

Received June 25, 2019, accepted July 20, 2019, date of publication July 30, 2019, date of current version August 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2931990

Advancing Deep Learning to Improve Upstream Petroleum Monitoring

CRISTINA HEGHEDUS^{ID1}, ANTON SHCHIPANOV², AND CHUNMING RONG^{ID1}, (Senior Member, IEEE)

¹Department of Computer Science and Electrical Engineering, University of Stavanger, 4021 Stavanger, Norway

²NORCE Norwegian Research Center, 4021 Stavanger, Norway

Corresponding author: Cristina Heghedus (cristina.heghedus@uis.no)

This work was supported by Project “SuperData” of the University of Stavanger, funded by Norwegian National Funding (NFR).

ABSTRACT Data analytics is rapidly growing field in both academia and industry dealing with processing and interpreting large and complex data sets. It has got already many successful applications via advancing machine (ML) and deep learning (DL) techniques, starting to evolve in the upstream petroleum industry as well. The industry operates now with huge amount of sensors installed in different facilities, particularly in production and injection wells. These sensors provide millions of measurements, such as pressure, temperature, and rate every year for every well. The measurements may be highly correlated and carry crucial information for decision making. This paper concentrates on pressure-rate data sets accumulated with massive installation of permanent downhole gauges in such wells. The non-linear autoregressive (NARX) and the long short term memory (LSTM) neural networks were assembled and tested on a synthetic data set to compare results of pressure prediction, already addressed in the literature. The LSTM provided better predictions, but did not manage to capture entirely the pattern of the data. The *shifting window* method was then applied to improve the LSTM prediction capabilities, based on previous successful application in forecasting electricity demand. The method implies smooth transition from training to prediction improving network performance. The LSTM with the *shifting window* provided more accurate results for pressure prediction, and it was then successfully applied for rate prediction. Testing of different configurations of the LSTM network has shown that the pressure prediction performs well with less number of nodes in the hidden layers if compared with the rate predictions. Significant error decrease is achieved relatively fast (after 20 iterations) for both prediction tasks, making such predictions feasible for large data sets. The results provide basis for filling gaps in well monitoring data and short-term performance forecast, crucial tasks for decision making in all the industries operating with wells.

INDEX TERMS Deep learning, NARX, LSTM, well monitoring, permanent downhole gauge (PDG).

I. INTRODUCTION

Many industries use production and injection wells, originated from hydrogeology and petroleum industry, wells are now operated in geothermal energy systems, geological storage of carbon dioxide etc. During well operations (production, injection or shut-in), data on well rate and pressure are collected with respect to time. Over time, the pressure has either an increasing or a decreasing trend, depending on the rate and well type (producer or injector). In this paper, it is considered an injection well distinguishing two cases of rate behavior: when the rate is kept at a non-zero constant value and when the rate is zero for a period of time. In the first

The associate editor coordinating the review of this manuscript and approving it for publication was Qiang Yang.

case, the pressure is increasing (pressure build-up) and in the second case the pressure will suddenly drop as the well is shut down and not injecting during this time (well shut-in and pressure fall-off).

In the petroleum industry, Pressure Transient Analysis (PTA) was traditionally associated with well testing, where a process of executing a set of planned well operations and data acquisition activities is aimed at acquiring knowledge of properties of in-situ hydrocarbons and characteristics of the saturated reservoir rocks. Such a knowledge contributes to the decision making process giving an estimate of hydrocarbon reserves at the exploration phase and potential well performance and life-span at the production phase.

In a wider context, PTA is used to characterize different flow regimes related to well trajectory, completion, reservoir

boundaries and well interference (like radial, linear, spherical flows) and well and reservoir parameters: well skin (damage or stimulation), reservoir flow capacity, average reservoir pressure etc., [1]. The scope of PTA applications was recently enlarged with massive installation of Permanent Downhole Gauges (PDGs) in modern wells, providing millions of pressure measurements per year. Analysis and interpretation of the massive PDG data sets, for example, applying PTA, requires a combination with other data sets like rate data. Such a combination requires a sequence of routines to be applied: synchronization, filtering, noise reduction and data mining. The last focuses on filling gaps in one data set, that may be performed based on the other data set. In many cases, frequency and quality of pressure and temperature data are much higher than those of rate data, leading to necessity to fill in the gaps in rate history for further analysis, e.g. with PTA.

In this paper, it is discussed a typical data set as recorded by a PDG, although the approaches developed may be further applied to other well data such as temperature or distributed (throughout the wellbore) measurements. PDGs are installed in oil and gas wells for the purposes of both observation and optimization and can monitor pressure at a single location or multiple points inside a well. Having such a sensor permanently mounted enables for real-time monitoring and automatic adjustments of well operational envelope (pressure and rate), which can help to ensure well integrity and to optimize the well performance. Widely used in the oil and gas industry, PDGs have also found applications in other industries operating with wells like geological storage of carbon dioxide and geo-thermal energy. Performance optimization of wells operating in the industries mentioned above will contribute to efficient resource management (key challenge of the modern world, considering Earth's finite resources), relying on Knowledge Engineering (KE) [2]. Combining PDG data with Artificial Intelligence (AI), this work aims to solve a KE problem in the industries mentioned above, which is mainly related to performance optimization. Machine Learning (ML) models may be used to establish the relationship between pressure and flow rate data, and therefore to achieve objectives of the study, namely filling the gaps in the data and making forward predictions.

The scope of this work is reproducing or prediction of well flow rate time series based on pressure time series from PDG and vice versa [3], which is, for example, of special interest to the data mining routines discussed above (a data-driven approach). A Deep Learning (DL) [4] technique was applied for such a prediction, which was not used before for such task. For comparison, another DL model [3], that was used for similar tasks before, was also tested. The rate reconstruction has a value for time-lapse PTA of PDG data [5], enabling filling gaps in rate measurements. Presence of such gaps is still a usual case in the petroleum industry due to installation of flow-meters for clusters of wells and following rate allocation per well. Constructing pressure evolution at given rate may be also used for predicting well performance.

Such a prediction is basis for decision making in the upstream oil and gas industry. The pressure prediction may also be considered in the context of PTA, where pressure derivative of the reconstructed pressure transients is of special interest.

II. TECHNICAL BACKGROUND

In this section, details of PDG data and the methods applied to such data sets are presented. The methods vary from widely used statistical, mathematical models to some AI models. Recently, the interest to AI applicability to well data increased and the advantages of automation became evident [7]. AI represents the ability of computers to mimic human behavior, in term of analyzing information and making decisions (either on a classification or a prediction task). In this work, an AI model is built, which receives as an input raw PDG and rate data and produces an output as it will be described in Section III. Traditional approaches are presented in the first part of the section on different tasks that have been solved, followed by a description of ML models and the problems, where these models have been applied.

A. PDG DATA AND EXISTING PTA TECHNIQUES

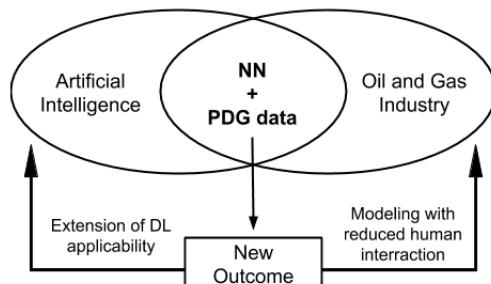
Data, collected with PDGs, form a basis for monitoring well and reservoir parameters and well performance as well as for making predictions about well behavior. For instance, PDG enables using transient data to analyze and interpret pressure behavior by taking measurements while intentionally varying the flow rate. Analyzing the PDG data in combination with flow rates in a manual manner is a challenge, due to large amounts of the data sets, synchronization and noise issues. Usually, PTA is applied on well-by-well basis, however in the recent years the focus was also shifted to multi-well interpretation [8].

