Hindawi Publishing Corporation
Journal of Petroleum Engineering
Volume 2013, Article ID 746315, 8 pages
http://dx.doi.org/10.1155/2013/746315

Research Article

Knowledge Discovery for Classification of Three-Phase Vertical Flow Patterns of Heavy Oil from Pressure Drop and Flow Rate Data

Adriane B. S. Serapião[1] and Antonio C. Bannwart[2]

[1] *DEMAC, IGCE, UNESP, CP 178, 13506-900 Rio Claro, SP, Brazil*
[2] *DEP, FEM, UNICAMP, CP 6122, 13081-970 Campinas, SP, Brazil*

Correspondence should be addressed to Adriane B. S. Serapião; adriane@rc.unesp.br

Received 26 August 2012; Revised 20 November 2012; Accepted 21 November 2012

Academic Editor: Guillaume Galliero

This paper focuses on the use of artificial intelligence (AI) techniques to identify flow patterns acquired and recorded from experimental data of vertical upward three-phase pipe flow of heavy oil, air, and water at several different combinations, in which water is injected to work as the continuous phase (water-assisted flow). We investigate the use of data mining algorithms with rule and tree methods for classifying real data generated by a laboratory scale apparatus. The data presented in this paper represent different heavy oil flow conditions in a real production pipe.

1. Introduction

The design of oil production pipelines involves evaluation of flow lines subject to multiphase flow of oil, water, and gas, where oscillations in pressure, temperature, and phase concentration typically occur. Furthermore, the phases usually flow on different geometrical distributions inside the pipe, named flow patterns. The identification of flow patterns is essential for the economic evaluation of the project, such as pressure drop and flow rate along the pipeline. These aspects are critical on offshore production conditions, where extensive distances and high costs are involved. Flow pattern identification is an important step to design separation equipments, slug catchers, gas lift operations, wellhead gathering systems, and production management and control. With the discovery of heavy oil reservoirs the lack of tools and methodologies for flow pattern identification deserves attention because the existing multiphase flow correlations are made for low API oils, where the oil-water mixture may be treated as a single liquid phase with average properties. However, for water-continuous flow of heavy oil below bubble-point, three distinct phases are present, that is, oil, water, and gas, thus making the traditional approach of flow pattern classification and pressure drop prediction in three-phase flow may have poor accuracy.

Basically, there are two types of models for flow pattern prediction: empirical and mechanistic. Empirical models are related to experimental data, where flow pattern maps are experimentally determined and analyzed with respect to mathematical relations representing the boundaries between the flow pattern regions. These relations depend on the amount of experimental data used and on the coordinate system in which the data are presented. Mechanistic models are based on balance equations [1]. Notwithstanding, these models are formulated to describe single or two-phase flows, and they cannot be highly extended for oil-gas-water mixtures, when the phases are considered heterogeneous, as in the "water-assisted flow" case, because of the complexity of phase interaction in the multi-phase flow. The same inadequate extension from two- to three-phase representation occurs for correlation models.

Pressure drop control for heavy oil production and transportation can be achieved by injecting water in the pipeline so as to create a water-continuous flow known as "water-assisted flow." In view of the complexity of three-phase flow, the development of an objective flow pattern identification scheme is of fundamental importance to obtain useful information about the flow nature and control purposes as well.

In a previous paper [2], the three-phase water-assisted flow of heavy crude oil with gas (air) in a vertical pipe was investigated at a low pressure laboratory set up. For each trio of flow rates, flow patterns were identified by means of direct visualization (with the help of movie recording, if necessary) and total pressure drop was measured for comparison with existing correlations.

The main difficulty in visual observation is that the picture is often confusing and difficult to interpret, especially when dealing with high velocity flows. So, in an automatic flow pattern identification tool is broadly interesting and useful for online applications in the oil industry, particularly when the flow visualization is not available.

In the present work, we investigate the automatic classification of the three-phase flow patterns described in [2] from the correspondent flow rates and pressure drop data. This topic is suitable for field applications when direct visualization or tomography (impedance, ultrasonic, optical, or other) of the flow is not possible or impracticable. So, the automatic classification can be used, for instance, as a first step of a decision making system for analyses of real heavy oil production pipelines. It might be regarded as an alternative to the traditional physical modeling of flow pattern transitions in rather complex flows such as three-phase, for which a unified predicting model is, to the best of our knowledge, unavailable.

