ignore useless features, so this should not be an issue (except for possible additional computational resources for training). We made several experiments showing that removing these features does not improve the result accuracy, although it slightly diminishes the training time due to a simpler neural network structure for the smaller number of input features.

To characterize the parameters available at surface, the time series of $\mathrm{WHP}(t), Q_w(t), Q_o(t), Q_g(t)$ were used as the input features. We found out in the experiments that passing the raw $\mathrm{WHP}(t), Q_w(t), Q_o(t)$, and $Q_g(t)$ values in the limited time window produces good results (the network learns how to combine them itself). However, if the window is big, this produces too many features which results in complex model and low training performance. Thus, the features were reduced using following observations: 1) the WHP values too far from the current value t do not help the prediction; 2) the values of flow rates that are far enough from the current value t can be averaged. The reduced set consists of the following features:

- $\mathrm{WHP}(t+i), i = \overline{-2, 2}$
- $Q_w(t+i), i = \overline{-5, 5}$, similar for oil and gas
- $(Q_w(t+i) + Q_w(t+1+i))/2, i = \overline{6, 12}$, similar for oil and gas
- $(Q_w(t+i) + Q_w(t-1+i))/2, i = \overline{-12, -6}$, similar for oil and gas

All the geometry and flow rates features give, in total, a vector of 76 values.

4. Experiments and results

In this work, we considered ANNs with two hidden layers. As to the size, two models were used:

- The basic model with 100 and 50 hidden units featuring lower training time.
- The complex model with 500 and 100 hidden units that is supposed to provide deeper analysis of the data at the cost of longer training.

The neural network containing 100 and 50 hidden units was chosen as a starting point of the analysis to demonstrate that the proposed approach is fundamentally capable of addressing the problem of highly transient wellbore flows. Subsequently, the larger model containing 500 and 100 hidden units was chosen to evaluate whether increasing the number of units leads to noticeable prediction quality increase or not. Evidently, in this problem certain number of hyperparameters can be introduced (e.g., learning rate, number of iterations, number of layers, number of units, etc.). However, the fine-tuning of the algorithm using hyperparameters is beyond the scope of this work. Hence, no sensitivity analysis to these parameters was carried out. As it is demonstrated below, the two proposed architectures provide a reasonable evaluation of capabilities of the presented algorithm.

4.1. Analysis of the entire dataset

As the first step, the mode with 100 and 50 hidden units was applied to the whole dataset. The metrics values for it are shown in Table 1, the histogram for NRMSE is shown in Fig. 6, and prediction examples are shown in Fig. 8. We also show the cumulative distribution function in Fig. 7 for more thorough analysis. In the rest of experiments, the cumulative distribution functions and histograms appeared to be essentially similar. For the sake of brevity, they are omitted in the remainder of the text. As can be seen, the mean NRMSE is below 5%, indicating that overall, the model is capable of predicting the BHP behavior for the samples that were not used for training. Nevertheless, it can be seen that the model still struggles to predict particular samples. Therefore, it is interesting to identify the samples that are particularly challenging for the algorithm to predict.

Metric	Mean	95%-quantile	Max
RMSE	5.6	15	33
MAE	3.4	7.8	18
SAE	30	80	190
MAPE	3.4%	9.3%	22%
SAPE	44%	130%	270%
NRMSE	**3.0%**	**6.8%**	**16%**

Table 1: Metrics values for the prediction using the basic model.

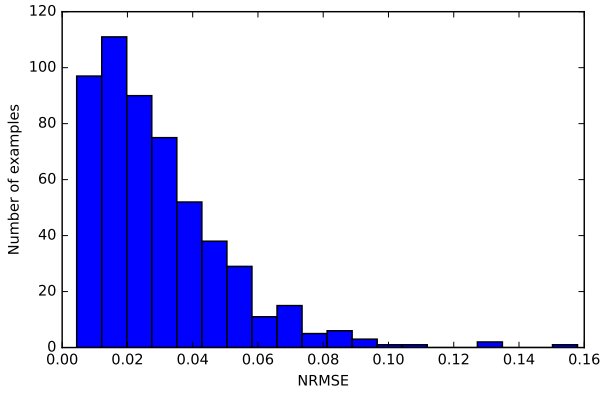


Figure 6: Histogram for NRMSE for the basic model.

4.2. Modification of the dataset

To identify the samples causing the highest errors, the filter based on the flow rate functions was applied to the dataset. We also observed that not all samples with a slug flow regime cause high prediction errors; the problem is mostly with the samples with rapid changes of the flow rate.

First, let us see if removing the most abnormal cases helps the model. To do that, a filter was developed, which removes the sample according to one of the following criteria:

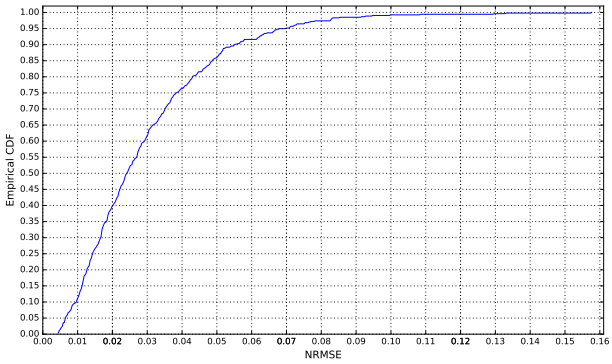


Figure 7: Cumulative distribution function for NRMSE for the basic model.

- The rates were negative for more than 20% of time
- The maximum absolute value of negative oil rate was higher than 10% of the average oil rate (the latter computed over the entire dataset)

This filter is referred to as the weak filter. The weak filter removed approximately 25% of the samples. The metrics values for the model with 100 and 50 hidden units are shown in Table 2, and prediction examples are shown in Fig. 9.

Metric	Mean	95%-quantile	Max
RMSE	5.6	17	60
MAE	3.5	9.8	31
SAE	29	90	280
MAPE	3.4%	9.5%	41%
SAPE	44%	160%	530%
NRMSE	**2.9%**	**7.4%**	**21%**

Table 2: Metrics values for the basic model with weak filter for data.

4.3. Investigating the geometry influence

Next, we noticed that the errors strongly depend on the geometry, and the geometries of types 2 and 5 are more prone to higher errors. Nevertheless, there is a reason to believe that the model can learn to predict answers for different geometries (which is the main task for this step), and the high error is produced not due to the geometry variations but due to too complex regime that can happen in any geometry. To support this statement, we trained a model on the part of the dataset that consists only of geometries 1, 3, and 4 (with weak filtering). The metrics values for this model are shown in Table 3 and the predictions examples are shown in Fig. 10.

It can be seen that there are still examples with slug flow regime in the dataset, and they are hard for the model to work with; however, it works much better with the maximum error close to the desired threshold of 5%. Note that here the model with 100 and 50 hidden units was trained.

