

A Machine Learning Approach for Virtual Flow Metering and Forecasting

Nikolai Andrianov *

* *The Danish Hydrocarbon Research and Technology Centre,
Technical University of Denmark, 2800 Kgs. Lyngby, Denmark
(e-mail: nandria@dtu.dk)*

Abstract: We are concerned with robust and accurate forecasting of multiphase flow rates in wells and pipelines during oil and gas production. In practice, the possibility to physically measure the rates is often limited; besides, it is desirable to estimate future values of multiphase rates based on the previous behavior of the system. In this work, we demonstrate that a Long Short-Term Memory (LSTM) recurrent artificial network is able not only to accurately estimate the multiphase rates at current time (i.e., act as a virtual flow meter), but also to forecast the rates for a sequence of future time instants. For a synthetic severe slugging case, LSTM forecasts compare favorably with the results of hydrodynamical modeling. LSTM results for a synthetic noisy dataset of a variable rate well test show that the model can also successfully forecast multiphase rates for a system with changing flow patterns.

© 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Artificial neural networks, multiphase flow, severe slugging, well testing.

1. INTRODUCTION

Accurate multiphase flow rate measurement is an indispensable tool for production optimization from oil and gas fields, especially in an offshore environment (see e.g. [Amin et al. \(2005\)](#)). Currently, there are two industry-accepted solutions for providing such measurements: using test separators and using multiphase flow meters. While these approaches have their advantages and disadvantages of (see e.g. [Corneliussen et al. \(2005\)](#)), both of them require hardware installations. This can limit the applicability of physical metering devices due to possible transportation issues, space and security considerations, and high costs.

A virtual flow meter (VFM) is a mathematical model which allows to estimate multiphase rates using available data on the flow. A VFM, primarily using readily available cheap measurements (such as pressure and temperature), can potentially serve as a cost-efficient addition to physical flow metering devices.

VFM models can be classified as hydrodynamical or data-driven. In the hydrodynamical approach one typically solves the phase conservation equations in a pipe geometry, which requires the choice of an adequate mathematical model, appropriate numerical method, and availability of a large number of input parameters. An advantage of this method is that one can estimate various parameters at arbitrary points of the flowline. A comparison of several hydrodynamical VFMs is presented in [Amin \(2015\)](#).

The data-driven approach is a system identification tool, which requires the user to accept one of generic model structures. Such models exploit no prior knowledge on the flow and produce essentially data descriptions. In practice, it is easier to setup a data-driven model as compared to a hydrodynamical one. However, data-driven predictions do

not have a physical interpretation and it is not possible to estimate parameters with no historical data. Despite these shortcomings, the use of data-driven VFMs is gaining momentum in the industry, see [Briers et al. \(2016\)](#).

One important difference between hydrodynamical and data-driven VFMs is the ability of the latter not only to *estimate* rates at the current time instant t_k , but also to *forecast* rates at future time instants t_{k+1}, t_{k+2}, \dots . Indeed, without *a priori* knowledge of time-varying boundary conditions, a hydrodynamical model is only able to yield forecasts at the next time instant t_{k+1} .

The goal of the present paper is to evaluate the forecasting capability of a class of data-driven VFMs which use artificial neural networks (ANNs). Feedforward ANNs have been successfully used in VFM predictions by many authors (see e.g. [Al-Qutami et al. \(2018\)](#) and the references therein). However, the forecasting capability of feedforward ANNs is limited because they are unaware of the temporal structure or order between observations.

Recent results in such applications as automatic text translation and image captioning suggest that the Long Short-Term Memory (LSTM) model of [Hochreiter and Schmidhuber \(1997\)](#) is an efficient tool for time series forecasting.

In order to assess the LSTM model performance for VFM applications, we consider a synthetic two-phase severe slugging case (see [Andrianov et al. \(2007\)](#)) and a three-phase well testing dataset.

For the severe slugging data set, we demonstrate superior performance of LSTM as compared to the feedforward ANN sliding window approach. We investigate the LSTM convergence as a function of provided distributed pressure

measurements and determine the optimal model configuration.

For the variable rate well test data set, we show that LSTM can successfully handle a noisy dataset, describing a system with changing flow patterns. The accuracy of the forecast improves with the the number of flow periods used for training the model.