The PTA methods have progressed from straight lines via type curve analysis to pressure derivative and deconvolution [9]. Today, the Log-Log (pressure and its derivative) plot is the mostly used tool, which is a representation of the numeric data on a two-dimensional graph, with both axes having logarithmic scale. A pressure transient obtained in a shut-in of flowing period is plotted here as dP , which is equal to the difference between pressure at time " t " and P_i , pressure at initial time (start of the period).

Although these models were widely used in the past years in a manual mode, they are not fully automated and human interaction is needed for their application. The ML/DL approaches have here advantages of performing routine operations and facilitating modeling based on raw data, using computational power and minimizing human interaction.

B. DATA ANALYTICS AND MACHINE LEARNING

Machine Learning (ML) techniques have been recently tested on different PDG data sets, in a single well [10] and multi-well analysis [11]. The core idea was to learn patterns and gain insights on future behavior from the PDG data. ML is a field of Computer Science that gives computers the ability to learn, based on data and errors, without specific instructions.

**FIGURE 1.** Interaction of AI with Oil and Gas Industry.

Within the area of Data Analytics, Machine Learning represents a way to build complex techniques and algorithms, able to make future predictions. This is also known as Predictive Analytics. The models used in predictive analytics enable developing strategies, that contribute to decision making and efficient resource management.

Over many years, Neural Networks (NN) were used in well test analysis with the focus on identifying relevant interpretation models [12] [13]. One of the recent works by Ahmadi *et al.* [14] also used NN with the scope of diagnosing the well testing model, incorporating dimensionality reduction and normalization of the dataset. Moreover, in 2017 a NN was applied [3] for predicting the pressure and temperature in a well based on flow rate, experiment which is partially reproduced in our paper.

Currently, applications of AI in well testing and monitoring, lack approved, tested and universally applicable techniques for reconstruction of one set of well measurements based on another (like rate based on pressure). Hereafter, in this work, DL models are tested, which could be capable of predicting flow rate based on time and pressure as input, and vice versa. Here, DL represents a set of neural networks (Deep Neural Networks - DNN) with complex layers and multiple functions interacting in specific ways.

In practice, challenges in PTA using Machine Learning models are also related to noisy rate and pressure data. De-noising is out of the scope of this work, as the dataset analyzed was generated using reservoir simulations. Advantage of synthetic data is the fact that the ‘true’ solution is known making development and testing of new approaches straight forward.

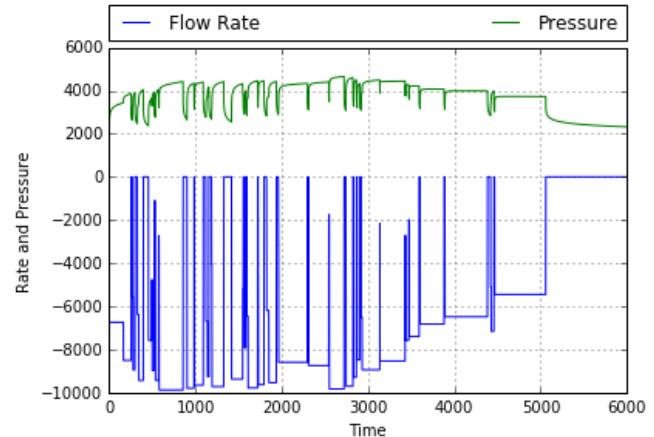
III. WORKFLOW

The main goal of this work is to develop and apply DL techniques to well data, without having major knowledge regarding the extraction process and physical relationship between the measured parameters. In other words, to test the chosen model’s performance on raw data from oil and gas industry, as presented in Figure 1.

The diagram from Figure 1 emphasizes the effect of combining AI with data from the sensors installed in wells. In the intersection we see NN (neural networks), a highly used class of AI for predictions, applied to PDG data (from the sensors).

TABLE 1. Sample of dataset from single well.

Elapsed Time (hr)	Pressure (psia)	Water rate
0	2000	-6742.395
0.006059	2092.739	-6742.395
0.012118	2125.952	-6742.395
0.018177	2149.773	-6742.395
0.024235	2168.851	-6742.395

**FIGURE 2.** Sample of dataset from single well, pressure and rate with respect to time.

A successful outcome of this combination will enlarge the applicability of AI in different fields and contribute to efficient modeling in the oil and gas industry with less human interaction. Although, in this work, a specific pressure-rate data set is considered, the tested methods may be further applied to other data sets, like temperature at a specified location from PDG or distributed through the wellbore from DTS (Distributed Temperature Sensing) etc.

A. DATA

A synthetic pressure and rate dataset representing a water injection into a geological formation was simulated on a reservoir model and used in this study for testing different ML techniques. Due to the fact that the dataset is synthetic, it is clean, and there is no noise present during training and testing. The dataset contains three parameters as it follows: time, pressure and flow rate. A sample of the data is presented in Table 1.

The sample as seen in Table 1, represents the first 5 rows from the whole dataset, and it offers an insight on the range of these values. On the other hand, in Figure 2, the whole dataset is plotted, where the x axis represents the time, and the y axis the flow rate (represented with blue) and the pressure (represented with green). This plot is crucial for understanding the pressure and rate pattern, but more importantly, the response of the pressure to rate change, which is visible in the plot.

This is a typical data set, where time and pressure values are continuous, however only the starting and the ending

points of a constant rate were reported. The process of transforming the data is presented in Section V.

B. SCOPE

The focus was split on 2 types of computational experiments for the given dataset, as follows:

1) *Pressure Prediction*: when giving as input the time and the flow rate. We expect that the network should reproduce the increasing and decreasing pressure patterns.

2) *Flow Rate Prediction*: when giving as input the time and pressure. The network should reproduce a sequence of constant rate periods for the sequential pressure transients.

The same dataset was used for 2 different scenarios, by exchanging the inputs and outputs. Moreover, 2 different DL models were designed for prediction tasks, and the motivation behind these models was as follows:

1) BASED ON EXISTING RESULTS

A recent work [3] served as inspiration for the *pressure prediction*, where a DL model was successfully implemented, the Non-Linear AutoRegressive with eXogenous inputs (NARX) recurrent NN, which is in fact a DNN. Pressure prediction was successfully addressed with NARX in [3] in a combination with predicting the temperature based on the same inputs. DNNs have more complex structure than simple NNs, with layers interacting in a special way [15]. In a simple NN there are 3 layers (input, hidden and output) and all the nodes are fully connected. By adding extra hidden layers, these become DNNs.

2) IMPROVEMENT AND EXTENSION

While in the work [3] it was successful, on the dataset presented in this paper, NARX offered weaker performance than expected. As a next step, another DL model was designed, Long Short Term Memory (LSTM), from the same class, recurrent NN (RNN), in an attempt to carry out better predictions for the pressure. And second, a powerful model was needed, in order to address the “2)*Flow rate Prediction*” task. The LSTM was successfully applied in numerous energy related works [16]–[19]. RNNs represent a class of NNs where the connection between the nodes form a directed graph and these are capable of “remembering” information over time.

IV. PREDICTION METHODS

Thus, two different types of Recurrent Neural Networks (RNN) were built and tested in this study: NARX [3] and LSTM. When the current values in the dataset are strongly correlated with the past values, it is straight forward the choice of RNN for forecasting future values. Because of the directed cycle formed by the nodes in the RNN, the information is facilitated to persist in time. This aspect is important, as the information carries, over time, features and hidden patterns which may have major impact on future data points.