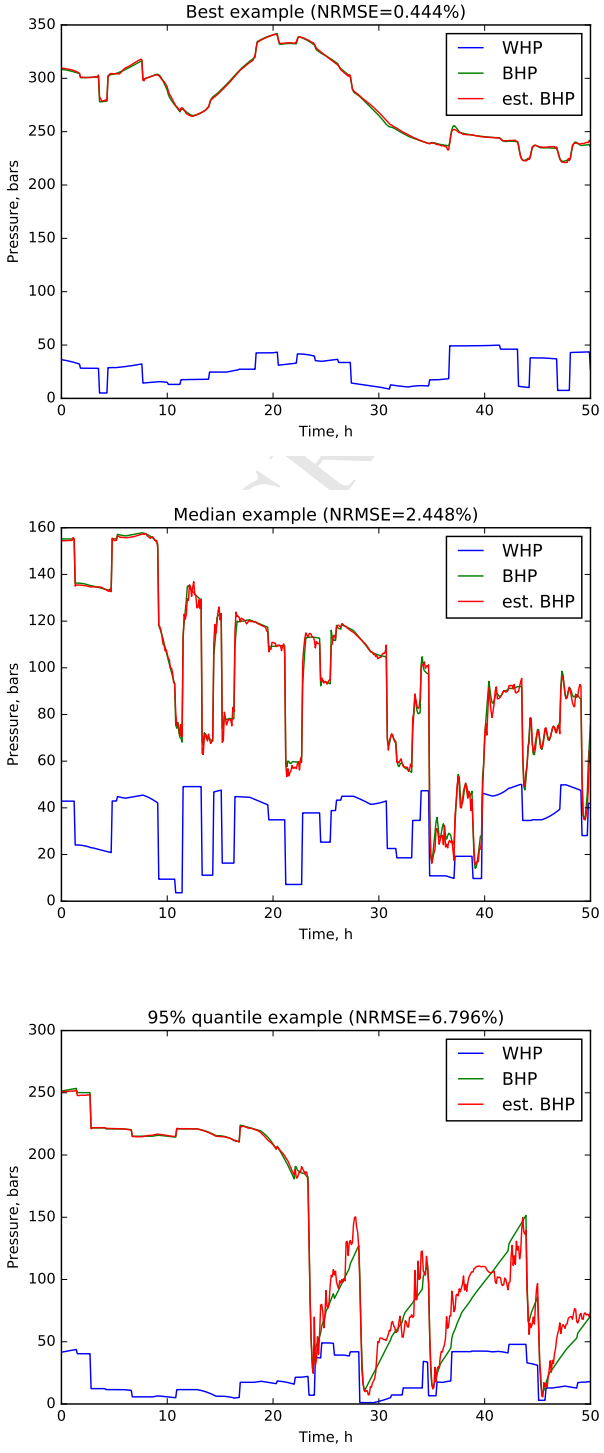
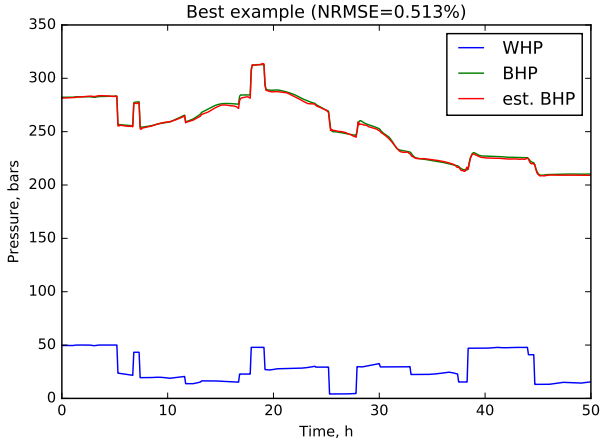
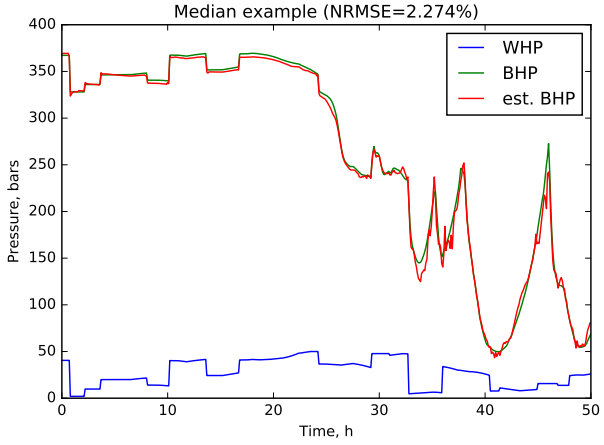


Figure 8: Prediction examples obtained using the basic model: blue curve is the WHP, green curve is the simulated BHP, and the red curve is the BHP estimated by the neural network (est. BHP).

We also found that one particular geometry of type 3a (see also details in Table A.1) was producing the samples with the highest error in the last experiment. After we additionally removed it from the dataset, the results im-





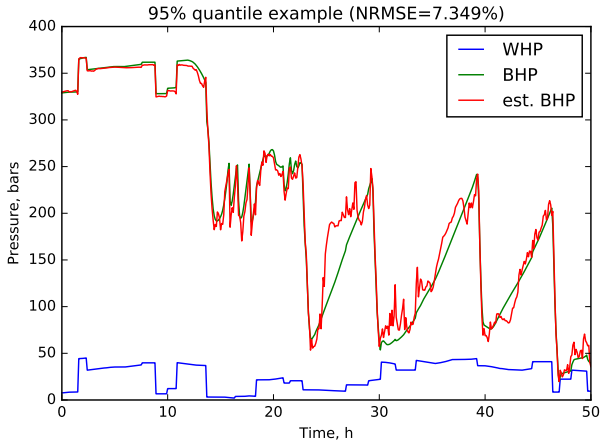


Figure 9: Prediction examples for the basic model with weak filter for data: blue curve is the WHP, green curve is the simulated BHP, and the red curve is the BHP estimated by the neural network (est. BHP).

proved; see Table 4 and prediction examples in Fig. 11. In this case, sometimes high errors were produced due to unstable values at the very end of the timeline (the 5 to 10 last points). In these points, the features values are unreliable because they use extrapolated values, so we decided to remove the 10 last points when calculating metrics. We notice that other experiments did not suffer from this issue, possibly perhaps the datasets were larger and the model learned to process these points separately. This approach of removing the last 10 samples is used in the further experiments with the smaller dataset size.

It can be seen that the result improved; however, it is unclear why this geometry is producing these challenging samples. Also note that in this dataset there were only approximately 200 samples for each geometry.

Metric	Mean	95%-quantile	Max
RMSE	2.1	5.8	18
MAE	1.3	3.0	9.8
SAE	13	39	100
MAPE	1.4%	3.6%	7.9%
SAPE	22%	74%	260%
NRMSE	**1.4%**	**3.1%**	**7.8%**

Table 3: Metrics values for the prediction model for data with geometries 1, 3, and 4.