As an alternative to overcome the prediction difficulties with empirical and mechanistic modeling, neural networks and other data mining techniques can be used at pattern recognition and trend prediction for processes that are nonlinear, poorly understood, and too complex for accurate mathematical modeling, as three-phase flow pattern classification/identification.

Classification is one of the important data mining tasks. Data mining is a progress to extract implicit, nontrivial, previously unknown and potentially useful information (such as knowledge rules, constraints, regularities) from data in database. Data mining is an interdisciplinary field. The obtained results must conform to three major requisites: accuracy, comprehensibility, and interest for the user [3].

The classification corresponding to the vertical upward flow patterns were elaborated by visual inspection by a petroleum engineer and some data mining methods were trained to learn this classification: PSO/ACO2 algorithm, neural networks, support vector machines (SVM), decision tree learning algorithms (J48, REPTree and LMT), rule induction algorithm (JRIP) [4], and rules built by an expert.

Many classification algorithms have been constructed and applied to discover knowledge from data in different applications, yet many suffer from poor performance in prediction accuracy in many practical domains. While it seems unlikely to have an algorithm to perform best in all the domains, it may well be possible to produce classifiers that perform better on a wide variety of real-world domains [5].

In this paper we have also two additional purposes. The first one is to study the contribution of recent natural computing algorithms to solve real-world applications in the petroleum engineering area. For this goal, the PSO/ACO2 algorithm receives special attention.

Rule and tree based methods are particularly recommended in applications where knowledge comprehensibility is very important, such as in engineering processes—where discovered knowledge should be carefully validated and interpreted by experts before they are actually used to diagnose a problem, suggest an intervention procedure or recognize a pattern, rather than blindly trust the result provided by an algorithm. In this context, the second purpose is to find the algorithm that leads to a best comprehensibility of the real problem, because it is very important topic whenever discovered knowledge will be used for supporting a decision made by a human user. When properly designed and trained, a classification tool for applications in multiphase flow systems can potentially improve on-line monitoring and diagnostics, playing the role of a reliable, objective, and quantitative indicator of flow regime.

2. Experiments in Three-Phase Water-Assisted Flow of Heavy Oil with Gas

In the following we summarize the test conditions and the main results obtained in the experiments described by Bannwart et al. [2].

The test section consisted of a 2.84 cm i.d., 2.5 m long vertical glass tubing for the three-phase flow. The oil flow rate was measured with a Coriolis mass flow meter, whereas the water and air flow rates were read in rotameters. Pressure data in the test section were measured with differential and absolute pressure transducers connected to a data acquisition system.

The oil utilized was a blend of crude dead oil with a viscosity of $\mu_o = 5,040$ mPa·s and a density of $\rho_o = 971\,\text{kg/m}^3$ at 25°C. The oil phase was observed to be a w/o emulsion. The water used was tap water contained in the separator tank and the air was provided by an existing group of compressors.