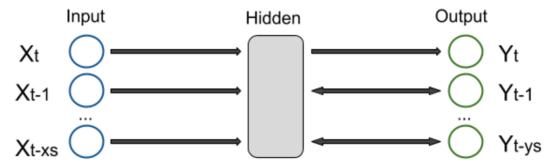


FIGURE 3. Interaction of the layers in NARX neural network.

A. NARX

In order for NARX to produce an output at any given time, it needs to receive two parameters: a new input value and previous output values (called further as “steps”). Thus, the current output represents a combination of new inputs and steps. The number of steps (one or more) can be customized for a given problem. Figure 3 is a graphical representation of the concept.

First, the inputs get transmitted to the hidden layer, then the values which are carried on by the activation function, are further sent to the output layer. To compute the current output Y_t , its previous steps are used in combination with the current inputs, as shown in the equations below:

$$\mathbf{M}_t = \mathbf{F}_H \left(\sum_{i=0}^{xs} \mathbf{X}_{t-i} \times \mathbf{U}_{hi} + \sum_{j=1}^{ys} \mathbf{Y}_{t-j} \times \mathbf{V}_{hj} + \mathbf{B}_h \right) \quad (1)$$

$$\mathbf{Y}_t = \mathbf{F}_O (\mathbf{M}_t \times \sum_{k=1}^N \mathbf{W}_{ok} + \mathbf{B}_o) \quad (2)$$

where, \mathbf{F}_H , \mathbf{F}_O are respectively the hidden layer’s activation function and the output layer’s mapping function, \mathbf{U}_{hi} , \mathbf{V}_{hj} , \mathbf{W}_{ok} , \mathbf{B}_h , \mathbf{B}_o are weights and biases corresponding to the hidden and the output layers. The limits xs and ys represent the number of inputs and respectively the number of steps transmitted to the hidden layer from both directions (input and output layers), and finally N is the number of output nodes.

As the communication of the layers is bidirectional, it computes the linear combination of the current inputs and steps, add biases and run it through \mathbf{F}_H activation function. It obtains the middle values \mathbf{M}_t , multiplies again these results with weights and add biases, and function \mathbf{F}_O will generate the outputs. Since the hidden layer’s size and the activation function is problem dependent, these will be determined in the optimization process. A generalized representation of Equation 1 and Equation 2 is written as follows:

$$\mathbf{H} = \mathbf{F}_H (\mathbf{X} \times \mathbf{U} + \mathbf{Y} \times \mathbf{V} + \mathbf{B}_h) \quad (3)$$

$$\mathbf{O} = \mathbf{F}_O (\mathbf{H} \times \mathbf{W} + \mathbf{B}_o) \quad (4)$$

In order to assess the performance of the proposed model, the Mean Squared Error (MSE) was computed as measure of effectiveness, which is the sum of squared differences between the true values and predictions, divided by the size of test set.

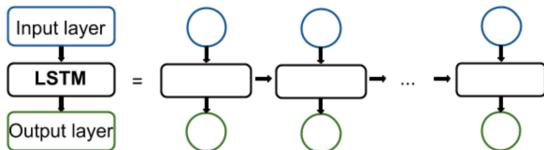


FIGURE 4. Unrolled structure of LSTM network.

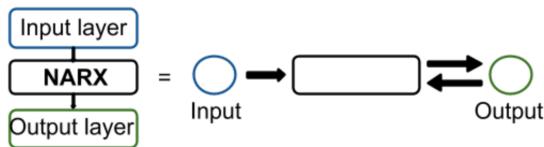


FIGURE 5. Communication between layers in NARX.

B. LSTM

LSTM is enumerated among the special types of recurrent neural networks. All RNNs contain loops which make it easy for the information to persist and be transmitted between the specified points of the network.

LSTM has been successfully used over the past years in numerous forecasting/prediction problems, therefore the aim in this work is to apply it to the oil and gas industry as well. In the Energy field, LSTM was used on solar power forecasting problems [16], together with Deep Belief Network and an AutoEncoder, and it outperformed the traditional Multi Layer Perceptron (a multi layered feed forward, back propagation neural network). In [17], on an electricity consumption forecasting problem, a variation of LSTM, named Sequence to Sequence LSTM offered better results than the standard LSTM, and the performance was comparable to that of the Restricted Boltzmann Machine in [18]. Recently, in 2018, LSTM outperformed, on an electricity consumption forecasting task, for buildings, a set of traditional ML models such as Random Forest, Ridge Regression, Artificial Neural Network and Gradient Boosting [19]. One of the reasons, for LSTM being so highly used, is that the network performs remarkably on time series data and PDG data is of same type, with the past values highly influencing the future values.

LSTMs have a chainlike structure, with four layers (functions) interacting in a specified manner. In Figure 4, the unrolled LSTM cell, with the information flow, is presented.

One of the main differences between NARX and LSTM is the communication of the input, hidden and output layers. In NARX there is a one directional information transfer from the input layer to the hidden layer and a two directional information transfer between the hidden and the output layers. Figure 5 is the graphical representation of this concept.

In the LSTM cell, the information flows in a specific way, based on the special interaction of the layers (functions) inside the cell. Some functions are designed to keep information and relevant values, others are designed to eliminate irrelevant values, all working together to decide the output in the end. This process is presented in Figure 6 and the equations below (describing the role of all four layers).

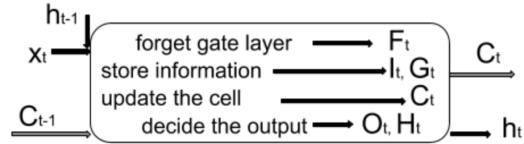


FIGURE 6. Structure of LSTM network.

In the first step the forget gate layer will neglect information from the cell state:

$$F_t = \text{sigmoid}(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{B}_f) \quad (5)$$

The store information layer will compute the input state:

$$I_t = \text{sigmoid}(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{B}_i) \quad (6)$$

Then, also in store information, tanh function will create a vector of new values, \mathbf{G}_t , to be added to the state.

$$\mathbf{G}_t = \tanh(\mathbf{W}_C[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{B}_C) \quad (7)$$

In the update layer, the old cell state \mathbf{C}_{t-1} is updated into the new cell state \mathbf{C}_t :

$$\mathbf{C}_t = \mathbf{F}_t * \mathbf{C}_{t-1} + \mathbf{I}_t * \mathbf{G}_t \quad (8)$$

To compute the output, it first runs a sigmoid function which will determine the part of the cell state that will be the output. Then the cell state is run through tanh and will be multiplied by the output of the sigmoid:

$$O_t = \text{sigmoid}(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{B}_o) \quad (9)$$

$$H_t = O_t * \tanh(\mathbf{C}_t) \quad (10)$$

where $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_C, \mathbf{W}_o$ are corresponding weights and $\mathbf{B}_f, \mathbf{B}_i, \mathbf{B}_C, \mathbf{B}_o$ are biases.

The main goal of testing these two models is to design a method for predicting flow rate based on pressure, over time. However, pressure prediction was the first scope and both will be treated in the upcoming section.

V. COMPUTATIONAL EXPERIMENTS

This section describes the types of experiments conducted and obtained results. After building the models, these will be trained and tested on different prediction tasks. There will be 2 different scenarios, by varying the inputs and outputs. Therefore, this section is divided in 3 subsections, Neural Network design - describing the process of building the models, and 2 different scenarios: Pressure Prediction and Rate Prediction, including the detailed results. For the sake of convenience, the term *rate* is used, to express *flow rate* further in the text.