Metric	Mean	95%-quantile	Max
RMSE	1.7	3.9	6.2
MAE	1.0	2.3	4.3
SAE	10	26	58
MAPE	1.2%	3.0%	6.6%
SAPE	17%	58%	230%
NRMSE	**1.2%**	**2.9%**	**5.0%**

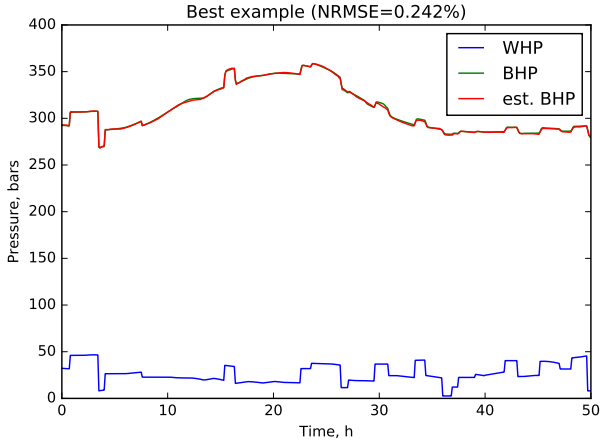
Table 4: Metrics values for the prediction model for data with geometries 1, 3b, and 4.

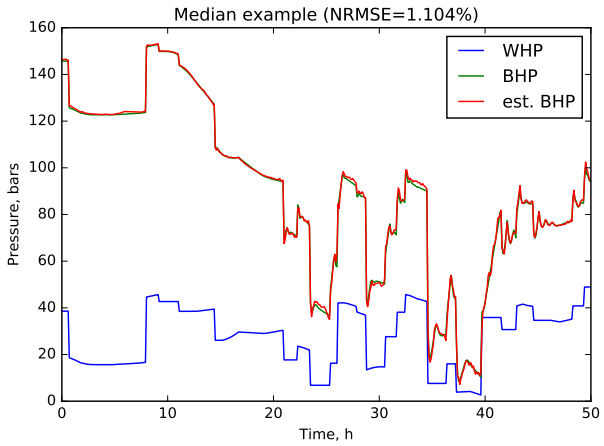
Finally we tested separately the geometry with the highest errors (geometry 5b in Table A.1). The metrics values are shown in Table 5, and the prediction examples are shown in Fig. 12. The complexity of the model was reduced (down to 50 and 25 hidden units) because there were only approximately 100 samples in the training set.

Metric	Mean	95%-quantile	Max
RMSE	11	26	59
MAE	6.8	16	32
SAE	62	170	240
MAPE	5.0%	12%	38%
SAPE	76%	240%	610%
NRMSE	**4.3%**	**10%**	**21%**

Table 5: Metrics values for the prediction model for geometry 5b

It can be seen that the results are much worse for this geometry; also note that the worst sample and the model predictions on it are similar to those in the dataset with full samples (cf. Fig. 9). This confirms that the model learns to differentiate the geometries successfully, and the problem is most likely related to the complexity of the slug flow





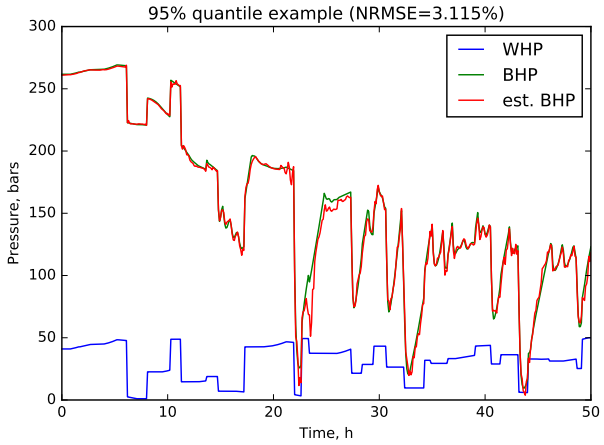
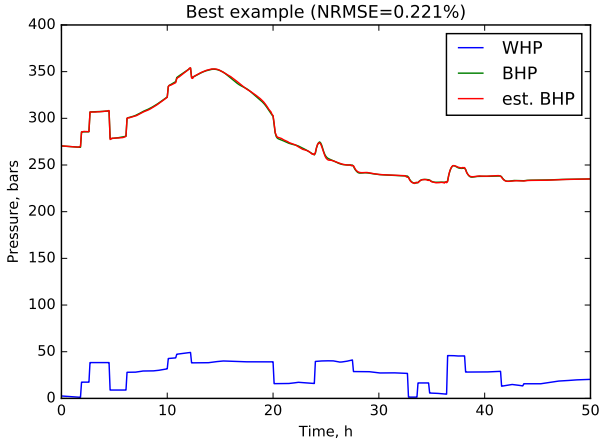
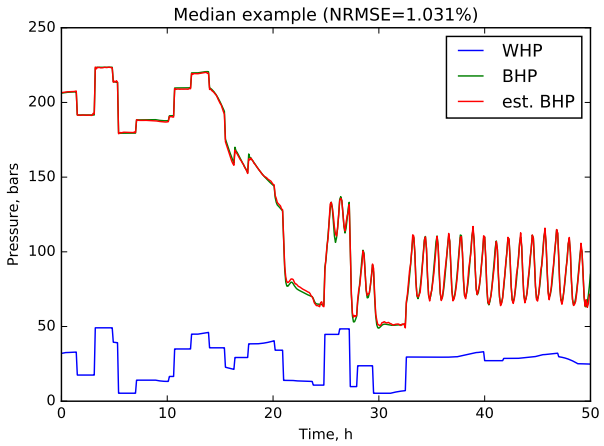


Figure 10: Prediction examples for data with geometries 1, 3, and 4: blue curve is the WHP, green curve is the simulated BHP, and the red curve is the BHP estimated by the neural network (est. BHP).





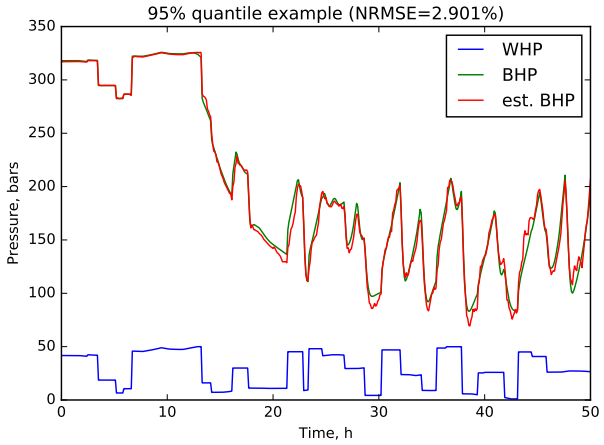


Figure 11: Prediction examples for data with geometries 1, 3b, and 4: blue curve is the WHP, green curve is the simulated BHP, and the red curve is the BHP estimated by the neural network (est. BHP).

regime. Our analysis demonstrates the fundamental possibility to predict the transient bottomhole pressure using the proposed neural network. The detailed investigation of the wellbore geometry influence on the prediction quality is left for future work.