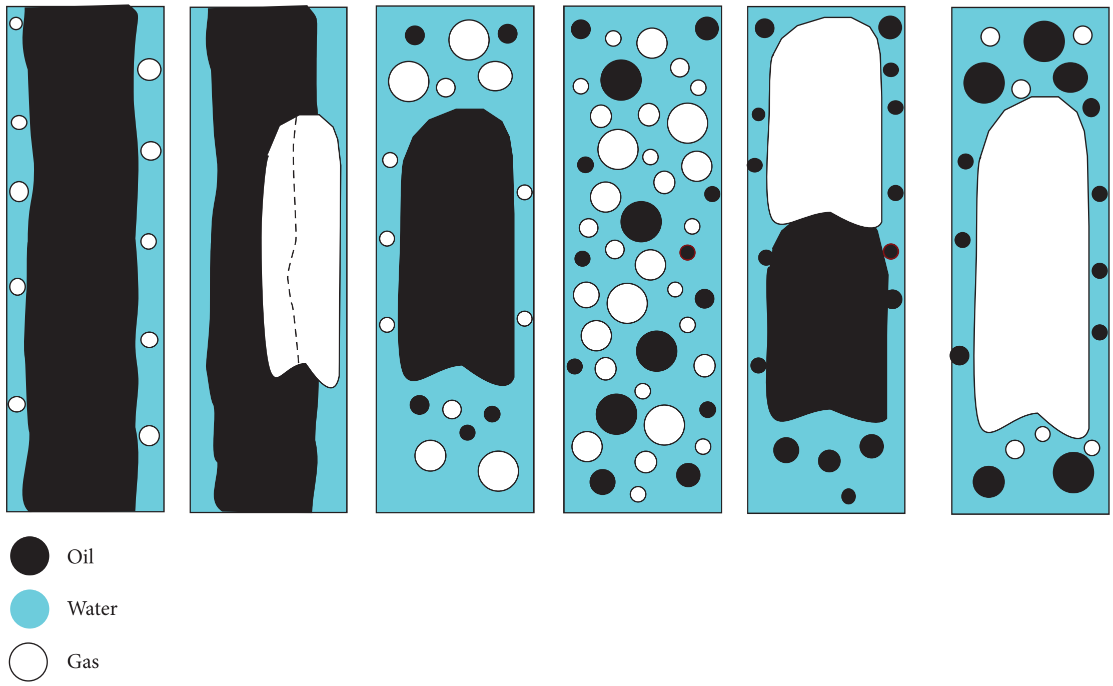
The experiments consisted of simultaneously flowing water, crude oil and air at several flow rate combinations. For each set a video footage of the established flow pattern was taken with a high-speed camera (1000 frames/s) and pressure data were collected. The experimental superficial velocities (J) varied within the following ranges:

(i) oil: $0.02 < J_o < 1.2$ m/s,

(ii) air: $0.04 < J_g < 9$ m/s,

(iii) water: $0.04 < J_w < 0.5$ m/s.

The experiments took place at ambient temperature and near atmospheric pressure. In all runs, water was always injected first (in order to make sure that it would be the continuous phase), followed by oil and air. The glass pipe was never observed to be fouled (hydrophilic behavior). Steady state operation was achieved after the flow rates and average pressure drop readings were observed to be stable.



(a) Ig-Ao (b) Ig-Ao (c) Bg-Io (d) Bg-Bo (e) Ig-Io (f) Ig-Bo

Figure 1: Three-phase flow patterns for vertical upward water-assisted flow of heavy oil in the presence of a free gas phase.

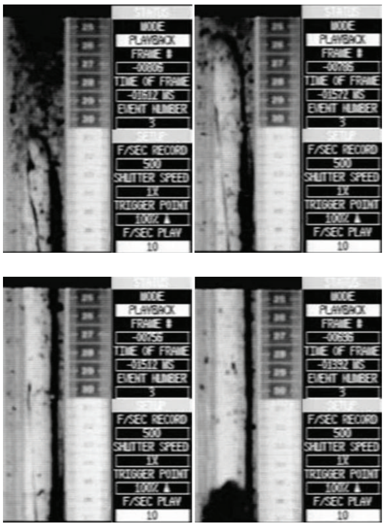


Figure 2: Illustration of the passage of large gas bubble in the Ig-Io flow pattern.

The recorded movies were played in slow motion in order to make possible the identification of the flow patterns. Figure 1 illustrates the six identified flow patterns, all of them water-continuous, which were named according to the gas and oil distributions within the water-continuous phase. Figure 2 provides details of one of the flow patterns observed in the test section.

Following a brief description of each flow pattern is presented.

(a) Bg-Ao: Bubbly Gas-Annular Oil. This pattern is similar to heavy oil-water core flow, except that here gas bubbles are seen in the water phase. The oil-water interface is typically sinuous. This pattern occurs for high oil and low gas superficial velocities.

(b) Ig-Ao: Intermittent Gas-Annular Oil. The gas phase forms large bubbles which partly surround a still continuous oil core. This pattern occurs for high oil and moderate gas superficial velocities.