2. LSTM MODEL SETUP

Consider a time series $\{\mathbf{x}(t_k)\}$ and $\{\mathbf{y}(t_k)\}$, where $\mathbf{x}(t_k)$ is a m -dimensional vector of *input features* and $\mathbf{y}(t_k)$ is a n -dimensional vector of *output features* values at equally spaced time instants t_k . In VFM applications, features are the measurement data acquired at different points of the flowline. One can select the sets of the input and output features independently from each other. In particular, a feature can simultaneously be used for both input and output (e.g., we might be willing to forecast future values of a flow rate from its past values).

We are interested in forecasting the sequences of output features of length l_o using the sequences of input features of length l_i . To this end, the terms $\mathbf{x}(t_k)$ and $\mathbf{y}(t_k)$ from the *training interval* $[t_0, t_L]$ are divided into N overlapping sequences of length $l = l_i + l_o$, shifted by an indentation step s . The result can be cast in form of the training array

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(t_0) & \dots & \mathbf{x}(t_{l_i-1}) \\ \mathbf{x}(t_s) & \dots & \mathbf{x}(t_{s+l_i-1}) \\ \dots & & \dots \\ \mathbf{x}(t_{L-l+1}) & \dots & \mathbf{x}(t_{L-l_o}) \end{bmatrix} \quad (1)$$

and the target array

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}(t_{l_i}) & \dots & \mathbf{y}(t_{l-1}) \\ \mathbf{y}(t_{s+l_i}) & \dots & \mathbf{y}(t_{s+l-1}) \\ \dots & & \dots \\ \mathbf{y}(t_{L-l_o+1}) & \dots & \mathbf{y}(t_L) \end{bmatrix}, \quad (2)$$

so that $\mathbf{X} \in \mathbb{R}^{N \times l_i \times m}$ and $\mathbf{Y} \in \mathbb{R}^{N \times l_o \times n}$.

LSTM maps an input sequence $\mathbf{x}(t_k), \dots, \mathbf{x}(t_{k+l_i-1})$ to the output sequence $\hat{\mathbf{y}}(t_{k+l_i}), \dots, \hat{\mathbf{y}}(t_{k+l-1})$ for $k = 0, \dots, N$. LSTM is conventionally represented as an *unfolded in time* layered network of *memory cells*, followed by a *time-distributed* layer, as illustrated in Fig. 1.

A memory cell is defined as follows,

$$g(t) = \varphi(\mathbf{W}^{gx} \cdot \mathbf{x}(t_k) + \mathbf{W}^{gh} \cdot \mathbf{h}(t_{k-1}) + b^g) \quad (3)$$

$$i(t) = \sigma(\mathbf{W}^{ix} \cdot \mathbf{x}(t_k) + \mathbf{W}^{ih} \cdot \mathbf{h}(t_{k-1}) + b^i) \quad (4)$$

$$f(t) = \sigma(\mathbf{W}^{fx} \cdot \mathbf{x}(t_k) + \mathbf{W}^{fh} \cdot \mathbf{h}(t_{k-1}) + b^f) \quad (5)$$

$$o(t) = \sigma(\mathbf{W}^{ox} \cdot \mathbf{x}(t_k) + \mathbf{W}^{oh} \cdot \mathbf{h}(t_{k-1}) + b^o) \quad (6)$$

$$s(t) = g(t_k)i(t_k) + s(t_{k-1})f(t_k) \quad (7)$$

$$h(t) = \varphi(s(t_k)o(t_k)), \quad (8)$$

where the hyperbolic tangent φ and the sigmoid σ are the *activation functions*; C is a number of memory cells in LSTM's layers; $\mathbf{W}^{\cdot,x} \in \mathbb{R}^{m \times C}$, $\mathbf{W}^{\cdot,h} \in \mathbb{R}^{C \times C}$, and $b \in \mathbb{R}$ are memory cells' *weights* and *biases*; $\mathbf{x}(t_k) \in \mathbb{R}^m$ is the input vector at the current time instant; and $\mathbf{h}(t_{k-1}) \in \mathbb{R}^C$ is the vector of current layer's output values of memory cells from the previous time instant.