4.4. Increasing the model complexity

The model with 500 and 150 hidden units was also trained on the considered dataset. The metrics for this

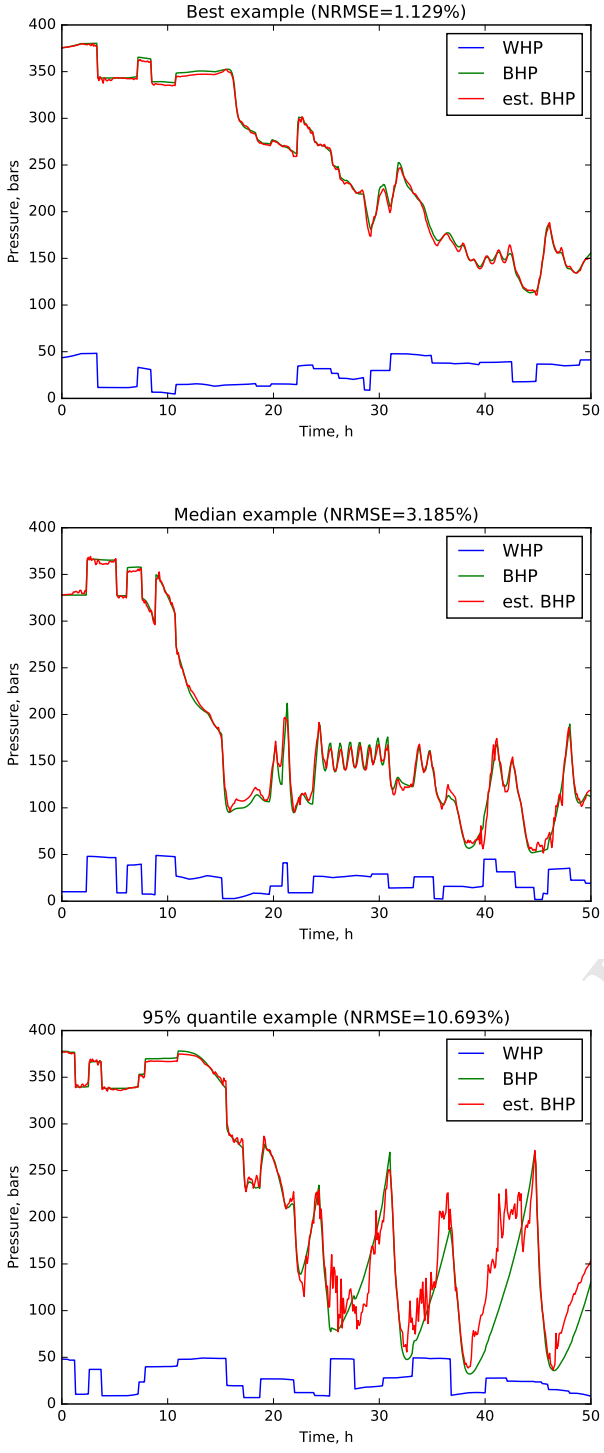


Figure 12: Prediction examples for geometry 5b: blue curve is the WHP, green curve is the simulated BHP, and the red curve is the BHP estimated by the neural network (est. BHP).

model are shown in Table 6, and the histogram and prediction examples are similar to the previous model, so they are not shown here. It turns out that the more complex model provided only slight improvement in accuracy compared to the basic model.

Metric	Mean	95%-quantile	Max
RMSE	4.9	16	64
MAE	2.8	8.1	33
SAE	27	87	270
MAPE	3.0%	9.9%	44%
SAPE	45%	180%	560%
NRMSE	**2.5%**	**7.2%**	**22%**

Table 6: Metrics values for the high-complexity prediction model with weak filter for data.

It can be seen that there are still samples in the dataset that produce high error and are in the slug flow regime. We also found that most of these example were still producing values of the flows with high negative amplitude. At this point, a stricter threshold was introduced to remove samples with the values of gas rate $Q_q < -2.4 \cdot 10^5$ sm^3/day. By applying the strong filter, approximately 10% of samples were removed additionally. The metrics values for model with 500 and 150 hidden units on these data are shown in Table 7, and the examples of the predictions are shown in Fig. 13.

Metric	Mean	95%-quantile	Max
RMSE	4.4	13	38
MAE	2.8	7.6	21
SAE	23	68	150
MAPE	2.8%	8.0%	21%
SAPE	41%	140%	1300%
NRMSE	**2.5%**	**5.8%**	**13%**

Table 7: Metrics values for the complex model with strong filter for data.

So, we can see that gradually removing the samples with complex behavior can help the model to learn better (compare Tables 1, 6, and 7). The high number of parameters is typically associated with the risk of overfitting. Alternatively, additional features to discriminate and describe more complex classes of flow regimes can be added in further improvement of the current model.

4.5. Data with noise

To investigate how the quality of predictions given by the developed model is affected by noise, the modified dataset was prepared. Typically, the wellhead has a hydraulic connection with the separator installed downstream. In the separator, the phases recovered from the wellbore are split and sent further (e.g., to production facilities). The pressure in the separator might oscillate, and the oscillations might propagate backwards and, in turn, affect the WHP. Hence, if the separator pressure is affected by noise, then the WHP may become noisy, too. Assuming this, we regenerated the datasets by running the simulator on the samples with the WHP affected by noise. The noise