(c) Bg-Io: Bubbly Gas-Intermittent Oil. The gas forms small bubbles and the oil forms large bubbles. This pattern occurs for moderate oil and low gas superficial velocities.

(d) Bg-Bo: Bubbly Gas-Bubbly Oil. This pattern was observed for low oil and gas superficial velocities, but only when the water superficial velocity was higher than about 0.3 m/s, which was enough to disperse the oil into bubbles.

(e) Ig-Io: Intermittent Gas-Intermittent Oil. The gas and the oil both form large bubbles which are very close to each other. Detailed observation shows that the oil bubble is sucked towards the low pressure wake behind the gas bubble. In this pattern, small oil and gas bubbles are spread in the water. An illustration of this flow pattern is shown in Figure 2.

This pattern occurs for high gas and oil superficial velocities, and also for moderate gas and oil superficial velocities.

(f) Ig-Bo: Intermittent Gas-Bubbly Oil. At high gas superficial velocities the gas forms large, high speed bubbles and the oil is dispersed into small bubbles. This pattern is typically pulsating, indicating a transition to annular gas-liquid flow.

3. Data Mining Methods

In this section, we present a brief description of the classification methods used in this work: PSO/ACO2 algorithm, J48, REPTree, LMT, JRIP, neural networks, and SVM.

3.1. PSO/ACO2 Algorithm. The algorithm PSO/ACO1, originally proposed in [6, 7], an algorithm proposed for the discovery of classification rules in data mining problems. It was designed to hierarchical classification, where the classes to be predicted are arranged in a tree-like hierarchy. This algorithm uses concepts of the Particle Swarm Optimization (PSO) algorithm, which is mainly inspired by social behavior patterns of organisms that live and interact within large groups [8], combined with Ant colony optimization (ACO), which takes inspiration from the foraging behavior of some real ant species [9].

The modified version of the PSO/ACO algorithm used in this paper, hereafter denoted PSO/ACO2 [10], uses a sequential covering approach to discover one-classification-rule-at-a-time. The modifications in this algorithm include changes in the pheromone update procedure, in the quality measure evaluation, and in the rule initialization method, as well as the splitting of the rule discovery process into two separated phases.

A rule consists of an antecedent (a set of attribute-values) and a consequent (class). The consequent of the rule is the class that is predicted by that rule. The antecedent consists of a set of terms. A term is defined by a triple ⟨*attribute*, *operator*, *value*⟩, where value is a value belonging to the domain of attribute. The operator used is "=" in the case of categorical/nominal attributes, or "≤" and ">" in the case of continuous attributes. In this context, each case is assigned to one predefined class according to the values of its attributes for the case. The discovered knowledge is expressed in the form of *IF-THEN* rules (1), as follows:

$$\begin{aligned} & IF \ \langle attrib = value\rangle \ And \cdots And \\ & \langle attrib \geq value\rangle \ THEN \ \langle class\rangle . \end{aligned} \tag{1}$$

3.2. J48. J48 is an implementation of the well-known Quinlan algorithm (C4.5) [11], which is an improvement of the basic ID3 algorithm. This classifier builds a decision tree for the given dataset, whose nodes represent discrimination rules acting on selective features by recursive partitioning of data, using depth-first strategy.

The algorithm uses the fact that each attribute of the data can be used to make a decision by splitting the data into smaller subsets. To make the decision, the algorithm considers all the possible tests that can split the data set and culls a test that gives the highest information gain. For each discrete attribute, one test with outcomes for each distinct value of the attribute is considered. For each continuous attribute, binary tests involving every distinct values of the attribute are considered. In order to gather the entropy gain of all these binary tests efficiently, the training data set belonging to the node in consideration is sorted for the values of the continuous attribute and the entropy gains of the binary cut based on each distinct values are calculated in one scan of the sorted data. Then a new feature is chosen and the splitting process is repeated for each attribute in a recursive manner until further splitting is not gainful. In the resulting tree structure, each inner node in the tree corresponds to an attribute, each branch represents a possible value or range of values of that attribute and each leaf represents the predicted value of target attribute.