A. NEURAL NETWORK DESIGN

The NARX neural network built contained an input layer, one hidden layer and an output layer. While designing the experiments it was clear that the input layer must have 2 nodes, as we will pass either $[time, rate]$ or $[time, pressure]$ as input vectors. As a baseline, the hidden layer will have 1 node,

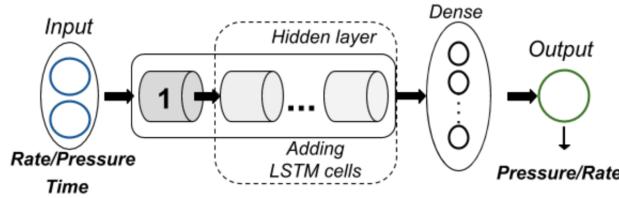


FIGURE 7. Layers in the LSTM network.

element which can be changed during the parameter optimization process. Consequently, the output layer will have 1 node, producing either pressure or rate.

Next, the LSTM neural network is built on the same basis, for the sake of comparison. All the hiper parameters, such as number of hidden layers, hidden nodes, iterations, input and output size, are set as for NARX. The scope is to conduct basic experiments with both models, in order to assess their performance. Important to mention is, that LSTM is chosen for this problem on the base of exemptional performance on other energy related time series prediction problems, such as electricity load forecasting [6].

A crucial step in preparing the dataset to be fed to the network, is normalization (scaling), which translates to bringing all values in the dataset to the same range. The advantages of normalization are presented in [20]. This is even more crucial in the present dataset, since the range of the values differ substantially, as it will be presented in the *Pressure Prediction* section. Regarding the performance of both models, another crucial step for efficiency is the parameter optimization. Common steps in the parameter optimization of both models, include the following: number of hidden layers and nodes, as described in the baseline and in Figure 7, type of activation function, and number of iterations.

The activation functions for LSTM are, by default, *sigmoid* and *hyperbolic tangent (tanh)*, and for the NARX, again the *tanh* function was chosen, for the sake of gradients. It has the benefit of faster convergence since *tanh* is a function symmetric about zero (origin), with range between $[-1,1]$ [20]. Concerning the number of iterations, for both models it started from 10 and further increased as necessary. At 10 iterations, the results still presented potential improvement, therefore, the limit was set higher to accurately spot the point where the results stop improving. In addition to these parameters, in case of NARX, a number of *steps* (defined in *NARX* subsection of the *Prediction methods* section) must be provided. This parameter was established based on trial and error, varying from 2 to 10 *steps*. The optimal number of *steps* was 5 (with 4 input values). For the implementation of LSTM a high level neural network API, Keras [21], with Tensorflow [22] backend was used.

B. PRESSURE PREDICTION

In this subsection the focus is on predicting the pressure, by using time and rate. Therefore, we have:

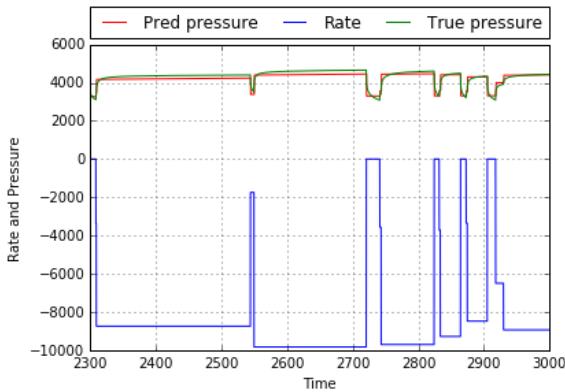
$$\mathbf{O}[t, Q] - > P \quad (11)$$

where, P is the predicted pressure, based on rate Q as a function of time t . A similar work has been conducted recently in [3], where the two main tasks were to predict *Pressure* and *Temperature*, based on *Rate* as function of *Time*. NARX was used in this work and tested on raw data and noisy data. Predicting the temperature is beyond the scope of this work, instead, the focus is on predicting the rate based on pressure as function of time. This represents a challenging task, as it will be described in the next subsection, and it was not addressed before, to the best of our knowledge, using DL. The scope is to use both models, NARX and LSTM for pressure prediction.

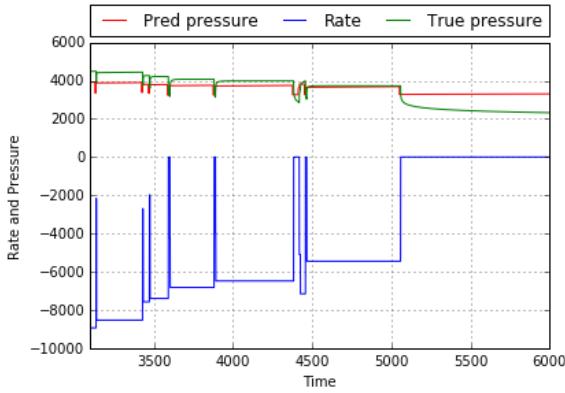
As only the starting and the ending points of rate were provided in the dataset, first interpolation was performed in order to obtain correspondence between pressure and rate points at each time t . Next, the types of pressure transients were analyzed (in terms of how the pressure values are increasing), and aimed to find similarities in specific periods. On the plot there are two different behaviors. For the first period of time, from beginning to 3000 (hr), the pressure monotonically increases with some slope for each constant rate period (a pressure transient). On the other hand, from time 3000 to 6000 (hr), the pressure grows suddenly and remains relatively constant for several transients, due to superposition effects [1] related to declining rate during this period of time. Therefore, for training and testing our models, the pre-processed dataset is used, for a period which has: first, the similar type of transient behavior (between 0 and 3000 on x axis), and second, 2 different types of transient behavior (between 0 and 3000, and between 3000 and 6000). Moreover, as seen in Table 1 and Figure 2, the range of the datapoints is extremely high (roughly in the intervals $(-10000, 0)$ for rate and $(2000, 5000)$ for pressure). Thus, it is essential to scale the values before feeding these to any network [20].

In the first case, the scope is to build and test the accuracy of NARX and LSTM for a period with same consistency. Let us call this “*Period 1*” (having similar pressure transient behavior, which means the same trend for pressure increase and decrease), where the training is on 75% of the data and predicting on remaining 25%. Training represents the ML term for learning the relationship between the pressure and flow rate datapoints, with respect to time, also called further as *history matching*, as usually used in reservoir simulations. The predicted dataset consists here of only time and rate as input, while the ‘true’ pressure is used to assess the correctness of the predicted pressure values.

In the second case, the goal is to verify the strength of the models by learning on the entire *Period 1* and predicting on *Period 2* (with a different transient behavior). In *Period 2* the pressure transients are quickly stabilized. The pressure increases quite rapidly in the beginning and then stays relatively constant throughout the build-up (unlike in *Period 1*, where the pressure change happened in a slower manner, with a lower slope). Moreover, there is a change in the rate. While in the first period it was ranging between negative 10000 and negative 8000 (excluding the shut in



(a) NARX performance - Period 1.



(b) NARX performance - Period 2.

FIGURE 8. Pressure prediction of NARX neural network on the two periods.

