The method of calculation the pressure gradient in multiphase flow in the pipe segment based on the machine learning algorithms

E Kanin[1], A Vainshtein[1], A Osiptsov[1] and E Burnaev[1]

[1]Skolkovo Institute of Science and Technology (Skoltech), 3 Nobel Street, 143026, Moscow, Russian Federation

E-mail: Evgenii.Kanin@skoltech.ru

Abstract. Engineering tools allowing pressure gradient calculation in the pipe segment commonly use stationary correlation and mechanistic models such as Beggs and Brill, Ansary, etc. This is well known and convenient way which gives rough estimate of pressure gradient due to friction losses and liquid phase interference. It avoids solving complex dynamic pressure equation and derives quick results with comfortable precision margin for large scale systems, such as horizontal pipes and wells. In order to enlarge the applicability zone and accuracy of existing methods, a new method of pressure gradient definition is evolved. It is included three surrogate models that are based on Machine Learning (ML) algorithms. The first model predicts liquid holdup in the segment, the second defines flow pattern and the third predicts pressure gradient. In order to create these models, several ML algorithms are applied such as Random Forest, Gradient Boosting, Support Vector Machine and Artificial Neuron Network.

Involvement of the latest machine learning algorithms will allow applying this method to wider range of input data compared with standard multiphase flow correlations and mechanistic models. The proposed method demonstrates high accuracy – on the collected experimental data set it gives $R^2 = 0.985$ for pressure gradient prediction. That is why it could help to carry out correct calculation of bottom hole pressure and pressure distribution along the length of the pipeline.

1. Introduction

Multiphase flow is simultaneous flow in pipes of two or more phases (liquid, gas or solid). It could be characterized by flow pattern according to physical distribution of phases in the pipe. During multiphase flow, the flow regime depends on magnitudes of forces that act on the fluid from other fluids or from pipe wall. Pressure gradient depends on flow pattern significantly, that is why it is necessary to identify it correctly before calculating pressure gradient. Several articles are devoted to the application of ML algorithms in the identification of flow pattern. In these papers, authors plotted experimental data points in suitable coordinates and tried to construct models in order to match these points. In paper [1] author created Artificial Neuron Network, in [2] authors applied Support Vector Machine algorithm and in [3] fuzzy inference system was used.

The other very important parameter of multiphase flow is liquid holdup. It is a fraction of pipe volume that occupied by liquid phase. This characteristic undoubtedly influences on pressure gradient of the flow that is why it should be correctly calculated. Liquid holdup is

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

also important in planning design of separation equipment. Several papers were devoted to the identification of this parameter by machine learning tools. For example, in [4] and [1] authors have applied Artificial Neuron Network.

These two parameters – flow pattern and liquid holdup – are calculated in different multiphase flow correlation for the pressure gradient definition. There are plenty of correlations that were developed using laboratory experiments. The most popular are Beggs and Brill [5], Mukherjee and Brill [6] correlations and others. Many studies have been done in order to find out the applicability of the correlations and authors of these articles considered that single correlation couldn't be used for any ranges of input parameters because each correlation works correctly under its own ranges of data. There are also several semi-empirical mechanistic models that are used for prediction of different multiphase flow characteristics. The most popular ones are Hasan and Kabir [7], Ansari [8] and others. Mechanistic models have advantages in specific flow pattern prediction but they aren't more accurate in pressure drop predictions compared to empirical correlations.

Researchers also tried to apply ML in order to predict pressure gradient or output pressure directly. In [9] article authors construct ANN for prediction the bottomhole pressure, in [10] authors also predict bottomhole pressure with the use of ANN but they suggested to divide well into segments and to define flow pattern in each segment.

In this paper, three machine learning models are constructed for prediction the following parameters of multiphase flow pipe segment: liquid holdup, flow regime and pressure gradient consistently. Pipe segment to be a part of the pipe which has homogeneous flow type and approximately constant pressure gradient. In the process of creating models, four machine learning algorithms are tried out such as Random Forest Regressor, Gradient Boosting Regressor, Support Vector Regressor and ANN in order to compare their predictive capability.

2. The method description

2.1. Method of calculation the pressure gradient in the pipe segment

In this part of the article, the process of calculation the pressure gradient in the segment will be described. The proposed method allows estimating pressure gradient in the segment that could be oriented to angles from $-90°$ to $90°$. Positive and negative angles describe uphill and downhill flow respectively and angle equals to zero is related to horizontal flow.

