

Multilabel Classification for Inflow Profile Monitoring*

Dmitry I. Ignatov¹[0000-0002-6584-8534] dignatov@hse.ru, Pavel Spesivtsev² PSpesivtsev@slb.com, Dmitry Kurgansky¹ mykurgansky@mail.ru, Ivan Vrabie^{1,2} vrabie93@mail.ru, Svyatoslav Elizarov¹ sorkerrer@gmail.com, and Vladimir Zyuzin^{2,3} VZyuzin@slb.com

¹ National Research University Higher School of Economics, Moscow, Russia

² Schlumberger Moscow Research, Moscow, Russia

³ Moscow Institute of Physics and Technology, Moscow, Russia

Abstract. The purpose of this study is to identify the position of non-performing inflow zones (sources) in a wellbore by means of machine learning techniques. The training data are obtained using the transient multiphase simulators and represented as the following time-series: bottom-hole pressure, well-head pressure, flowrates of gas, oil, and water along with a target vector of size N , where each element is a binary variable indicating the productivity of the respective inflow zone. The goal is to predict the target vector of active and non-active inflow sources given the surface parameters for an unseen well. A variety of machine learning techniques has been applied to solve this task including feature extraction and generation, dimensionality reduction, ensembles and cascades of learning algorithms, and deep learning. The results of the study can be used to provide more efficient and accurate monitoring of gas and oil production and informed decision making.

Keywords: Multi-phase flow, multilabel classification, time series, bottomhole pressure

1 Introduction

During the production phase of oil and gas wells it often happens that oil does not enter every inflow point, which leads to a decrease in the efficiency of the operation and undesired economic consequences⁴. It is beneficial to determine which of the inflow points are inactive to properly design the intervention operations. The main research hypothesis here is as follows: using the machine learning approaches, the active and non-active inflow points can be predicted based on the measurements of certain parameters at the wellhead, including pressure and total gas and oil productivity.

* Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

⁴ This paper continues our research “Development of Data Analytics Algorithms for Predicting the Parameters of Oil and Gas Well Flows” [8, 5]

The paper is organized as follows. In Section 2 we formulate the studied problem as a multilabel classification. Sections 3 and 4 explain the data generation process and detail the performed data transformations, respectively. Section 5 describes the time-series specific feature extraction process. Section 6 presents the obtained classification results along with feature importance estimation. Section 7 concludes the paper.

2 Problem Statement

The problem of inflow profile monitoring can be formulated as follows.

There are descriptions of objects $X \subseteq \mathbb{R}^d$, where d is the size of feature space, and a finite set of class labels $Y \subseteq \{0, 1\}^L$. A finite training set of observations is given as follows:

$$\{x^{(i)}, y^{(i)}\}_{i=1}^N, \text{ where}$$

$x^{(i)} = (x_1, \dots, x_d) \in X$ is the description vector of i -th object (one measurement), $y^{(i)} = (y_1, \dots, y_L) \in Y$ is the label vector with

$$y_j = \begin{cases} 1, & \text{if there is an oil inflow at } j\text{-th position} \\ 0, & \text{otherwise.} \end{cases}$$

However, in our case, the description vector $x^{(i)}$ can be recast as containing time series of d sensors within a certain time interval $T = \{1, 2, \dots, t\}$:

$$x^{(i)} = ((x_1, \dots, x_t)_1, \dots, (x_1, \dots, x_t)_d) \in \mathbb{R}^{d \times t}.$$

Using a training set $s_t = \{x^{(i)}, y^{(i)}\}_{i=1}^N$, it is necessary to construct a mapping function (classifier):

$$h : X \rightarrow Y$$

For each test instance $\tilde{x} \in X$, we get a prediction: $\hat{y} = h(\tilde{x})$.

Thus, the problem of multilabel classification is to be solved, in which the object can belong to several classes at the same time, and the classes are not mutually exclusive. For example, this type of problem arises in text mining, namely in automatic tag's assignment, text categorization and classification, similarly for categorization of images, etc. Multilabel classification is an extension of the traditional classification problem with several classes, i.e. multi-class problem. Approaches to solve this problem are mentioned in section 6 and can be partially found in [10, 7].