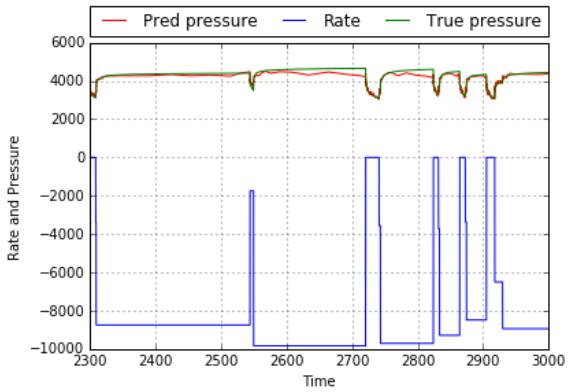
when rate is zero), in the second period it keeps decreasing (in absolute values).

1) RESULTS

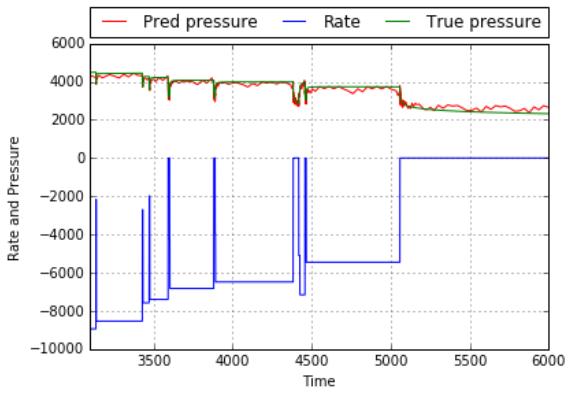
The performance of NARX in *Period 1* and *Period 2* is shown in Figures 8a and 8b respectively, and the performance of LSTM on the same periods is plotted in Figures 9a and 9b. The red points indicate the predicted pressure values on the 25% prediction dataset. The green point and the blue points represent the true pressure and rate values respectively.

There are two different behaviors observed for NARX and LSTM. The predictions made by NARX, did follow the initial pressure pattern, but these form rather straight lines (not transients). It tends to learn the increasing trend in pressure, however in Figure 8b, struggled to follow the shape of the last transient, which changed direction. On the other hand, LSTM seems to perform differently. Following the trend of the pressure transients slightly better than NARX, the predicted values are not smooth enough to simulate accurately the transients. However, LSTM presents a better performance. It outperformed NARX in both test periods, managing to learn the transient pressure trends.

At this point, NARX and LSTM had in the hidden layer 2 hidden nodes and the models were run for 50 iterations. Same experiments were also conducted after adding hidden layers



(a) LSTM performance - Period 1.



(b) LSTM performance - Period 2.

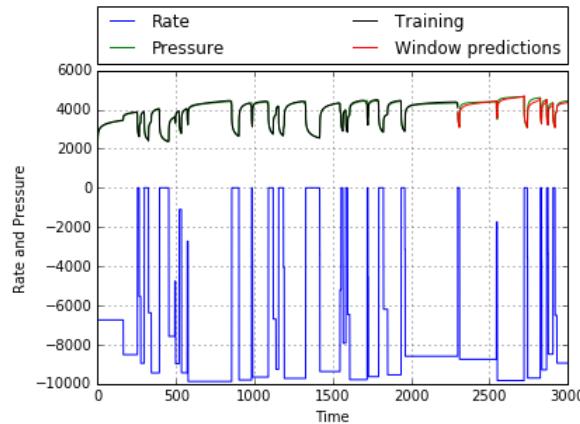
FIGURE 9. Pressure prediction of LSTM neural network on the two periods.

and nodes to the network (the network build-up is presented the design from Figure 7). Increasing the complexity of the DL models did not offer satisfactory results. Due to the small number of inputs, no more than 2 hidden nodes were necessary. Also, more hidden layers increase the risk for overfitting on a dataset of such size. In addition the learning did not improve after 50 iterations (in fact the improvement from 10 to 50 is relatively small, but still visible).

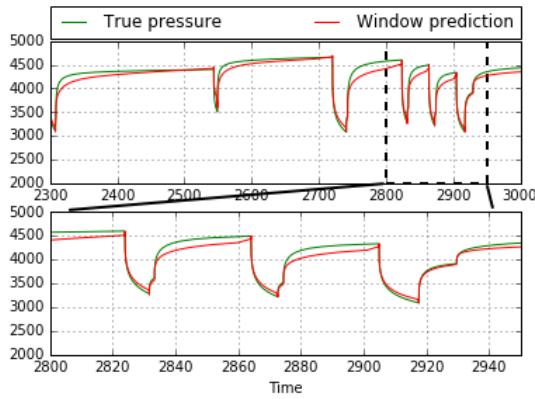
LSTM showed slightly higher performance than NARX for pressure prediction, that is visible if comparing MSE values for both periods. NARX produced an MSE equal to 0.0123 for the first period and 0.0221 for the second period, whereas LSTM produced an MSE equal to 0.0119 and 0.0214 for the first and second period respectively.

Based on the preliminary tests described above, we aim to build a more accurate model providing better prediction. LSTM was chosen for further tests, taking into account better performance and potential of transferring of the successfully implemented idea from the electricity load forecasting studies [6]. The idea is further called the *shifting window* method and it works as follows:

1. a window of 10 values is created and filled with true pressure values as starting point;
2. at each next step, one true value from the window is substituted with a predicted value from the previous step;



(a) LSTM performance - Period 1.



(b) Zoomed in interval.

FIGURE 10. Pressure prediction in Period 1, including the history matching and prediction datasets, and a close visualization of the difference between the true and predicted values.

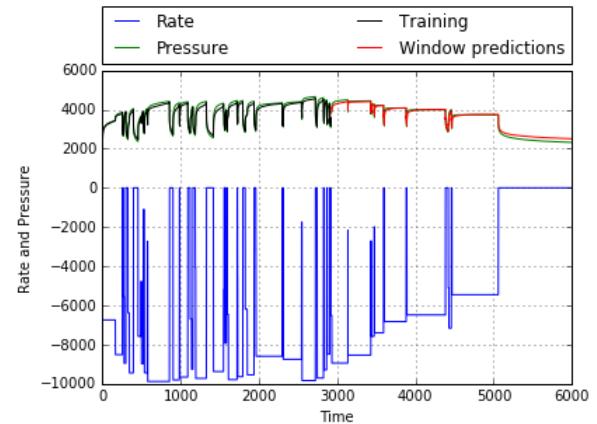
3. after 10 steps, the window consists of only predicted pressure values, and the process continues to the end of the prediction dataset.

In other words, the input is extended with one predicted pressure value every time. Moreover, for ensuring that the model carries out smooth predictions, the previous time and rate values are also added to the input.

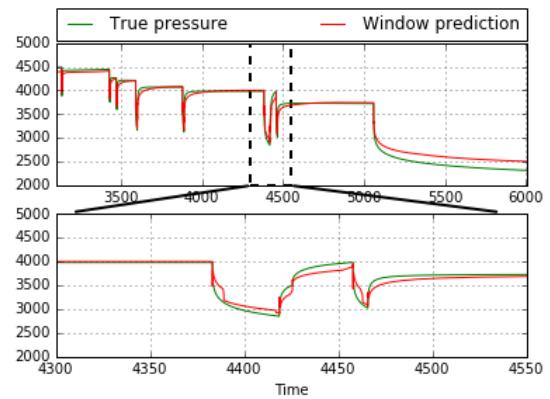
2) WINDOW PREDICTION RESULTS

The shifting window method has increased size of input data and more complexity in the training or history matching process. This resulted in a much larger neural network with the following hyper parameters: 1 hidden layer, 50 hidden nodes and 500 iterations. Naturally, the number of hidden nodes was first kept at 2 as in the initial step, then decided based on trial and error method.