The time-distributed layer cells are defined via a linear transformation

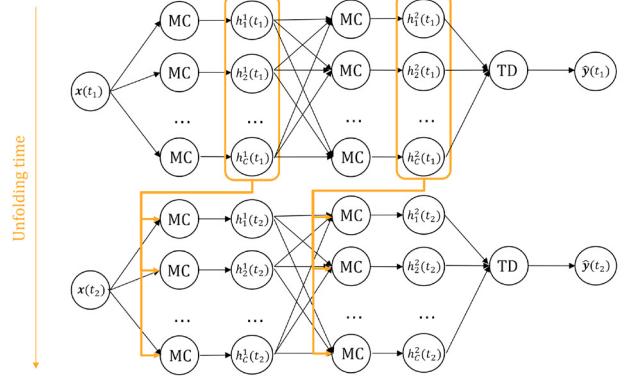


Fig. 1. An example of a two layer LSTM with each layer containing C memory cells (MCs), followed by a time-distributed (TD) layer. Temporal dependencies are highlighted in yellow.

$$\hat{\mathbf{y}}(t_k) = \mathbf{W}\mathbf{h}(t_k) + \mathbf{b}, \quad (9)$$

where $\mathbf{W} \in \mathbb{R}^{n \times C}$ and $\mathbf{b} \in \mathbb{R}^n$ are the corresponding weights and biases.

The trainable parameters of the model (i.e., weights and the biases of all cells) are iteratively updated to minimize a loss function, which penalizes the distance between the output and the target sequences. The original LSTM by Hochreiter and Schmidhuber (1997) is limited to the case when $l_i = l_o$; Cho et al. (2014) and Sutskever et al. (2014) introduced an encoder-decoder architecture to generalize the LSTM applicability for cases with $l_i \neq l_o$. See Lipton et al. (2015) for a review.

In this work, we use Keras implementation of LSTM, see Chollet et al. (2015). The simulation scripts with the corresponding datasets (see below) are publicly available under <https://github.com/nikolai-andrianov/VFM>.

3. EXPERIMENTS

3.1 Severe Slugging Case

Consider a two-phase isothermal gas-liquid flow in a 60 m section of an offshore pipeline, ending with a 14 m long riser. The flow can be described by a set of partial differential equations, expressing conservation of mass and momentum for the phases. We will be using the mathematical model, numerical method, and the specifications for the test case, presented in Andrianov et al. (2007).

Under certain constant boundary conditions at the pipeline inlet and at the riser outlet, the numerical solution exhibits a typical severe slugging behavior, see Fig. 2.

We will utilize this numerical solution as a “ground truth” for forecasting the liquid and gas rates at the riser bottom using the data from virtual pressure gauges distributed along the flowline, see Fig. 3.

In order to run LSTM forecasts, we resample the normalized pressure and flow rate data with a uniform timestep of 1 sec, and use half of the total hydrodynamical simulation time as a training interval, $[t_0, t_L] = [0, 1500]$ sec.

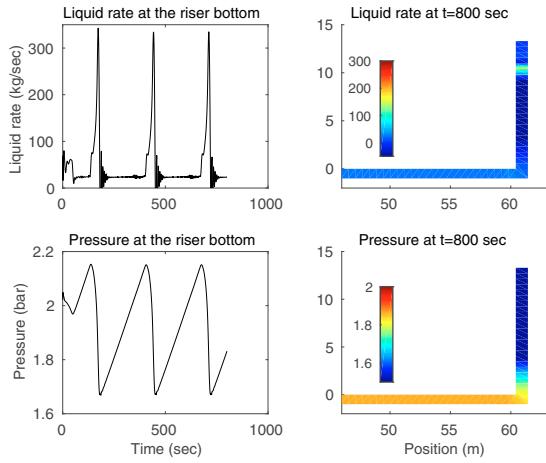


Fig. 2. A snapshot of the numerical solution for the severe slugging case at intermediate time instant.