3 Data Generation

The training data are obtained as a result of numerical simulations that describe the physical processes taking place in wells [9]. For the given input parameters (wellbore geometry, initial distribution of volume fractions of phases,

pressure in the wellhead, choke size, etc.), the simulator models the behavior of the wellbore for a given time interval T and generates the following time series:

- $BHP(t)$ is the bottomhole pressure (measured at the source closest to the surface);
- $WHP(t)$ is the wellhead pressure;
- $Q_o(t)$ is the surface oil flowrate;
- $Q_w(t)$ is the surface water flowrate;
- $Q_g(t)$ is the surface gas flow rate.

The target vector y of length 20 is generated randomly and consists of ones and zeros, characterizing the presence or absence of inflow in one of the 20 pre-defined inflow points along the wellbore. In the present work, 5000 simulation realizations are used.

4 Data preparation

Given that each time series is large and has complex structure, which may carry latent complex patterns, it is necessary to transform it to a smaller space of more informative features than only the values of the series at a certain time-steps t . For example, one can extract minimal and maximal values, the number of local maxima and minima (“peaks”), take the average and median values, etc. In addition, many machine learning algorithms are sensitive to data scaling. Such algorithms, for example, include nearest neighbor method, Support Vector Machine, etc. In this study, we will use two common types of data normalization: normalization by standard deviation and the Min-Max normalization. Another important task is to reduce the dimension of the feature space using different methods, and we will examine the most popular ones, such as:

1. Principle Component Analysis (PCA)
2. Independent Component Analysis (ICA)
3. Truncated Singular Value Decomposition (TSVD).

Hence, the original task is divided into two subtasks:

1. Determination of the appropriate feature space $X' \subseteq \mathbb{R}^k$.
2. The choice/tuning of the optimal classifier h .

The average size of 0/1-loss on the test sample of size M is used as a quality criterion. To characterize the average prediction accuracy of each inflow point one can consider the whole vector of 0/1-loss for all inflow points. Thus, in our experiments the averaged accuracy of an inflow point at different positions varies showing higher values for several first positions (closer to the surface, see Fig. 1).

5 Feature extraction from time series

The set of predictors for training in the initial sample is represented by time series, from which it is possible to extract a set of additional parameters that can positively affect the quality of algorithms [3].

Fourier transform is one of the basic tools in signal analysis. This transform allows to move from time domain to frequency domain, that is, to get rid of the signal shifts in time. Discrete Fourier Transform (DFT) is used for discrete signals.

An alternative to the Fourier transform is the wavelet transform, which is a convolution of the wavelet function to the signal. The wavelet transform translates the signal from the time representation to its time-frequency representation. For discrete signals, a discrete wavelet transform is applied by a set of filters. First, the signal is passed through a low-frequency filter (LF-filter) with a pulse response g :

$$\hat{s}[n] = \sum_{k=-\infty}^{+\infty} s[k]g[n-k]$$

At the same time, the signal is similarly decomposed using a high-frequency filter f (HF-filter). The result contains detailed coefficients (after the HF-filter) and approximation coefficients (after the LF-filter). After completing the procedure the samples of the signals are downsampled by a factor of 2.

Different output values of linear regression were also used as features. In our case, we used a sample from a time series as a predictor, and a discrete sequence from 0 to a number equal to the length of the sample minus 1 as the target variable.

Another attribute is the mean squared of the time series, which is given below:

$$E = \sum_{i=1, \dots, n} x_i^2.$$

The average absolute change was also taken into account, which is simply the following:

$$\frac{1}{n} \sum_{i=1, \dots, n-1} |x_{i+1} - x_i|.$$

Among many more parameters that can be used to enlarge the feature space are average, standard deviation, median, dispersion, min/max value, trend, number of min/max values, lower/upper quartile, and last position of min/max value.

All the aforementioned features in this section can be calculated by specialized Python libraries. Here we have used tsfresh library [2] and produced more than 1200 features⁵.