The results are presented in Figures 10 and 11 for *Period 1* and *Period 2* respectively. In Figure 10a, the true pressure is represented in green, the corresponding flow rate in blue, the history matching in black and the predicted pressure values, using the shifting window method in red. The improvement is first visible in the pattern of the predicted values. In addition, the closeness of these values to the true pressure is



(a) LSTM performance - Period 2.



(b) Zoomed in interval.

FIGURE 11. Pressure prediction in Period 2, including the history matching and prediction datasets, and a close visualization of the difference between the true and predicted values.

further revealed in Figure 10b, by using a zoom of a random short period. Similarly, Figures 11a and 11b, designate the aforementioned characteristics for *Period 2*.

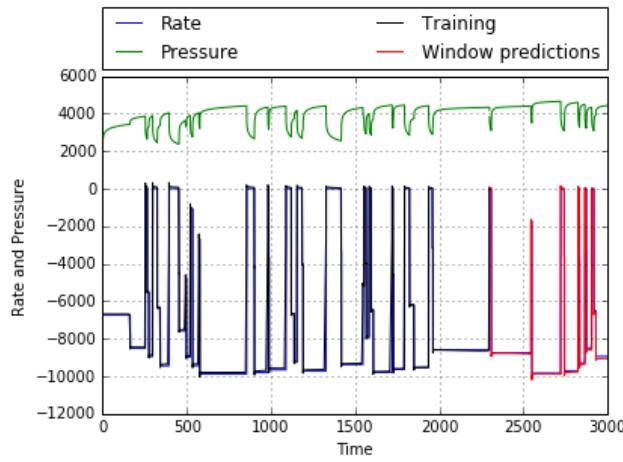
The MSE in these cases are 0.001 and 0.0011 for *Period 1* and *Period 2* respectively. For the sake of comparison, the error values during the training period are represented in Figure 14, for the best (50 hidden nodes) and worse (2 hidden nodes) cases.

C. FLOW RATE PREDICTION

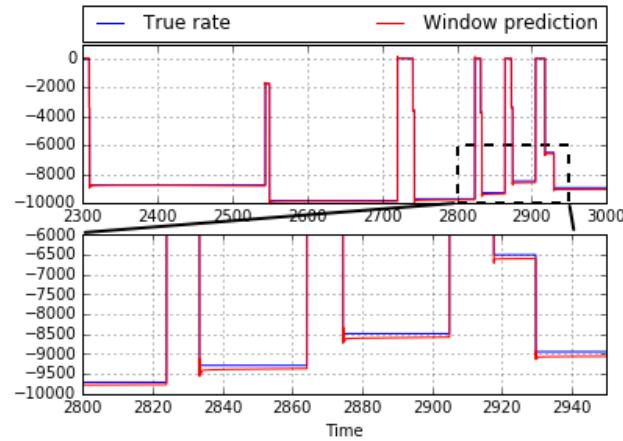
This subsection represents an inverse of the previous task and it is also more challenging. Here, the rate is predicted based on time and pressure, using only the LSTM neural network, due to its performance on pressure prediction and previous successful experiences in the energy field. Therefore, we have:

$$\mathbf{O}[t, P] \rightarrow Q \quad (12)$$

which represents that the rate is a function of time and pressure. The shifting window method was proven to be effective for the pressure prediction task, therefore it is adopted for rate prediction in similar manner.



(a) LSTM performance - Period 1.



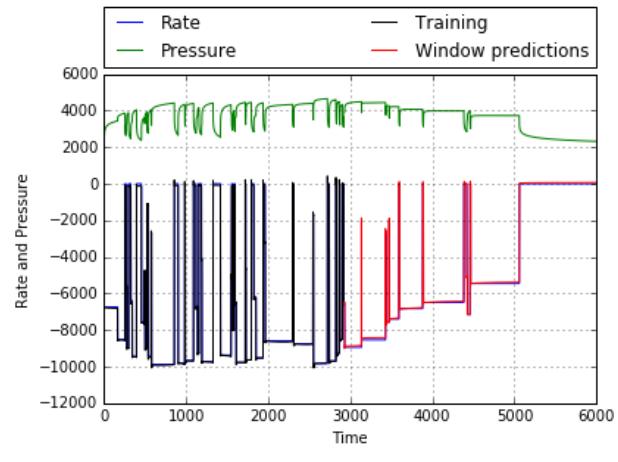
(b) Zoomed in interval.

FIGURE 12. Flow Rate prediction in Period 1, including the history matching and prediction datasets, and a close visualization of the difference between the true and predicted values.

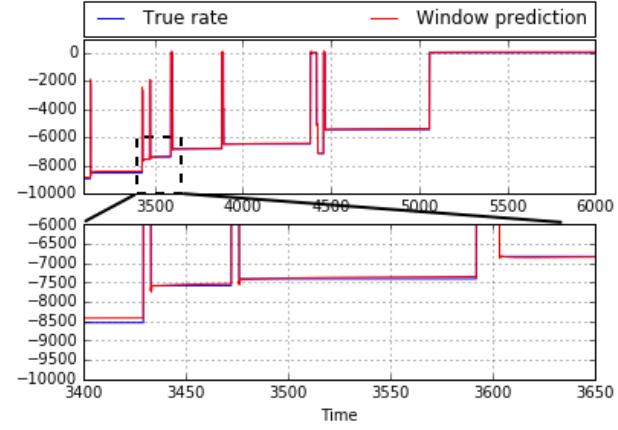
By analyzing the rate data in Table 1 and Figure 2, it is noticeable that the values are constant for a specific period of time and even when it changes, it goes to constant for another specific period. Whereas the pressure is either constantly increasing or decreasing over the whole time, sometimes slowly and other times suddenly, as it was shown in the previous subsection. As another observation on the dataset, if we analyze *Period 1*, we notice that the constant rate intervals for pressure build ups are shorter than in *Period 2*.

1) WINDOW PREDICTION RESULTS

For consistency, the experiments are conducted similarly to the pressure predictions above. First, LSTM is run on *Period 1*, keeping the 75% of data for history matching and 25% for predictions. Then, as in the previous case, LSTM is learning on the entire *Period 1* and predicting on the entire *Period 2*, for robustness. As a baseline, the model has only 1 hidden layer and 2 nodes, as for pressure prediction, then the model is built up, as seen in Figure 7, and the final structure of the neural network is established. The first objective is to determine whether the same network structure, from



(a) LSTM performance - Period 2.



(b) Zoomed in interval.

FIGURE 13. Flow Rate prediction in Period 2, including the history matching and prediction datasets, and a close visualization of the difference between the true and predicted values.

pressure prediction, carries out satisfactory results in the rate prediction as well. If not, changes will be made in the model parameters.

After the baseline experiment is run, the model with 1 hidden layer and 2 hidden nodes is not accurate enough (although it learned the pattern in the dataset). Such situation was expected, due to the different nature and size of the input (as seen from true data, pressure has transient trend with many values for a constant rate period).

Therefore, the number of hidden nodes is increased to 50, as for pressure prediction. However, the results still show potential improvement, as the predicted values are quite far from the true rate. Consequently, the number of nodes is further increased to 100 and 200, in order to seek for more accurate predictions. All the models, with varying number of hidden nodes, are run to be able to verify where the prediction accuracy stops improving. Meanwhile, the aim is to observe how other parameters, such as number of iterations, need to be adjusted.