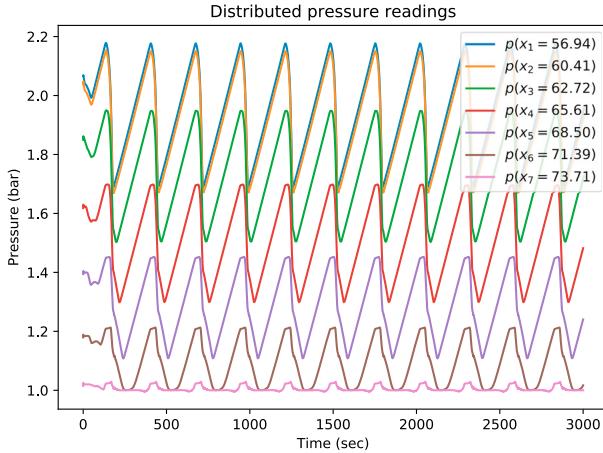


Fig. 3. Pressure data used as input to forecast the flow rates.

We first analyze the quality of LSTM forecasts when the network is trained using only pressure readings as input and liquid rate as output. The training data is divided into $N = 1127$ sequences of length $l = 374$ sec with $l_i = l_o$, shifted by the indentation step $s = 1$ sec. The network details are given below:

- Deep LSTM with 3 hidden layers and 10 memory cells at each layer;
- Total number of trainable parameters is $2171, \dots, 2411$ with validation split of 0.05 for the number of input features $m = 1, \dots, 7$ and number of output features $n = 1$;
- Fixed random seed for repeatability in parameter initialization;
- Mean squared error (MSE) loss function and Adam optimizer of Kingma and Ba (2014) with batch size of 1 and number epochs equal to 10.

These network training parameters were determined by trial-and-error. For the case considered, the forecasting results were most sensitive to the number and lengths of input/output sequences.

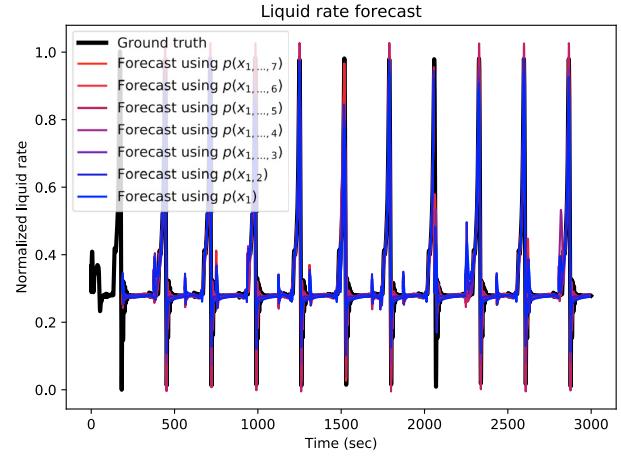


Fig. 4. LSTM liquid rate forecasts using various number of pressure readings as input.

The forecasting capability of an LSTM can be quantified with the ratio

$$f = \frac{l_o}{t_L - t_0} \cdot 100\%, \quad (10)$$

which we will term the *relative forecasting interval*. For the severe slugging case $f = 12.4\%$, i.e. the LSTM can forecast the future flow rates for the time interval which length is 12.4% of the length of LSTM's training interval.

The forecasts are plotted as 15 *non-overlapping* sequences of length $l = 374$ sec with $l_i = l_o = 187$ sec, shifted by the indentation step $s = 187$ sec, see Fig. 4. Observe that even when trained on a single pressure reading, LSTM yields excellent agreement with the ground truth hydrodynamical solution in terms of the frequency and amplitude of the liquid rate peaks. This is in striking contrast to the results of a feedforward ANN using sliding window approach with 3 hidden layers and 10 neurons at each layer, trained on the same dataset as the LSTM, see Fig. 5.

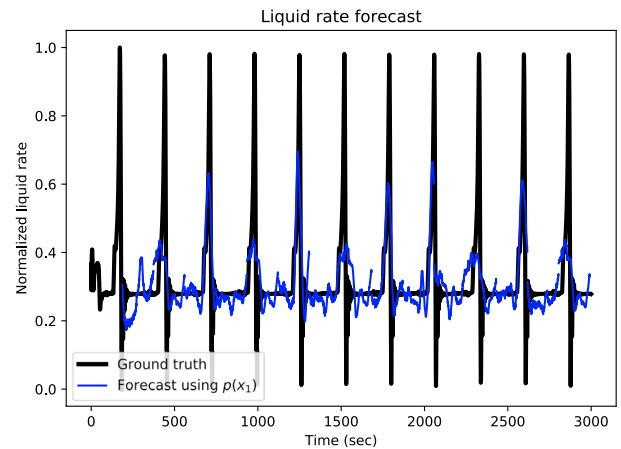


Fig. 5. Feedforward ANN liquid rate forecasts using a single pressure readings as input.