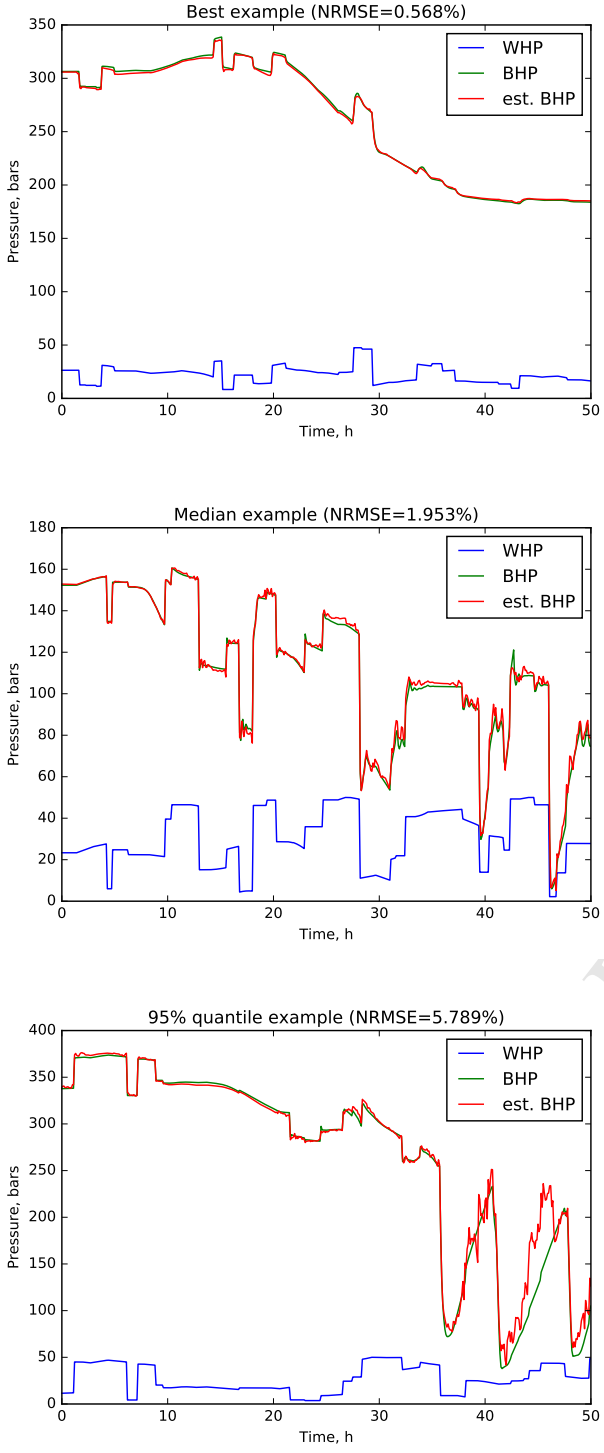


Figure 13: Prediction examples for the complex model with strong filter for data: blue curve is the WHP, green curve is the simulated BHP, and the red curve is the BHP estimated by the neural network (est. BHP).

was added in the following manner. The original WHP generated as described above was modified by adding a random noise:

$$\mathrm{WHP}_{\mathrm{noise}}(t) = \mathrm{WHP}(t)[1 + 0.25 * \mathrm{rand}(-1, 1)],$$

where $\mathrm{rand}(-1, 1)$ is the random value distributed uniformly in the range $[-1, 1]$. After this, the simulator is launched to provide the resulting BHP for the new set of input data containing $\mathrm{WHP}_{\mathrm{noise}}(t)$. The rest of inputs (e.g., sources and well geometry) remain the same.

An example of one sample generated using the noisy data in the input and, as a result of simulation, in the output, is shown in Fig. 14. As one can see, introducing noise in the WHP significantly affected all simulation results (compare with Fig. 4 without noise). The surface rates now are oscillating significantly. The BHP contains oscillations that resulted from the complex interaction of the noise in the input WHP and the transient effects captured by the simulator.

Next, the basic model trained on the whole dataset without noise is applied to the test data set containing noise. Note, that we paid special attention to make sure that the original training set did not overlap with the testing set containing data with noise. In other words, we are performing the learning transfer from the data without noise to predict the BHP for input features containing noise. The metrics values are shown in Table 8, the histogram for NRMSE is shown in Fig. 15, and the cumulative distribution function is shown in Fig. 16. Overall, the errors increased. However, the mean value of NRMSE is slightly above 6%. Selected prediction examples are shown in Fig. 17. Overall, the model is capable of capturing the main features of the transient BHP including amplitudes of noise. Some discrepancies are observed regularly in the beginning of the well cleanup. These discrepancies can be investigated further in possible extensions of this work.

Metric	Mean	95%-quantile	Max
RMSE	11.972	26.502	55.131
MAE	8.793	18.248	41.450
SAE	51.138	114.134	168.169
MAPE	6.199%	12.669%	23.098%
SAPE	53.302%	154.507%	598.044%
NRMSE	**6.226%**	**11.557%**	**19.240%**

Table 8: Metrics values for the prediction using basic model – data with noise.

5. Conclusions

5.1. Current results

The proposed machine learning algorithm for multiphase wellbore flows predicts the BHP using the problem-specific input parameters: wellhead pressure, surface flow rates, and wellbore geometry. As the transient aspects of behavior are well captured and allow us to make predictions of the change in variables over time one can conclude

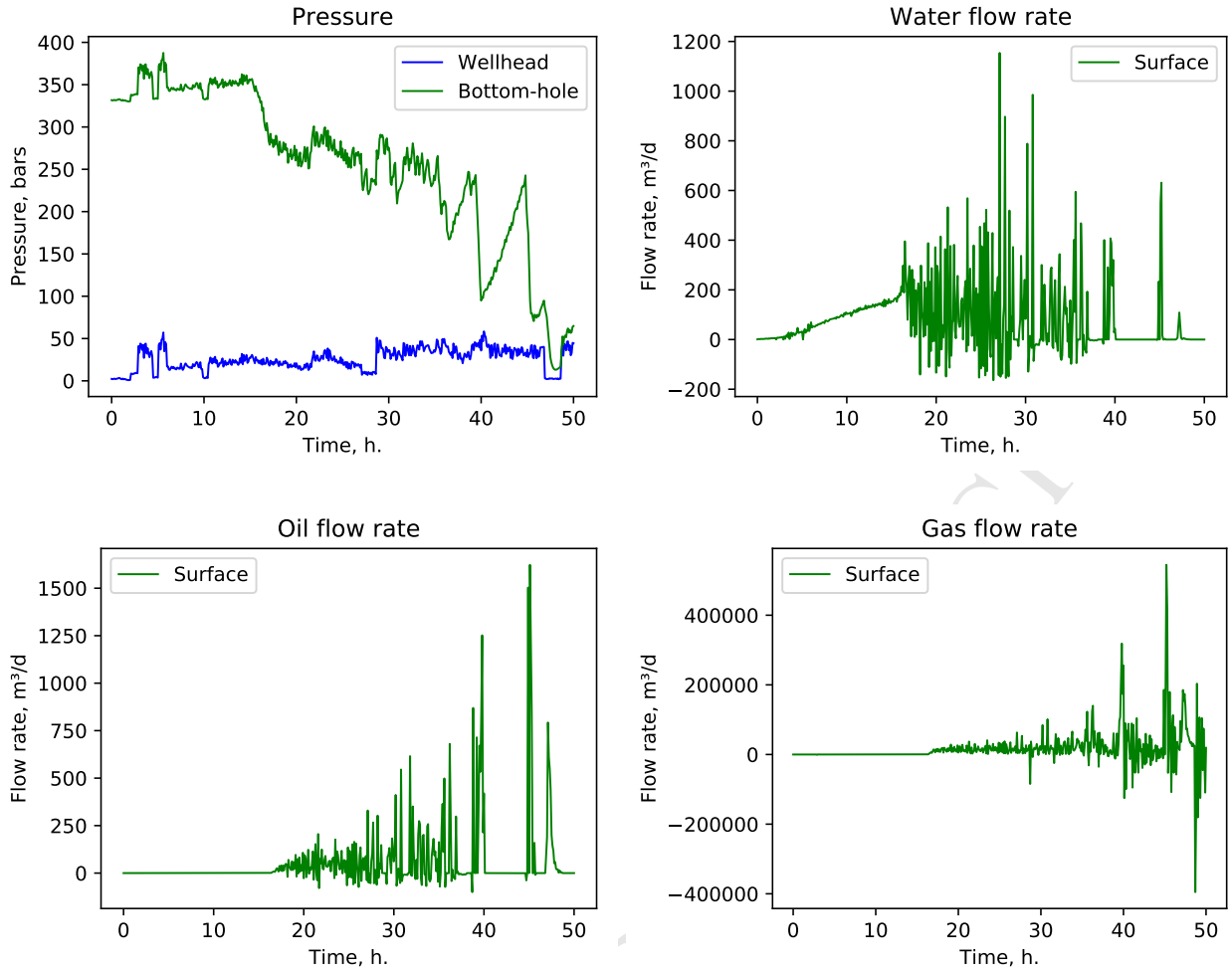


Figure 14: Example of a sample with graphs of $\mathrm{WHP}_{\mathrm{noise}}$ and $\mathrm{BHP}(t)$ functions, and the surface flow rates $Q_o(t)$, $Q_w(t)$, and $Q_g(t)$ for the simulation affected by noise.