In order to decrease the number of input parameters for ML algorithms and to make the model similar for different types of liquids, the set of dimensionless variables is used according to paper [11]. These parameters are called velocity number of gas, velocity number of liquid, diameter number, viscosity number and defined by the following equations:

$$N_{vg} = v_{sg}\sqrt[4]{\frac{\rho_l}{g\sigma_{lg}}}, N_{vl} = v_{sl}\sqrt[4]{\frac{\rho_l}{g\sigma_{lg}}}, N_d = d\sqrt{\frac{\rho_l g}{\sigma_{lg}}}, N_\mu = \mu_l\sqrt[4]{\frac{g}{\rho_L\sigma_{lg}^3}} \quad (1)$$

where v_{sg} and v_{sl} are gas and liquid superficial velocities, ρ_l - liquid density and σ_{lg} is surface tension between liquid and gas phases, d - diameter of the tube and μ_l - liquid viscosity.

In order to predict pressure gradient in the segment, the regression model is constructed. Since pressure gradient depends significantly on the flow pattern and liquid holdup in the segment, these features are included into input parameters of the model. These two characteristics are calculated by other two ML models which will be described later. Apart flow pattern and liquid holdup, the following parameters are among input features of this regression model: inclination angle, dimensionless parameters (eq. 1), average pressure and temperature.

The second ML model that is a multi-class classifier for prediction flow regime. In this paper, the following four flow patterns are distinguished: bubble, slug, annular mist and stratified flows. The input features of this classifier are inclination angle, dimensionless parameters (eq. 1), liquid

holdup, average pressure and temperature. In order to increase the predictive power of the model, the dataset is divided into three physically homogeneous parts: data points related to horizontal flow, to up-flow and to down-flow and, consequently, besides the general model that uses all data, three ML models were constructed and trained on the appropriate data set.

In both previous models, the liquid holdup parameter is also included in input variables. As a result, the final regressive model is devoted to liquid holdup prediction. This model is trained on the following input parameters: inclination angle, dimensionless parameters (eq. 1), average pressure and temperature. Similar to the flow pattern prediction model, in this case three ML models for horizontal flow, upstream and downstream are constructed.

2.2. Sources of datasets

In order to compose dataset for training ML models, the data points are collected from the articles, books and PhD dissertations published in open source. From paper [12] 111 data point for horizontal flow are taken. Author of this paper carried out experiments using kerosene and water as liquid phase and air as the gas phase. Next 88 data points are from the article [13] in which author performed an experiment for horizontal flow using kerosene and air. Further, 1400-points dataset is taken from [14] which consists of uphill, downhill and horizontal flows in pipes, with inclination angle varying from $-90°$ to $90°$. The author uses kerosene and lube oil as liquid phase and air as the gas phase. From [15] 238 data points of horizontal multiphase flow of water and natural gas are used. The final 188 data points of water and air multiphase flow are taken from [16]. Among this data we also find flow in pipes oriented to angles $-10°$ and $10°$ except horizontal flow.

As a result, the total number of data points for construction ML model for liquid holdup prediction and flow pattern identification is approximately 2000. Among these points, about 1100 points are applied for training, validation and testing ML model for pressure gradient prediction. In the composed dataset the flow pattern is provided for about 1400 data point. In order to fill remaining 400 points the flow pattern map created by Mukherjee [6] is used.

2.3. Applied Machine Learning algorithms, tuning and evaluation scores

Four Machine Learning algorithms are considered in this paper: Random Forest, Gradient Boosting, Support Vector Machines and Artificial Neuron Networks. All these algorithms are tried out in each of aforesaid models for liquid holdup prediction, flow pattern identification and pressure gradient calculation in order to compare its predictive capability.

Firstly, let's introduce some designations. $\mathbf{X}$ is a matrix with arguments with size $m \times d$ (m is a number of samples and d is number of features). y is a matrix that contains interested real values. It has size $m \times r$ where $r = 1$ when single output in the problem and $r > 1$ when multiple output task. Value $\hat{y}$ is also introduced that is a matrix with predicted values and has the same size as matrix y.

The first ML algorithm that is applied – **Random Forest** [17]. It is an ensemble machine learning method which is used for solving regression and classification problems. This algorithm builds several independent decision trees and average them in order to obtain final result:

$$h_N(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^{N} h_i(\mathbf{X}) \tag{2}$$

where $h_i(\mathbf{X})-$ is decision tree with number i, N - number of decision trees.

In this article, Random Forest algorithm is used in the form that is implemented on Python language in the Scikit-learn library [18].