Adding more pressure data as the training input does generally increase the accuracy of LSTM forecasts. However, this improvement is not monotonous, and starting from a certain number of pressure readings (in this case

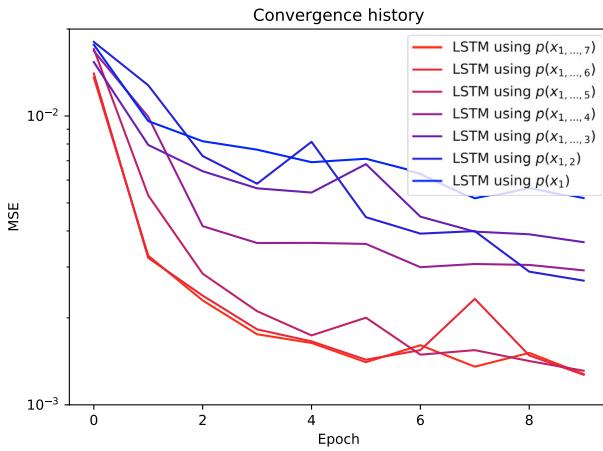


Fig. 6. LSTM convergence history as a function of number of pressure readings used to train the network.

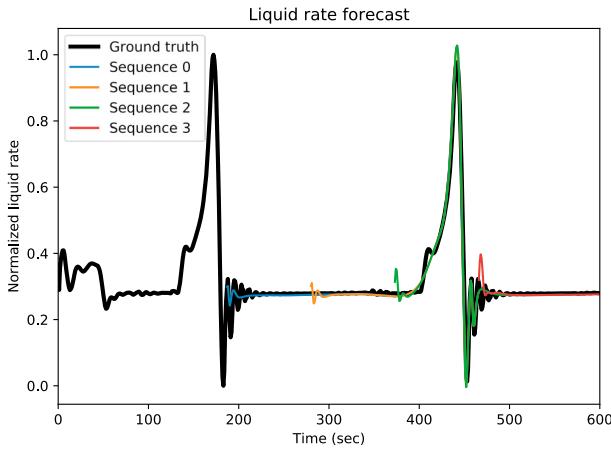


Fig. 7. First output sequences of zoomed LSTM liquid rate forecasts using 5 pressure readings as input.

5 readings) the accuracy remains essentially the same, see Fig. 6. One can argue that this new data is similar to the one already used for model training (cf. Fig. 3), so the model performance does not improve any further.

Note that there are spurious oscillations visible in LSTM forecasts on Fig. 4. We replot the zoomed LSTM forecasts using 5 pressure readings as *overlapping* sequences of the same length $l_i = l_o = 187$ sec, but shifted by the indentation step $s = 93$ sec, see Fig. 7.

Observe that the spurious oscillations are located at the beginning of each sequence. This is not surprising because LSTM learns its weights within the input sequence. However, these oscillations do not affect the accuracy of the forecasts if we use overlapping output sequences. Indeed, referring to Fig. 7 we have by the end of $t = l_i$, the 0th sequence forecast till $t = 2l_i$, which is oscillation-free by the time $t = l_i + s$, when the new 1st sequence forecast is made till $t = 2l_i + s$. We keep using the 0th sequence forecast until the oscillations in the 1st sequence forecast disappear, and repeat the process.