3.3. REPTree. Reduced Error Pruning Tree (REPTree) is a simple procedure for learning and pruning decision trees. It can build a decision or regression tree using information gain as the splitting criterion and prunes trees using reduced-error pruning. It only sorts values for numeric attributes once.

The procedure covers the internal nodes of the tree from the bottom to the top and verifies each internal node. The goal is to state whether replacing it with the most frequent class does not reduce the trees accuracy. If accuracy is not reduced then the node is pruned. Pruning is used to find the best sub-tree of the initially grown tree with the minimum error for the test set. This process continues until any further pruning would decrease the accuracy. The procedure stops with the smallest accurate (lowest classification error) sub-tree with respect to a given pruning set.

3.4. LMT. Logistic Model Tree (LMT) combines a standard tree structure with logistic regression functions at the leaves using posterior class probabilities to produce a single decision tree [12]. LMT consists of a tree structure that is made up of a set of inner nodes and a set of leaves or terminal nodes in an instance space. A test on one of the attributes is associated with every inner node. For numeric attributes, the leaf node has two child nodes which is branched right and left depending on the threshold. If the value of the numeric attribute is smaller than the threshold it is sorted to left branch and value of attribute greater than the threshold it is sorted to right branch otherwise. The threshold is usually fixed by Logit Boost method. For nominal attributes with *n* values, the branch has *n* child nodes, so that the instances are sorted down one of the *n* branches, according to their value of the attribute.

An algorithm for building logistic model trees has the following three steps: growing the tree, building the logistic models, and pruning. Briefly, the tree induction process identifies subdivisions by recursively splitting the instance space in a divide-and-conquer fashion until further subdivisions are not gainful.

3.5. JRIP. JRIP implements *repeated incremental pruning to produce error reduction* (RIPPER) in Java, a prepositional rule learner, as proposed by Cohen [13]. The RIPPER rule learning algorithm is an extended version of learning algorithm IREP (*incremental reduced error pruning*). Rules are created for every class in the training set and are then pruned. In this algorithm, the discovered knowledge is represented in the form of *IF-THEN* prediction rules, which have the advantage of being a high-level and symbolic knowledge representation contributing towards the comprehensibility of the discovered knowledge [14].

The method is based on the construction of a rule set in which all positive examples are covered. Initially, the current set of training examples are partitioned into two subsets, a growing set and a pruning set. The rule is constructed from examples in the growing set. The rule set initiates with an empty rule set and rules are added incrementally to the rule set until no negative examples are covered. After that, JRIP replaces or revises individual rules by using reduced error pruning in order to increase the accuracy of rules. It replaces or revises individual rules by using reduced error pruning. To prune a rule the algorithm takes in account only a final sequence of conditions from the rule and sorts the deletion that maximizes the function.

3.6. Neural Networks. Artificial neural networks (ANNs) are adaptive systems with the power of a universal computer. They are massive parallel processors comprised of single interconnected artificial neurons and are able to realize an arbitrary mapping (association) of one vector space (inputs) to the other vector space (outputs) finding patterns in data, using a computational model for information processing. They are being used with increasing frequency for high dimensional problems, either to approximate a posteriori probabilities for classification or regression [4]. Neural networks are analytical techniques, which are capable of acquiring the knowledge of the complex process of an environment from the observed data, storing the knowledge of underlying process, and making it available to apply to new observations of same data types for pattern recognition purposes. Two commons neural networks' models used for these tasks are Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF).

In the most common form, neural networks are composed by hidden layers of multiple artificial neurons connected to the inputs and outputs neurons with different weights, which correspond to synapses in the biological neuron. Neurons are processing units that that apply nonlinear activation functions to approximate complex functions in the data. The weights are iteratively adjusted during the training procedure using any given approximation technique, such as gradient descendent method, by comparing desired inputs with observed inputs, until a stop criterion is reached. Training or learning procedure is the process of finding the best set of weights for the neural network, mapping the relationships between predictor and target variables.

One key criticism of ANNs is that are a "black box." The nature of the relationship between independent (inputs) and dependent variables (outputs) is usually not revealed and the importance of each variable is not made explicit. Any functional form or map of the relationships is not supplied by ANNs, because of the complexity of the functions used in the neural network approximations.