The second method that is used in the construction of ML models is **Gradient Boosting** [19]. This method also refers to ensemble methods and is used in both regression and classification

problems but has another structure. Algorithm constructs several decision trees and results as the weighted sum of them:

$$h_N(\mathbf{X}) = \sum_{i=1}^{N} \alpha_i h_i(\mathbf{X}) \tag{3}$$

Each decision tree $h_i(X)$ tries to fit anti-gradient of loss function (logistic, exponential loss fucntions):

$$-\frac{\partial L(f(x_j), y_j)}{\partial f(x_j)}\bigg|_{f(\mathbf{X})=h_i(\mathbf{X})}, j = 1, ..., m$$

where $f(x_j) = \hat{y}_j$ is predicted value for sample with number j.

Weights in the sum of decision trees (α_i) are found from the minimization of loss function:

$$\alpha_i = \text{argmin}_{\alpha>0} \sum_{j=1}^{m} L(f_j + \alpha h_i(\mathbf{x}_j), y_j)$$

Similar to Random Forest, Gradient Boosting algorithm is applied in the version of the Scikit-learn library [18].

Further **Support Vector Machine** algorithm [20] is considered. SVM is a machine learning algorithm that is used in classification and regression analysis. Very often matrix of arguments $\mathbf{X}$ is transformed into high-dimensional feature space by non-linear mapping before application of SVM algorithm F. Dot product of transformed vectors $\mathbf{x}$ and $\mathbf{x'}$ is a kernel of transformation $K(\mathbf{x}, \mathbf{x})$: $K(\mathbf{x}, \mathbf{x}) = F(\mathbf{x}) \cdot F(\mathbf{x'})$. In the construction of ML models using SVM in this paper Gaussian kernel with width σ is taken: $K(\mathbf{x'}, \mathbf{x}) = \exp\left(\frac{||\mathbf{x}-\mathbf{x'}||^2}{2\sigma^2}\right)$. This transformation is applied in order to make the data linearly separable (in the classification task) or to make the transformed data fit with the line function: $\mathbf{w} \cdot F(\mathbf{X}) + b$.

SVM algorithm solves the optimization problem of maximizing margin defined as $\frac{1}{||\mathbf{w}||}$ which is the same to minimize the norm $||\mathbf{w}||$. This optimization problem is solved in dual formulation with the use of Lagrangian.

SVM algorithm is also exploited in the form of the Scikit-learn [18].

The final ML technique used in this paper is **Artificial Neuron Networks** [21]. This is a mathematical representation of neurons in the brain. The mathematical or computer model of representation of the biological system is also called multi-layer perceptron. This ML algorithm is also applied in classification and regression problems. ANN consists of several layers and in each layer have a certain number of nodes. The first layer contains input parameters, so it has p nodes. The last one contains output parameters: in the case of regression problems or binary classification there is only one node in the case of multiclass classification the number of nodes which is equal to the number of classes. The algorithm consists of forward and backward propagation. In the forward propagation process, algorithm calculates values in each node in the following way: in order to obtain value in node k in layer $h + 1$ algorithm performs linear combination of values in nodes in layer h with definite weights and apply to this linear combination activation function $g(\cdot)$:

$$\mathbf{y}_{k,h+1} = g(\theta^T_{k,h\to h+1}\mathbf{y}_h) \tag{4}$$

In the backward process, the algorithm adjusts weights by using gradient descent optimization algorithm in order to decrease the value of loss function $L(f_i, y_i)$.

In classification problems sigmoid $\left(f(x) = \frac{1}{\exp(-x)+1}\right)$ or hyperbolic tangent ($f(x) = \tanh(x)$) activation functions are used. In the case of regression - ReLu ($f(x) = \max(0, x)$).

Artificial Neuron Network is presented in many libraries on Python language. In this paper, the functions from Scikit-learn library [18] are used.

For tuning models hyperparameters M cross-validation procedure is used. In this technique dataset is divided into M equally sized parts and $M-1$ partitions are exploited as training dataset and remaining partition as validation dataset This process is repeated M times and on each iteration different validation partition is used. The model that gives the best score is the model with optimal hyperparameters. In order to evaluate the score of the model with the best hyperparameters $N \times M$ cross-validation procedure is applied. In this method, dataset is divided into M equally sized parts N times and in each N-step the partitions are different. Thanks to this technique the calculated score of the model ($mean$) is more correct compared to M cross-validation and also allows to build confident intervals of such a score ($\pm\, 2 \cdot std$). In the case of M cross-validation $M = 5$ is used and in the case of $N \times M$ cross-validation $M = 5$ and $N = 20$ are exercised.