Figures 12a and 12b present the best performance of LSTM on rate prediction, for *Period 1*. For being more space

TABLE 2. Summary of the results for LSTM, for predictions in both Period 1 and Period 2. In both cases the best model runs with 200 nodes and the decreasing of MSE (final error) is represented over changing the number of hidden nodes.

Period 1						
History matching	Predictions	MSE at 500 iterations				
		2 nodes	10 nodes	50 nodes	100 nodes	200 nodes
0 - 2297.77	2297.78 - 3000	0.049	0.043	0.037	0.034	0.033

Period 2						
History matching	Predictions	MSE at 500 iterations				
		2 nodes	10 nodes	50 nodes	100 nodes	200 nodes
0 - 2918.08	2918.13 - 6000	0.030	0.023	0.020	0.020	0.019

efficient, we only plot the best case, which was proven to be with 200 nodes, and compare the prediction loss later on in Figure 14. In Figure 12a we have the true rate values (blue), the history matching (black) and the prediction values (red), as well as the pressure (green) for reference points. However, in Figure 12b there are zoomed rates only for better visualization. The reason for choosing that specific period for zooming in is that it contains multiple different rate observations, so that it makes it easier to read the plot.

Similarly, all the experiments are run on *Period 2*. Interestingly, this validates the model for the previous period, with 200 hidden nodes, however, the final error, MSE decreases slower when increasing the number of nodes from 2 to 50, 100 and finally 200. All errors occurred, at 500 iterations, while changing hidden nodes number are presented in Table 2.

The results for *Period 2* are presented in Figures 13a and 13b, in the same manner as Figure 12. It is noticeable that for some pressure transients the model produced quite accurate predicted values, however, for other transients the predictions and the true values are more easily distinguished from each other. This is more visible in Figures 12b and 13b, for both periods.

Figure 14 presents the loss during the training period for the minimum and maximum number of hidden nodes. The difference lies in how rapidly MAE (Mean Absolute Error - loss function used during optimization) is decreasing, over same number of iterations. For verifying the accuracy of final rate predictions, MSE was computed, which, for *Period 1* is 0.049 (2 nodes) and 0.033 (200 nodes), and for *Period 2* is 0.030 (2 nodes) and 0.019 (200 nodes).

In Figure 14a, all loss values from the training are plotted for *Period 1*. In case of 2 nodes, it is noticeable that for both pressure and rate, the loss decreases more slowly. While increasing the number of nodes to 50 for pressure and 200 for rate, the model starts to learn faster, the error decreasing more drastically before 100 iterations and slowly improving until the end. The same statement stands for Figure 14b, which naturally represents the training loss values for *Period 2*.

Consequently, for the present dataset, we can affirm that: LSTM with 1 hidden layer and 50 nodes, for pressure, and LSTM with 1 hidden layer and 200 nodes, for rate, are the

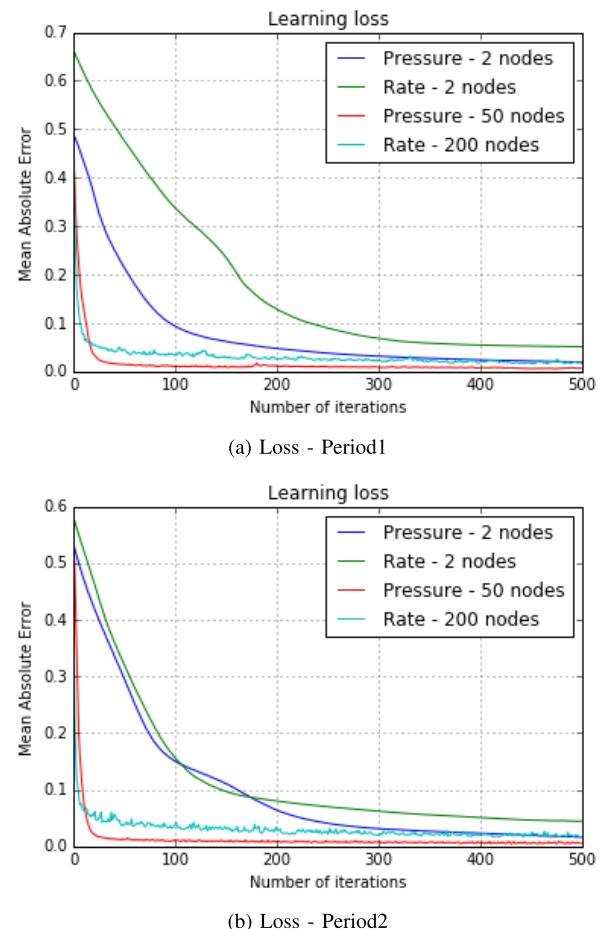


FIGURE 14. Learning loss of LSTM on both periods (plotted for 2 nodes and the best case, 50 and 200 nodes, for pressure and rate respectively).

most accurate. Adding more nodes or even layers to this network doesn't seem to improve accuracy.

Table 2 presents a summary of LSTM's performance on both *Period 1* and *Period 2*. It is structured as follows: the top half and the bottom half of the table summarize *Period 1* and *Period 2* respectively; for both cases, the first column indicates the time duration of the history matching (training) dataset, the second column represents the time duration

of the prediction period (testing), and the remaining five columns contain the MSE values for all tested scenarios, with 2, 10, 50, 100, and 200 hidden nodes in LSTM. The smallest error, at 200 nodes, is represented in bold on blue background.

Analyzing Figure 14 and Table 2, a general conclusion is that increasing the number of nodes in the LSTM training gives much faster learning (fast reduction of Mean Absolute Error, MAE) and much lower error for the same iteration (Figure 14), while predictions (Table 2) are less sensitive to this number. The best LSTM network (based on MAE curve) with 200 nodes learned quite fast providing reasonable results after 20 iteration, while showing minor MAE decline afterwards. A balance between number of nodes and the iterations required to achieve requested error level is subject to further studies. Another interesting observation is that the pressure prediction task requires a network with less number of nodes comparing to the rate prediction task. In practical applications, attention may be paid to assembling neural networks with balancing between number of nodes and iterations, which will be related with size and nature of data sets in focus, like in this work it was shown on comparison of pressure and rate predictions.

While in general, LSTM is successfully applied on large datasets, in the presented work, it was chosen due to its ability to process complicated data. It was able to establish complex relationships between the pressure and rate points, more importantly, predict rate based on pressure, which consequently means that by increasing the number of iterations and the depth of the layers, it can be applied to real case PDG data.

Finally, with this performance, LSTM further extends the applicability of DL in all the industries operated with wells, the goal mentioned in Section III. *Workflow*, and sketched in Figure 1. LSTM demonstrates efficient handling of pressure and flow rate data from PDG sensors, and it can be customized for other sensor data in the industries. This way, modeling becomes more automated allowing the user to control pressure and rate based on the knowledge gained from the designed model, facilitating decision making.

VI. CONCLUSION

The purpose of this work was to test and further develop Deep Learning (DL) techniques to be applied for filling data gaps and predictions of well behavior in the upstream petroleum and other industries. Synthetic pressure and rate data, as measured with Permanent Downhole Gauges (PDG) installed in injection (and production) wells, was used as an example data set and two tasks of well pressure prediction based on specified rates and rate prediction based on pressure were addressed. Pressures and rates are interconnected parameters and physical models may be used to relate them, like physical-based simulation was used to generate the synthetic data set. In this study, DL techniques were solely employed to reveal such interconnection without physical models (data-driven analysis). The NARX and LSTM neural networks

were assembled and tested on pressure prediction. Although, LSTM slightly outperformed NARX, neither model offered the expected performance in the first round.