⁵ The full list of possible features to extract can be found by the link https://tsfresh.readthedocs.io/en/latest/text/list_of_features.html

6 Experiments

To conduct experiments with the data obtained by the simulators, a set of 5000 numerical simulations was generated, for each of which there are indications of 4 different sensors that produce measurements for 3600 seconds with a sampling rate of 1 Hz. The average 0/1-loss for each inflow point (or averaged by all of them) on the test sample is used as a quality criterion. The split into training and test sets was made by randomly sampling generated observations in the ratio of 4:1.

The first experiment was to test the approach of independent classifiers, separately for each of the 20 sources. In addition to selection of the optimal classifier, it is necessary to correctly determine the appropriate feature space X' . For this purpose, many different methods of dimensionality reduction and normalization have been tested both for the initial data and for the extracted time series features. Every dimensionality reduction method was tested on the following set of classification algorithms: Random Forest (RF), SVM, kNN, XGBoost [1]. The mean 0/1-loss varies from 0.36 to 0.39.

The best algorithm was XGBoost with PCA over z-score normalization of features obtained from the time series. The same combination of the dimension reduction method and the algorithm, but with min-max normalization resulted in the third best performance.

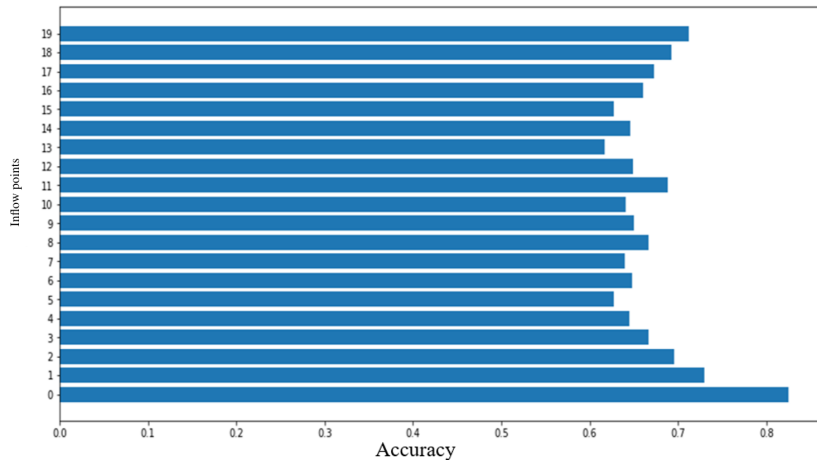


Fig. 1: Average prediction accuracy for each of the inflow points

Experiment 2 was to build an ensemble of the top 10 of the best performing algorithms and determine the label by majority voting. As it had been expected, the results were slightly better, the average value of the loss function 0/1 was equal to 0.31.

During the third experiment aimed at testing the approach of classifier chains [7], a correlation matrix was built between the values of all sources. By chain of classifiers, Read et al. [7] mean a simple classifier cascade where after prediction of the first component of a target vector, the second component is predicted on the same set of features plus the prediction for the first component (or its known value for training data) as an extra feature, and similarly for the sequence of the remaining components. In the resulting matrix there were no correlation greater than 0.1, so the option of building classifier chains would not bring significant improvement in quality.

The fourth experiment was originally to predict the number of active inflow zones. For each sample in the available training data, the number of active inflow zones was counted and the task of multiclass classification was compiled. The prediction accuracy was 1. Having received such a good result, it was proposed to build a version of the cascade classifier, working on the following scheme:

1. We predict the number of working sources.
2. We obtain the probabilities of class 1 for each source separately.
3. Sort the probabilities in descending order.
4. Get the number of sources equal to one at different probability thresholds (calibration step).
5. If the number of sources labeled by “1” (i.e., active sources) at a given probability threshold is greater than the predicted number of sources, then put the label “0” for the sources whose probability is the lowest until the number of active inflow points (predicted working sources) becomes equal to their predicted number.