The performance of LSTM trained on pressure *and* liquid rate is presented in Fig. 8. Observe that increasing the

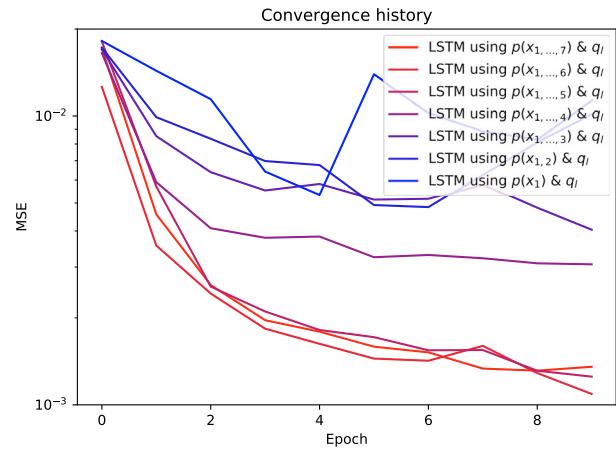


Fig. 8. Convergence history of LSTM trained on several pressure readings and the liquid rate.

number of measurements used to train the network does not improve the accuracy of the forecast, cf. Fig. 6. Moreover, if few pressure readings are used for training, the performance of LSTM trained on pressure and rate data becomes worse than that of LSTM trained just on pressure data. In other words, the improvement in model accuracy is not monotonous with respect to the amount of training data used.

The accuracy of LSTM forecasts of both liquid and gas rates (i.e., $n = 2$ output features) is essentially the same as the results presented above for liquid rate forecasts only.

We also tested the encoder-decoder LSTM of Cho et al. (2014) and Sutskever et al. (2014), but the forecasts were less accurate compared to the results presented above.

Wall time required for training of LSTMs described above with was approx. 30 mins using a single core of i7-7700HQ CPU. Using 8 cores of the same CPU resulted in approx. 20% speedup.

3.2 Variable Rate Well Test

Consider a synthetic dataset of pressure, temperature, and oil, gas and water rates measurements during a well test, see Fig. 9 and Fig. 10. The data is characteristic for a rich gas condensate *deliverability* test, which involves flowing the well on successively larger choke openings in order to determine the well's inflow performance relationship (IPR) and maximize gas condensate recovery. (In what follows, we will refer to gas condensate as "oil".)

The dataset consists of 5 flow periods, which are characterized by the corresponding choke opening. Within each flow period, the measurements are generally sampled with the uniform timestep of 1 min. We are interested in forecasting the multiphase rates using the values of pressure and temperature.

To this end, we will be utilizing essentially the same procedure as for the severe slugging experiment. We train the network on first flow periods using pressure and temperature readings as input features and multiphase rates as output features. Then, the multiphase forecasts are run for all flow periods.

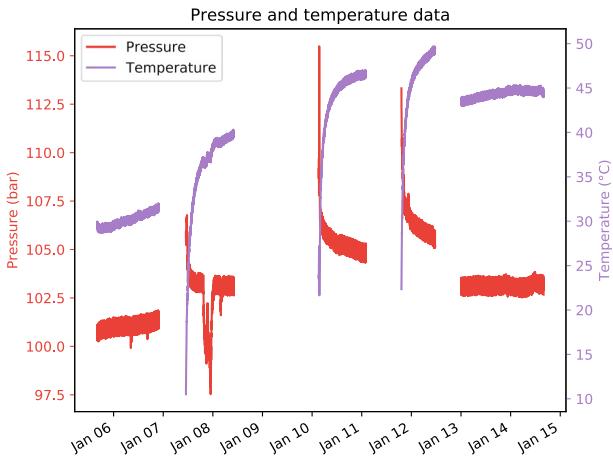


Fig. 9. Pressure and temperature for the variable rate well test.

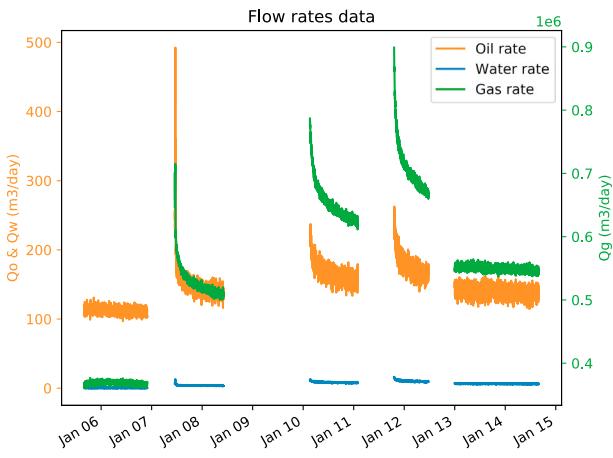


Fig. 10. Multiphase rates for the variable rate well test.