The LSTM network has been proven to successfully generate predictions in the energy consumption field, early addressed by the authors [6], where *shifting window* method has improved the network performance. In this work, LSTM's performance was also further improved by adopting *shifting window* method: a moving window with size of 10 values was created in addition to the ordinary input to address transition from training to prediction and further on, where predicted values for previous steps were accounted for while predicting the value at a new time step. Including predicted values to the input has proven to be successful, and as a result LSTM managed to follow the pressure and rate patterns.

The main contribution of the paper is an efficient application of LSTM to rate prediction, helping in filling the gaps in rate data, which is still a challenge in well monitoring and performance analysis. The DL based pressure predictions, progressed in this paper, may be further used to forecast short-term well performance. Both tasks are crucial for supporting decision making in the upstream petroleum industry. In future, experiments with larger datasets and parameter optimization would contribute to understanding of how ML and DL may be universally applied to a diversity of wells and, furthermore, to a diversity of sensors. Integrating physical models into DL frameworks may help in constraining DL output and facilitating better performance via reflecting physical principles in the network structure / interconnections.

REFERENCES

- [1] Bourdet, *Well Test Analysis: The Use of Advanced Interpretation Models*. Amsterdam, The Netherlands: Elsevier, 2002.
- [2] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: Principles and methods," *Data Knowl. Eng.*, vol. 25, nos. 1–2, pp. 161–197, Mar. 1998
- [3] C. Tian and R. N. Horne, "Recurrent neural networks for permanent downhole gauge data analysis," in *Proc. Soc. Petroleum Eng.*, Oct. 2017, pp. 2, 3, 4 and 6. doi: [10.2118/187181-MS](https://doi.org/10.2118/187181-MS).
- [4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, MA, USA: MIT Press, 2016.
- [5] A. Shchipanov, R. Berenblyum, and L. Kollbotn, "Pressure transient analysis as an element of permanent reservoir monitoring," in *Proc. Soc. Petroleum Eng.*, Jan. 2014, p. 2. doi: [10.2118/170740-MS](https://doi.org/10.2118/170740-MS).
- [6] C. Hegedus, A. Chakravorty, and C. Rong, "Energy load forecasting using deep learning," in *Proc. IEEE Int. Conf. Energy Internet (ICEI)*, May 2018, pp. 146–151.
- [7] M. A. Proett, S. M. Ma, N. M. Al-Musharfi, and M. Berkane, "Dynamic data analysis with new automated workflows for enhanced formation evaluation," in *Proc. Soc. Petroleum Eng.*, Oct. 2017, p. 2. doi: [10.2118/187040-MS](https://doi.org/10.2118/187040-MS).
- [8] N. Myakeshev, A. Aslanyan, R. Farakhova, and L. Gainutdinova, "Carbonate reservoir waterflood efficiency monitoring with cross-well pulse-code pressure testing," in *Proc. Soc. Petroleum Eng.*, Nov. 2017, p. 2. doi: [10.2118/189258-MS](https://doi.org/10.2118/189258-MS).
- [9] A. C. Gringarten, "From straight lines to deconvolution: The evolution of the state of the art in well test analysis," in *Proc. Soc. Petroleum Eng.*, Feb. 2008, p. 2. doi: [10.2118/102079-PA](https://doi.org/10.2118/102079-PA).

- [10] C. Tian and R. N. Horne, "Applying machine learning techniques to interpret flow rate pressure and temperature data from permanent down-hole gauges," in *Proc. Soc. Petroleum Eng.*, Apr. 2015, p. 2. doi: [10.2118/174034-MS](https://doi.org/10.2118/174034-MS).
- [11] C. Tian and R. N. Horne, "Machine learning applied to multiwell test analysis and flow rate reconstruction," in *Proc. Soc. Petroleum Eng.*, Sep. 2015, p. 2. doi: [10.2118/175059-MS](https://doi.org/10.2118/175059-MS).
- [12] A. U. Al-Kaabi and W. John Lee, "Using artificial neural networks to identify the well test interpretation model (includes associated papers 28151 and 28165)," *SPE Formation Eval.*, vol. 8, no. 3, pp. 233–240, Sep. 1993.
- [13] M. A. Sultan and A. U. Al-Kaabi, "Application of neural network to the determination of well-test interpretation model for horizontal wells," in *Proc. SPE Asia Pacific Oil Gas Conf. Exhib. Soc. Petroleum Eng.*, Jan. 2002, p. 3.
- [14] R. Ahmadi, S. Jamal, and A. Babak, "Automatic well-testing model diagnosis and parameter estimation using artificial neural networks and design of experiments," *J. Petroleum Explor. Prod. Technol.*, vol. 7, no. 3, pp. 759–783, Sep. 2017.
- [15] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
- [16] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM neural networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2016, Art. no. 002858.
- [17] D. L. Marino, K. Amarasinghe, and M. Manic, "Building energy load forecasting using deep neural networks," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2016, pp. 7046–7051.
- [18] E. Mocanu, P. H. Nguyen, M. Gibescu, and L. Wil Kling, "Deep learning for estimating building energy consumption," *Sustain. Energy, Grids Netw.*, vol. 6, pp. 91–99, Jun. 2016.
- [19] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," *Energies* vol. 11, no. 7, p. 1636, Jun. 2018.
- [20] Y. A. LeCun, L. Bottou, G. B. Orr, and K. R. Müller, *Efficient Backprop. in Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 9–48.
- [21] F. Chollet et al. (2015). *Keras*. [Online]. Available: <https://keras.io>
- [22] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, *arXiv:1603.04467*. [Online]. Available: <https://arxiv.org/abs/1603.04467>



CRISTINA HEGEDUS received the B.S. and M.S. degrees in mathematics and informatics with the focus on statistics, probability, analysis, and different kinds of programming. She is currently pursuing the Ph.D. degree in data science with the Department of Computer Science and Electrical Engineering, University of Stavanger, Norway. She was a Mathematics and Informatics Teacher for secondary school, for a period of four years. Her main research interest includes energy informatics and her scientific papers are centered around efficient public transit, electrical energy supply and demand, as well as automation in the oil and gas industry.



ANTON SHCHIPANOV received the M.Sc. degree in applied mathematics and the Ph.D. degree in reservoir simulation from Perm State University. His research interests include reservoir characterization and simulation, pressure transient and well performance analyses, naturally and induced fractured reservoirs, improved oil recovery, and carbon dioxide utilization and storage. He leaded Research and Development and industry-oriented projects in Norway and was involved as a Researcher and a Work Package Leader with large-scale Norwegian and EU Research and Development projects.



CHUNMING RONG was the Vice President of CSA Norway Chapter, from 2015 to 2016. He is currently the Co-Chair of the IEEE Blockchain, the Chair of the IEEE Cloud Computing. He is also the Head of the Center for IP-based Service Innovation (CIPSI) with the University of Stavanger (UiS) and the Adjunct Chief Scientist leading Big-Data Initiative with IRIS. His research interests include data science, cloud computing, security, and privacy. He has been a member of the Norwegian Academy of Technological Sciences (NTVA), since 2011. He has extensive contact network and projects in both the industry and academic. He is also a Founder and the Steering Chair of the IEEE CloudCom conference and workshop series. He is the Steering Chair and an Associate Editor of the IEEE TRANSACTIONS ON CLOUD COMPUTING (TCC), and the Co-Editor-in-Chief of the *Journal of Cloud Computing* (ISSN: 2192-113X) by (Springer). He has extensive experience in managing large-scale Research and Development projects funded by both industry and funding agencies, both in Norway and EU.