However, this algorithm not only did not reduce the $\{0, 1\}$ -loss function more than the ensemble, but significantly increased it to 0.44. This can be explained by the fact that in the current scheme of the cascade algorithm, we did not process the option when the number of sources is less than that of the predicted ones. In addition, a significant part of the probabilities of sources to belong to class 1 is very similar, which does not allow one to exclude only the wrong values.

The fifth experiment was designed to use both the initial data and the extracted features from the initial data. The XGBoost method was chosen as an algorithm, the following set of features was used as a feature space:

- 300 ICA components applied on the training set transposed time series normalized by Z-score;
- 300 PCA components applied to more than 1200 features extracted from the time series;
- the number of working sources for this simulation (can be predicted by a simple binary classifier, e.g., logistic regression);

The result of this method was the reduction of the loss function error to 0.26, which is the best result in this study.

For the sake of comparison, a series of experiments using deep neural networks was conducted in Keras over Tensorflow. We used both LSTM ([4]) and

CNN networks ([6]) as well as their mixture over all 5000 examples given as normalized and concatenated time-series in 4500/500 learning scenario for training and validation. The highest validation accuracy was about 0.59.

7 Conclusion

We considered and tested methods of extracting significant features from multivariate time series, methods of data normalization and dimensionality reduction. Several basic algorithms and their ensembles were tested as well as a cascade of two classification algorithms was proposed and applied.

The best result, 0.26, in terms of average $\{0, 1\}$ -loss was shown by the XGBoost method with specially constructed sets of features.

The results of our experiments are summarized in Table 1

Table 1: Experiment results

Method	$\{0, 1\}$ -loss
XGBoost with PCA	0.39
Ensemble of 10 algorithms	0.31
Cascade classifier	0.44
XGBoost+PCA+ICA+working sources	0.26
CNN+LSTM	0.41

Our analysis demonstrates that inflow profile monitoring using surface measurements is a challenging problem. However, the combination of machine learning techniques allows to get results significantly better than random guess. We hope that further enhancement of specially designed methods based on classifier ensembles, relevant deep neural networks architectures and times-series features extraction techniques may further improve the quality of multi-label prediction in the studied problem.

Acknowledgments The work of Dmitry Ignatov (Sections 2, 6, and 7) was supported by the Russian Science Foundation under grant 17-11-01294 and performed at National Research University Higher School of Economics, Russia.

References

1. Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794, 2016.

2. Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. Time series feature extraction on basis of scalable hypothesis tests (tsfresh a python package). *Neurocomputing*, 307:72 – 77, 2018.
3. Marco Fagiani, Stefano Squartini, Leonardo Gabrielli, Marco Severini, and Francesco Piazza. A statistical framework for automatic leakage detection in smart water and gas grids. *Energies*, 9(9), 2016.
4. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.
5. Dmitry I. Ignatov, Konstantin Sinkov, Pavel Spesivtsev, Ivan Vrabie, and Vladimir Zyuzin. Tree-based ensembles for predicting the bottomhole pressure of oil and gas well flows. In Wil M. P. van der Aalst et al., editor, *Analysis of Images, Social Networks and Texts - 7th International Conference, AIST 2018, Moscow, Russia, July 5-7, 2018, Revised Selected Papers*, volume 11179 of *Lecture Notes in Computer Science*, pages 221–233. Springer, 2018.
6. Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. In Michael A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 255–258. MIT Press, Cambridge, MA, USA, 1998.
7. Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.
8. Pavel Spesivtsev, Konstantin Sinkov, Ivan Sofronov, Anna Zimina, Alexey Umnov, Ramil Yarullin, and Dmitry Vetrov. Predictive model for bottomhole pressure based on machine learning. *Journal of Petroleum Science and Engineering*, 166:825 – 841, 2018.
9. Pavel E. Spesivtsev, Andrey D. Kharlashkin, and Konstantin F. Sinkov. Study of the transient terrain-induced and severe slugging problems by use of the drift-flux model. *SPE Journal*, 22(SPE-186105-PA), 2017.
10. Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *IJDWM*, 3(3):1–13, 2007.