One key difference of the variable rate well test case from the severe slugging case considered in the previous section is that the flow pattern in the well test case changes drastically from one flow period to another, see Fig. 9 and Fig. 10. This constitutes a challenge to the neural network, because we try to approximate the behavior of the *changing* flow system with the same ANN. Another difference and a challenge for the neural network is that the dataset is noisy.

In what follows, we will compare the forecasting accuracy of LSTMs trained on first 2 and 3 flow periods. For these two cases, the training data is divided into $N = 2705$ and $N = 3828$ sequences, respectively. In both cases the sequence length is $l = 244$ min with $l_i = l_o$, and the indentation step is $s = 1$ min. The relative forecasting intervals are $f = 3.8\%$ and $f = 2.6\%$, respectively. The LSTM structure is the same as described in the previous section. The forecasts are presented in sequences of $l = 244$ min with the indentation step is $s = l_i/2$ min.

The results for the LSTM, trained on first 2 flow periods, are presented in Fig. 11.

The model reproduces well the training data from the first 2 flow periods. The best accuracy is achieved for forecasted

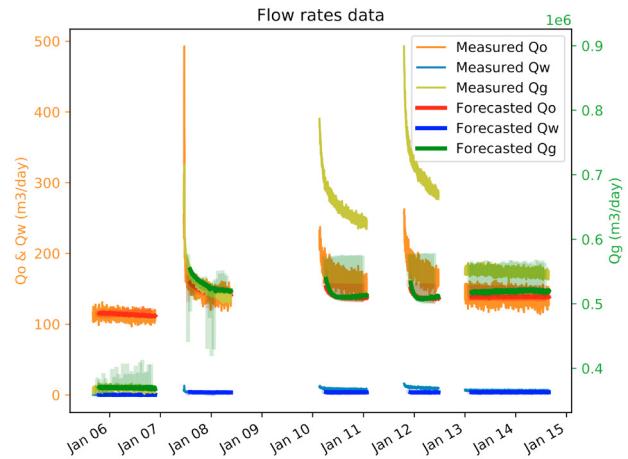


Fig. 11. Multiphase rates forecast for the LSTM trained on first 2 flow periods. Spurious oscillations are plotted semi-transparent.

values of oil and water rates, while the gas rate is slightly overestimated. Still, the trends for all rates are captured correctly.

On the testing set (flow periods 3 to 5), the model yields reasonable values for the oil and water rates. However, the forecasts for gas rate are non-satisfactory. This can be explained by the fact that both oil and water rates lie in a same range throughout all flow periods, which is not the case for the gas rate. Also, note that the first l_i data points on each training sequence are not covered by any output sequence. Consequently, the sharp peaks at the beginning of the flow periods 2, 3, and 4 are not included in the training dataset.

On Fig. 11 we witness the same spurious oscillations as discussed in the previous section, cf. Fig. 7. To see this, in Fig. 12 we plot the measured gas rate together with the first output sequences of forecasted gas rate during the 1st flow period. Observe that the peaks are located at the beginning of each output sequence of length $l_o = 122$ min. By following the same reasoning as in the previous section, we argue that these spurious oscillations do not affect the quality of the forecast.

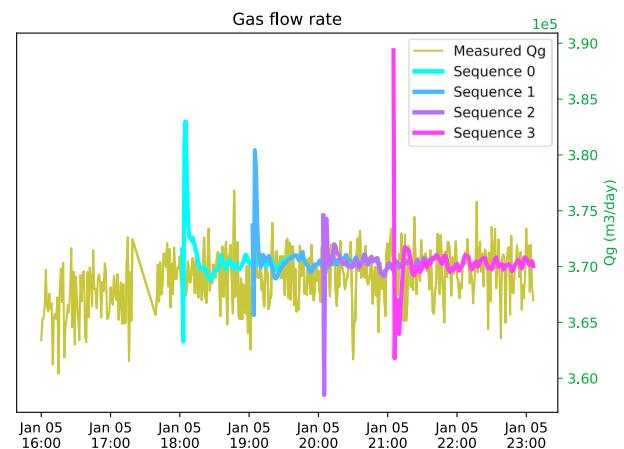


Fig. 12. First output sequences of zoomed LSTM gas rate forecasts.