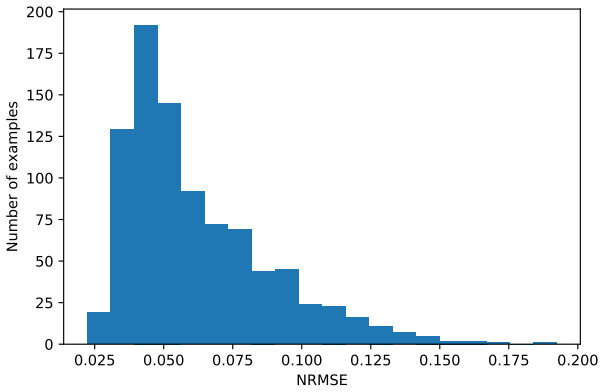


Figure 15: Histogram for NRMSE for the basic model, using data with noise.

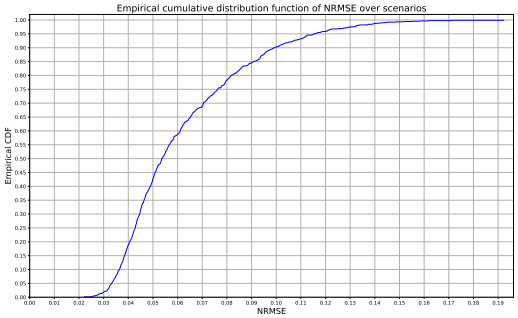


Figure 16: Cumulative distribution function for NRMSE for the basic model, using data with noise.

that the model does identify physical correlations intrinsic to the considered phenomena.

The presented approach uses the multiphase wellbore flow data generated by physics-based simulator of Spesivtsev et al. (2017) as the training dataset for a deep neural network. In the considered problem, the simulator captures the physical effects affecting the dynamics of the BHP, such as non-uniform and transient distribu-

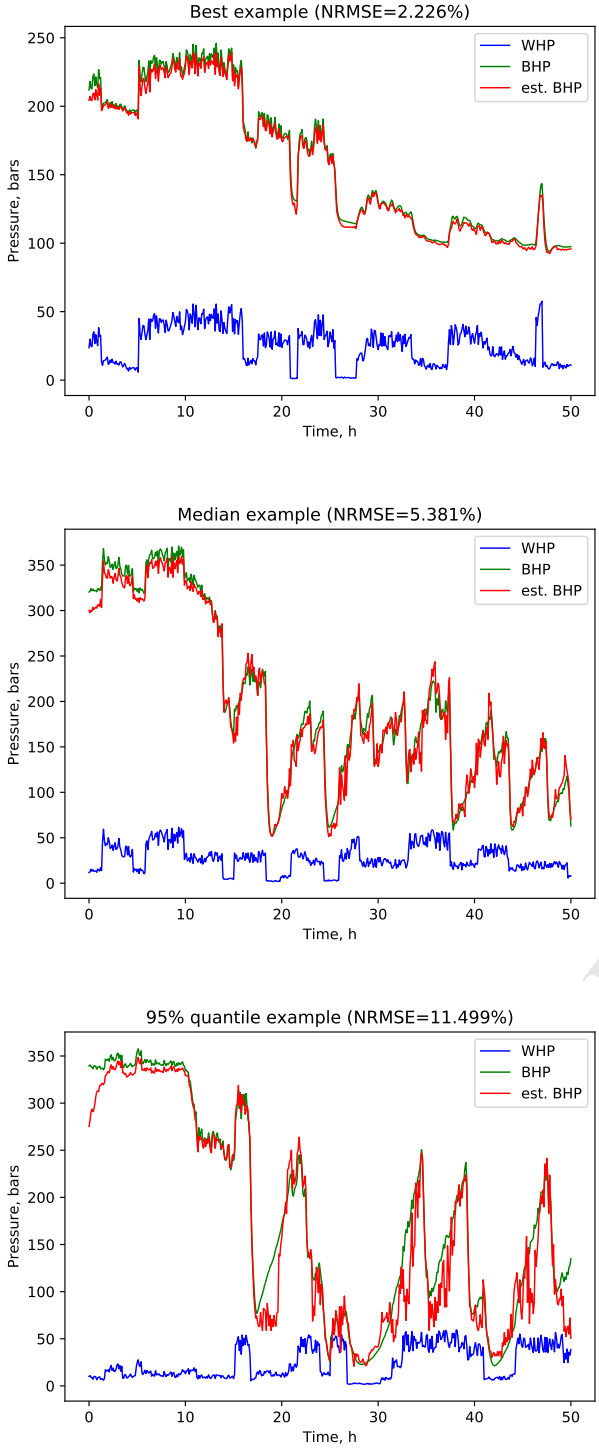


Figure 17: Prediction examples obtained using the basic model and data with noise: blue curve is the WHP, green curve is the simulated BHP, and the red curve is the BHP estimated by the neural network (est. BHP).

tion of phases along the wellbore, velocity difference between phases, release of gas from the oil phase, friction with the pipe walls, and others. It is demonstrated that the machine learning algorithm identifies physical correlations intrinsic to these phenomena. After the training is over, the prediction is almost instantaneous, taking only dozens of milliseconds on the average modern personal computer. Hence, the developed machine learning approach has a potential to process the multiphase wellbore flow data recorded in field operations directly to yield the BHP predictions. The training on actual field data is left for future work. Nevertheless, the presented approach can be used as an alternative to traditional solutions of the inverse problem of determining the BHP from surface measurements for the transient multiphase wellbore cleanup problem.

The training dataset contained samples with complex flow regimes and different well geometries. About 80 features were chosen for the description of the problem. The search for the model that can make good predictions for these samples involved some trial and error and finally led us to the artificial neural network with two hidden layers and the following specifics: 100 and 50 hidden units in these layers, respectively, for the basic model (500 and 150 hidden units for the larger model), ReLU activation function, batch normalization, and AdaDelta optimization.

We observe the following:

- The model can learn to use the geometry information and predict the transient BHP values for different geometries.
- It is demonstrated that the model is capable of predicting the BHP even in the slug flow regime. However, some of the samples in this regime can result in significant prediction errors.
- In most considered cases, the mean NRMSE value was below 5% and the 95%-quantile did not exceed 7%.
- The model trained on original data without noise was applied to predict the BHP for the test dataset containing noise. Although average error increased, the model was capable of capturing the basic features of the noisy transient BHP behavior.