During construction of ML models the following set of evaluation metrics is applied. In the case of multi-classification problem, following accuracy score is exploited:

$$\text{Accuracy} = \frac{1}{m}\sum_{i=1}^{m} 1_{f(x_i)=y_i} = \sum_{y} \frac{TP_y + TN_y}{TP_y + FN_y + TP_y + FN_y} \tag{5}$$

where TP_y - true positives of class y, FP_y - false positives of class y, FN_y - false negatives of class y and TN_y - true negatives of class y.

In the regression problems the coefficient of determination (R^2 score) is used:

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2} \tag{6}$$

where $\bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i$

3. Results

In this part of the paper, the obtained results are represented, namely, scores of the algorithms with the best predictive capability, cross-plots.

Let's start with the first model that predicts liquid holdup. In all considered cases – horizontal, uphill, downhill flows and in the case when all dataset is used in model construction the Gradient Boosting algorithm has the best predictive capability. When all dataset is used for training, validation and testing model, Gradient Boosting has $R^2 = 0.925 \pm 0.033$. In model for horizontal flow $R^2 = 0.974 \pm 0.015$, for uphill flow $R^2 = 0.965 \pm 0.013$ and for downhill flow $R^2 = 0.83 \pm 0.102$. From these results, the following conclusion could be made: models for horizontal and uphill flows demonstrate well predictive capability with high coefficient of determination, while the model for downhill flow reveals the relatively low result.

Further, let's move on to results of the second model that predicts flow pattern in the segment. Gradient Boosting algorithm has the best predictive capability when all dataset is used for model creation with accuracy score 0.85 ± 0.037. In the model for horizontal flow also Gradient boosting has the best accuracy score that is equal to 0.924 ± 0.043, for uphill flow neuron network has the best predictive capability with accuracy 0.875 ± 0.052. Finally, in the case of uphill flow ANN performs the best result with an accuracy score 0.769 ± 0.063. From these results, one could make a conclusion that models for horizontal and up flows have good accuracy score, while the model for downhill flow reveals a poor result.

Now, let's consider the third model which predicts pressure gradient in the segment. In this case division on horizontal, uphill and downhill doesn't apply because of small data points that belong to each class. So, the only one model for pressure gradient prediction is built which uses

all dataset in training, validation and testing stages. In this model, flow pattern feature is used among input parameters. It is included via one hot encoding method in which four new columns are created (their number is equal to the number of flow regimes in this problem) instead of one column which contains names of flow pattern. These new features consist of 0 and 1. When sample belongs to slug flow pattern it will have 1 value in the column that is responsible for indication of an appurtenance to the slug flow class. In this model, SVM algorithm gives the best score – $R^2 = 0.985 \pm 0.011$ whereas Gradient Boosting demonstrates slightly worse result – $R^2 = 0.971 \pm 0.014$.

Since model for the pressure gradient estimation is the most important in the proposed method for pressure drop calculation, it is necessary to construct a cross-plot. In this graph on X-axis real values are plotted while predicted values are on Y-axis. At Figure 1 such cross-plot is demonstrated.

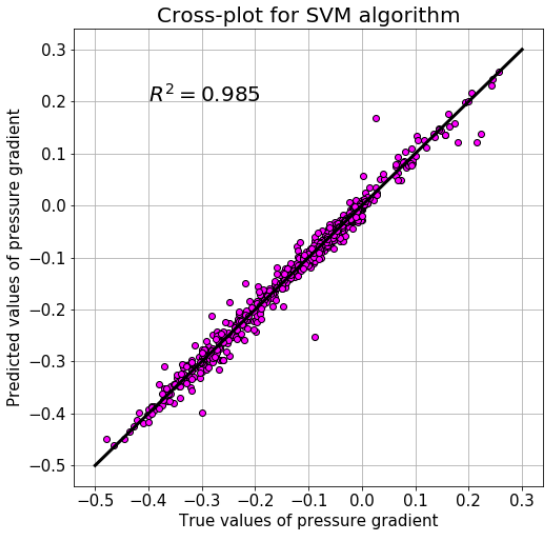

Figure 1. Cross-plot for SVM algorithm in the pressure gradient model.

In order to summarize results of created models, the Table 1 with scores is written. In this table results of all applied machine learning algorithms in the case where all data set is used are represented.