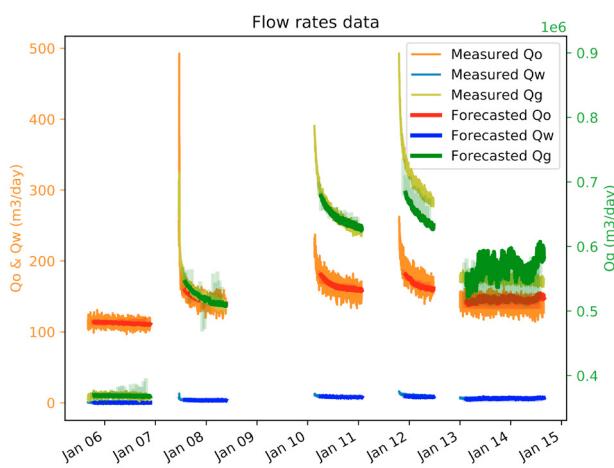


Fig. 13. Multiphase rates forecast for the LSTM trained on first 3 flow periods. Spurious oscillations are plotted semi-transparent.

The results for the LSTM, trained on first 3 flow periods, are presented in Fig. 13. The model reproduces well all training data from the first 3 flow periods. It is interesting to note that the forecasts are less noisy as compared to the measured data. There are spurious oscillations visible on the graphs, but their amplitude is less than that of the model, trained on 2 flow periods.

On the testing set (flow periods 4 and 5), the model performance is best for oil and water rates, and less satisfactory for gas rates. Again, this can be explained by a larger variability of the gas rate as compared to oil and water rates. Overall, the accuracy of the forecast is considerably better compared to that of the model, trained just on 2 flow periods.

Wall time required for training of LSTMs on first two and three flow periods using a single core of i7-7700HQ CPU was approx. 50 and 90 min, respectively.

4. CONCLUSION

In this work, we have shown that LSTM can be considered as a promising tool for forecasting the values of multiphase rates using pressure and temperature data. The best accuracy was achieved when the lengths of the input and output sequences to LSTM were equal. Consequently, we are limited in the length of the time interval suited for forecasts. Removing this limitation without sacrifice on the accuracy of the forecast can be an interesting topic of future research.

ACKNOWLEDGEMENTS

The author would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

- Al-Qutami, T.A., Ibrahim, R., Ismail, I., and Ishak, M.A. (2018). Virtual multiphase flow metering using diverse neural network ensemble and adaptive simulated annealing. *Expert Systems with Applications*, 93(Supplement C), 72–85.

Amin, A. (2015). Evaluation of commercially available virtual flow meters (VFM's). 1–26. Offshore Technology Conference. 25764-MS.

Amin, A., Riding, M., Shepler, R., Smedstad, E., and Ratulowski, J. (2005). Subsea development from pore to process. *Oilfield Review*, 17(1), 4–17.

Andrianov, N., Coquel, F., Postel, M., and Tran, Q.H. (2007). A relaxation multiresolution scheme for accelerating realistic two-phase flows calculations in pipelines. *Int. J. Numer. Meth. Fluids*, 54(2), 207–236.

Briers, J., Goh, K., Sniekers, A., Schotanus, D., Hofland, J., and Adun, D. (2016). Looking back 2006 - 2016 - ten years of data driven well rate estimates for real-time surveillance and optimization. 1–12. Society of Petroleum Engineers. 181032-MS.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *ArXiv e-print 1406.1078*.

Chollet, F. et al. (2015). Keras.

<https://github.com/fchollet/keras>.

Corneliussen, S., Couput, J., Dahl, E., Dykesteen, E., Frøysa, K., Malde, E., Moestue, H., Moksnes, P.O., Scheers, L., and Tunheim, H. (2005). *Handbook of Multiphase Flow Metering*. Norwegian Society for Oil and Gas Measurement.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv e-print 1412.6980*.

Lipton, Z.C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *ArXiv e-print 1506.00019*.

Sutskever, I., Vinyals, O., and Le, Q.V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, 3104–3112.