5.2. *Future work*

The slug flows are a challenging problem for both data-driven predictions and classical physics-based modeling and simulation. The slug flows are characterized by periods of time when there is no liquid production rate observed at the surface at all. Hence, the boundary between the regimes where the model performed well and the challenging samples can be established as follows. If the flow rates are continuous and positive at surface, then the developed ANN model performs well according to the required metrics. Typically, if the well is drilled, completed, and stimulated successfully, this should be the case during the well startup and initial production. If, however, for some periods of time, there is no liquid outflow observed from an open wellbore, then most likely the slug

flow regime is formed and, in some cases, predictions might be inaccurate. The analysis of the liquid flow rate continuity can help to discriminate between the flow regimes. To address the issue, a classification step can be run to identify if the wellbore operates in the slug flow regime (e.g., according to the boundary defined above) and then specific ANN focused on slug flows should be applied. By doing this, the overall quality of predictions can be increased. This approach can be extended to cover other flow regimes and should be investigated further. Also, the detailed study of noise influence on the quality of predictions is left for future work.

Acknowledgements

The authors are grateful to Vasily Baydin and Leonid Dovgilovich from Schlumberger Moscow Research for many fruitful discussions and help with the testing of prototype. Also, the authors are thankful to Marina Bulova, Dean Willberg, Bertrand Theuveny, and Schlumberger for supporting this work.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. URL: http://tensorflow.org/. software available from tensorflow.org.

Ahmadi, M.A., Soleimani, R., Lee, M., Kashiwao, T., Bahadori, A., 2015. Determination of oil well production performance using artificial neural network (ANN) linked to the particle swarm optimization (PSO) tool. Petroleum 1, 118 – 132. URL: http://www.sciencedirect.com/science/article/pii/S2405656115000280, doi:https://doi.org/10.1016/j.petlm.2015.06.004.

Al-Jarrah, O.Y., Yoo, P.D., Muhaidat, S., Karagiannidis, G.K., Taha, K., 2015. Efficient machine learning for big data: A review. Big Data Research 2, 87–93. doi:10.1016/j.bdr.2015.04.001.

Ashena, R., Moghadasi, J., 2011. Bottom hole pressure estimation using evolved neural networks by real coded ant colony optimization and genetic algorithm. Journal of Petroleum Science and Engineering 77, 375–385. doi:10.1016/j.petrol.2011.04.015.

Barnea, D., Taitel, Y., 1994. Interfacial and Structural Stability of Separated Flow. Int. J. Multiphase Flow. 20, 387–414. doi:10.1016/0301-9322(94)90078-7.

Bishop, C., 2007. Pattern Recognition and Machine Learning (Information Science and Statistics), 1st edn. 2006. corr. 2nd printing edn.

Bontempi, G., Taieb, S.B., Le Borgne, Y.A., 2013. Machine learning strategies for time series forecasting, in: Business Intelligence. Springer, pp. 62–77.

Chollet, F., 2015. Keras. https://github.com/fchollet/keras.

Ebrahimi, A., Khamehchi, E., 2015. A robust model for computing pressure drop in vertical multiphase flow. Journal of Natural Gas Science and Engineering 26, 1306–1316. doi:10.1016/j.jngse.2015.08.036.

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

Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al., 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine 29, 82–97. doi:10.1109/MSP.2012.2205597.

Hyndman, R.J., Koehler, A.B., 2006. Another look at measures of forecast accuracy. International Journal of Forecasting 22, 679 – 688. URL: http://www.sciencedirect.com/science/article/pii/S0169207006000239, doi:https://doi.org/10.1016/j.ijforecast.2006.03.001.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint abs/1502.03167. URL: http://arxiv.org/abs/1502.03167.

Jahanandish, I., Salimifard, B., Jalalifar, H., 2011. Predicting bottomhole pressure in vertical multiphase flowing wells using artificial neural networks. Journal of Petroleum Science and Engineering 75, 336–342. doi:10.1016/j.petrol.2010.11.019.

Jin, Z., Sun, Y., Cheng, A.C., 2009. Predicting cardiovascular disease from real-time electrocardiographic monitoring: An adaptive machine learning approach on a cell phone, in: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE. pp. 6889–6892. doi:10.1109/IEMBS.2009.5333610.

Jones, E., Oliphant, T., Peterson, P., et al., 2001–. SciPy: Open source scientific tools for Python. URL: http://www.scipy.org/. [Online; accessed 2017-03-12].

Jordan, M., Mitchell, T., 2015. Machine learning: Trends, perspectives, and prospects. Science 349, 255–260. doi:10.1126/science.aaa8415.

Koziel, S., Leifsson, L., 2013. Surrogate-based modeling and optimization. Springer.

Malhotra, P., Vig, L., Shroff, G., Agarwal, P., 2015. Long short term memory networks for anomaly detection in time series, in: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning proceedings, Presses universitaires de Louvain. pp. 89–95.

Mohaghegh, S.D., 2011. Reservoir simulation and modeling based on artificial intelligence and data mining (AI & DM). Journal of Natural Gas Science and Engineering 3, 697–705. doi:10.1016/j.jngse.2011.08.003.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted Boltzmann machines, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807–814.

Ong, Y.S., Nair, P.B., Keane, A.J., 2003. Evolutionary optimization of computationally expensive problems via surrogate modeling. AIAA journal 41, 687–696. doi:10.2514/2.1999.

Osman, E.A., Ayoub, M.A., Aggour, M.A., 2005. An artificial neural network model for predicting bottomhole flowing pressure in vertical multiphase flow, in: SPE Middle East Oil and Gas Show and Conference, Society of Petroleum Engineers. doi:10.2118/93632-MS.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Rabiei, M., Gupta, R., Cheong, Y., Sanchez Soto, G., 2010. A novel approach in extracting predictive information from water-oil ratio for enhanced water production mechanism, diagnosis. APPEA Journal 50, 567–79. doi:10.1071/AJ09034.

Sagiroglu, S., Sinanc, D., 2013. Big data: A review, in: 2013 International Conference on Collaboration Technologies and Systems (CTS), IEEE. pp. 42–47. doi:10.1109/CTS.2013.6567202.

Song, X., Peng, C., Li, G., He, Z., Wang, H., 2016. Optimization of operation parameters for helical flow cleanout with supercritical CO2 in horizontal wells using back-propagation artificial neural network. PLOS ONE 11, 1–29. doi:10.1371/journal.pone.0156358.

Spesivtsev, P., Kharlashkin, A., Sinkov, K., 2017. Study of the transient terrain-induced and severe slugging problems by use of the

drift-flux model. SPE Journal doi:10.2118/186105-PA.

Theano Development Team, 2016. Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints abs/1605.02688. URL: http://arxiv.org/abs/1605.02688.

Theuveny, B., Mikhailov, D., Spesivtsev, P., Starostin, A., Osiptsov, A., Sidorova, M., Shako, V., 2013. Integrated approach to simulation of near-wellbore and wellbore cleanup, in: SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers. doi:10.2118/166509-MS.