Models	Model 1	Model 2	Model 3
Algorithm \ Metric	R^2	Accuracy	R^2
Random Forest	0.917 ± 0.04	0.848 ± 0.036	0.955 ± 0.02
Gradient Boosting	0.925 ± 0.033	0.85 ± 0.037	0.971 ± 0.014
SVM	0.858 ± 0.043	0.846 ± 0.039	0.985 ± 0.011
ANN	0.884 ± 0.04	0.833 ± 0.037	0.98 ± 0.011

Table 1. ML models results of all applied machine learning algorithms in the case of usage of all dataset.

4. Conclusions and future directions

In the present paper, the new method of pressure gradient calculation in the pipe segment is developed. It consists of three surrogate models that are based on the Machine Learning algorithms such as Random Forest, Support Vector Machine, Gradient Boosting and Artificial Neuron Network. The first model predicts liquid holdup parameter. The best R^2-score in this model achieved by using Gradient Boosting algorithm and it is equal to 0.925. The second one is focused on flow pattern identification. The best accuracy score in this model also belongs to Gradient Boosting algorithm and equal to 85.5%. The last one calculated pressure gradient. In this case SVM method performs best of all and shows R^2-score equal to 0.985. The distinguishing feature of the proposed method is high accuracy. As a result, this method calculates bottomhole pressures, pressure distributions along the pipelines and other very important characteristics in petroleum engineering domain more correctly allowing to solve different optimization (production maximization e.g.) and planning tasks (oil transportation system enchancements e.g.) more effectively.

There are also some possible ways to improve the model, its predictive capability and expand boundaries of applicability. Firstly, it is necessary to add data points to the collected dataset: it could be experimental data points from other researchers, synthetic data calculated by using multiphase flow simulators such as OLGA, ANSYS Fluent and others. Secondly, temperature effects should be introduced. In the present paper, temperature is linearly interpolated between boundaries but it is more correct to calculate entropy, heat transfer along the tube.

References

[1] Osman E S 2004 *Oil Production & Facilities* **19** 33–40
[2] Li X, Miskimins J L, Sutton R P, Hoffman B T *et al.* 2014 *SPE Annual Technical Conference and Exhibition* (Society of Petroleum Engineers)
[3] Popa F, Dursun S, Houchens B *et al.* 2015 *SPE Annual Technical Conference and Exhibition* (Society of Petroleum Engineers)
[4] Shippen M E, Scott S L *et al.* 2002 *SPE Annual Technical Conference and Exhibition* (Society of Petroleum Engineers)
[5] Beggs D H, Brill J P *et al.* 1973 *Journal of Petroleum technology* **25** 607–617
[6] Mukherjee H, Brill J P *et al.* 1983 *Journal of Petroleum Technology* **35** 1–003
[7] Hasan A, Kabir C *et al.* 1986 *SPE California Regional Meeting* (Society of Petroleum Engineers)
[8] Ansari A, Sylvester N, Shoham O, Brill J *et al.* 1990 *SPE Annual Technical Conference and Exhibition* (Society of Petroleum Engineers)
[9] Osman E S A, Ayoub M A, Aggour M A *et al.* 2005 *SPE Middle East Oil and Gas Show and Conference* (Society of Petroleum Engineers)
[10] Li X, Miskimins J, Hoffman B T *et al.* 2014 *SPE Annual Technical Conference and Exhibition* (Society of Petroleum Engineers)
[11] Duns Jr H, Ros N *et al.* 1963 *6th World Petroleum Congress* (World Petroleum Congress)
[12] Minami K, Brill J *et al.* 1987 *SPE Production Engineering* **2** 36–44
[13] Abdul-Majeed G 1996 *Journal of Petroleum Science and Engineering* **15** 271–280
[14] Mukherjee H 1979 *An Experimental Study of Inclined Two-Phase Flow* Ph.D. thesis U. of Tulsa
[15] Eaton B 1966 *The prediction of flow patterns, liquid holdup and pressure losses occurring during continuous two-phase flow in horizontal pipelines* Ph.D. thesis U. of Texas at Austin
[16] Beggs H 1973 *An Experimental Study of Two-Phase Flow in Inclined Pipes* Ph.D. thesis U. of Tulsa
[17] Breiman L 2001 *Machine learning* **45** 5–32
[18] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M and Duchesnay E 2011 *Journal of Machine Learning Research* **12** 2825–2830
[19] Friedman J 1999 *GreedyFuncApproxSS. pdf*
[20] Cortes C and Vapnik V 1995 *Machine learning* **20** 273–297
[21] Rosenblatt F 1958 *Psychological review* **65** 386