Tian, C., Horne, R.N., 2015. Applying machine learning techniques to interpret flow rate, pressure and temperature data from permanent downhole gauges, in: SPE Western Regional Meeting, 27-30 April, Garden Grove, California, USA, Society of Petroleum Engineers. doi:10.2118/174034-MS.

Zeiler, M.D., 2012. ADADELTA: an adaptive learning rate method. arXiv preprint abs/1212.5701. URL: http://arxiv.org/abs/1212.5701.

Zhao, B., Zhou, H., Li, X., Han, D., et al., 2006. Water saturation estimation using support vector machine, in: SEG/New Orleans 2006 Annual Meeting, pp. 1693–1696. doi:10.1190/1.2369848.

Appendix A. Parameters of the geometries

Table A.1 presents the geometric parameters of wellbores used in numerical experiments.

Type	Segment	MD, m	TVD, m	d, m
1a	1	1000.0	1000.0	0.125
	2	3000.0	2638.3	0.125
1b	1	1000.0	1000.0	0.045
	2	3000.0	2638.3	0.095
2a	1	2000.0	2000.0	0.150
	2	5000.0	2000.0	0.150
2b	1	2000.0	2000.0	0.065
	2	5000.0	2000.0	0.135
3a	1	2000.0	2000.0	0.100
	2	3000.0	2939.7	0.100
	3	4000.0	2939.7	0.100
3b	1	2000.0	2000.0	0.030
	2	3000.0	2939.7	0.080
	3	4000.0	2939.7	0.080
4a	1	1000.0	1000.0	0.120
	2	2000.0	1707.1	0.120
	3	3000.0	2707.1	0.120
4b	1	1000.0	1000.0	0.045
	2	2000.0	1707.1	0.100
	3	3000.0	2707.1	0.100
5a	1	3000.0	3000.0	0.110
	2	4000.0	3173.7	0.110
	3	5000.0	3000.0	0.110
	4	6000.0	3173.7	0.110
	5	7000.0	3000.0	0.110
5b	1	3000.0	3000.0	0.060
	2	4000.0	3173.7	0.110
	3	5000.0	3000.0	0.110
	4	6000.0	3173.7	0.110
	5	7000.0	3000.0	0.110

Table A.1: Parameters of the geometries.

Appendix B. Learning curves

Let us show here the learning curves for neural networks for several experiments in this study. For each study, the dataset is split into training and validation sets that do not overlap. Then for each epoch (going through all samples), the training set is used to minimize the loss function. Then the NRMS for the loss function at the current epoch is calculated and plotted for both training and validation datasets as shown in Figs. B.1–B.3. After this, the process is repeated at the new epoch in anticipation of further minimization of the loss function. As can be seen from the figures, in all presented cases, the error is oscillating; however, it is decreasing in general, indicating that the model is successfully learning. The difference between the error on the training set and the error on the validation set

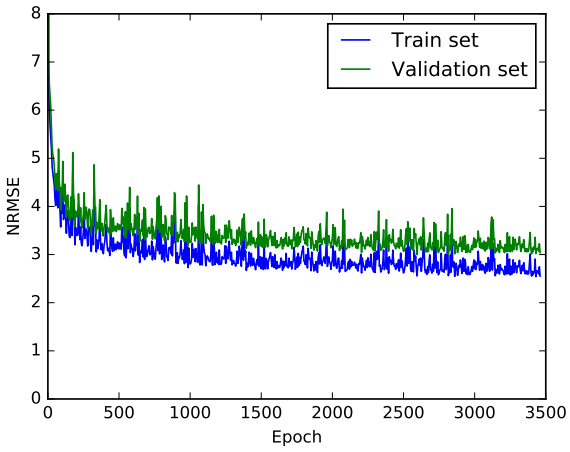


Figure B.1: Learning curves data without filtering.

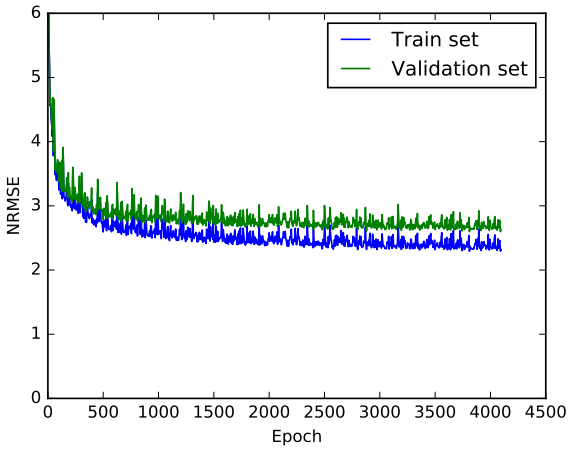


Figure B.2: Learning curves for data with weak filtering.

shows slight overfitting; however, the model still generalizes well. The plots indicate that in our studies, we trained the model for sufficient amount of epochs and the learning error is close to minimum.

Below, we provide the learning and testing time for a typical computer. All given experiments use a model with two hidden layers with 100 and 50 neurons. The computer has the following characteristics:

- CPU Intel Core i7, 4 cores 2.9 GHz
- RAM size: 8 Gb
- GPU: NVidia 5200M
- Disk type: HDD

The learning curves of the prediction model for data without filtering (see Table 1 and Fig. 8) are shown in Fig. B.1. The model learned for about 25 hours.

The learning curves of the prediction model for data with weak filtering (see Table 2 and Fig. 9) are shown in Fig. B.2. The model learned for about 19 hours.

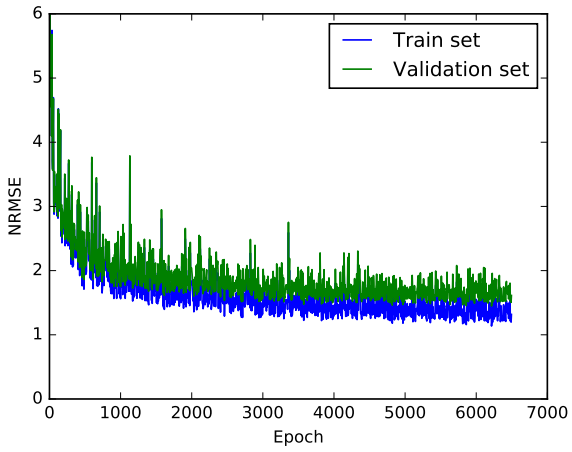


Figure B.3: Learning curves for data with geometries 1, 3 and 4.

The learning curves of the prediction model for data with geometries 1, 3, and 4 (see Table 3 and Fig. 10) are shown in Fig. B.3. The model learned for about 15 hours.

The significant difference in training time is observed because the filtering changes the dataset size and the convergence speed. Making a prediction for one new example takes approximately 0.02 s.

Appendix C. Used libraries

The work was done in Python programming language (version 3.6.0) using the following set of open-source libraries:

- SciPy (Jones et al., 2001–), set of scientific tools
- Scikit-learn (Pedregosa et al., 2011), general machine learning library
- Theano (Theano Development Team, 2016) and Tensorflow (Abadi et al., 2015), neural networks libraries
- Keras (Chollet, 2015), high-level neural networks library