3.7. Support Vector Machine. Support Vector Machine (SVM) is primarily a classifier method that performs grouping tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. In order to construct an optimal hyperplane, SVM employees an iterative training algorithm, that is used to minimize an error function. The foundations of SVM have been developed by Vapnik [15] and are gaining popularity due to many attractive features, and promising empirical performance.

SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. The kernel function allows the algorithm to fit the maximum-margin hyperplane in the transformed feature space. There are four basic kernels: linear, polynomial, radial basic function (RBF) and sigmoid. SVMs are based on the *structural risk minimization* (SRM) principle from statistical learning theory.

This classifier method is also considered a "black-box" and does not offer typically any interpretation about the relationship between independent and dependent variables.

4. Results and Discussion

The classifiers tested in this work were modeled using the four inputs, representing the independent variables, which in this case are the superficial velocities of oil, water and gas (respectively, J_o, J_w, J_g) and pressure gradient ($\Delta P/L$). The output is one of the six target class previously classified by the expert (Bg-Ao, Bg-Bo, Bg-Io, Ig-Ao, Ig-Bo, and Ig-Io) representing each flow pattern according to Figure 1.

Experimental records of three-phase flow of heavy oil with gas and water in a vertical pipe data set consisting of 119 samples were used for the training and evaluation of the implemented classifier. For the classifiers' training, the whole data set was randomly separated into two subsets: 75% as training subsets (89 samples) and 25% as testing subsets (30 samples) after training. The training set contains 9, 5, 6, 21, 38 and 10 samples, respectively, for Bg-Ao, Bg-Bo, Bg-Io, Ig-Ao, Ig-Bo, and Ig-Io classes. The correspondent distribution for the test dataset is 2, 1, 3, 5, 13, and 6 samples, respectively.

In the classification task, after the discovered rules from a set of training data, those rules must be applied to a set of test data (unseen during training), and hopefully predict the correct class in the test set. Thus, the accuracy rate for unseen samples is for us the most important index to evaluate the classifier's efficiency, because it proofs its generalization ability.

The performance of the PSO/ACO2 classifier employed to identify the flow patterns was assessed through comparisons between original and estimated outputs taken from the data subsets used both in training and in testing procedure samples for four different types of machine learning algorithms, one knowledge rule model developed by a petroleum engineer expert, two neural networks model and three SVM kernels. Experiments for classification with swarm intelligence techniques were conducted using PSO/ACO2 software developed by Nicholas Holden and provided in the SourceForge (available at http://sourceforge.net/projects/psoaco2/)

project site. Upon experimentation, the suitable numbers of particles and iterations were found to be both 100 and the number of uncovered examples per class (*MaxUncovExampPerClass*) to 2. For PSO algorithm the constriction factor is $\chi = 0.72984$, and cognitive and social learning coefficients are $c_1 = c_2 = 2.05$.

For the Machine Learning experiments with rule and tree based algorithms (JRip, J48, RepTree and LMT), we used well-known tree and rule based classification algorithms available in the Waikato Environment for Knowledge Analysis (WEKA) software package [14].

The neural networks (NN) models explored for this experiment are: Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF). On MLP neural network, four neurons were used in input layer, representing the independent variables like PSO/ACO2, ten neurons were employed in hidden layer and six neurons were used in output layer, representing each flow pattern. The amount of neurons in the hidden layer was found by experimentation, after assessing which configuration would yield the least global training error. The activation function used at each level was sigmoid tangent for hidden and output layer neurons. The supervised training algorithm used was the Backpropagation with weights and biases updating according to Levenberg-Marquardt optimization algorithm. For the RBF network, 35 hidden layer neurons were employed using Gaussian function with coefficient of spread (σ) equal to 0.1, with a linear activation function had been used on the six output neurons. The software Matlab MathWorks Inc. v.7.4 was used to implement the neural networks models employed in this work.

We have also used multi-class SVM Type 1 for the classification task, considering the "one-against-one" method for decomposing multi-class problem in binary subproblems. Linear, polynomial, and RBF functions were tested as the mapping function (kernel) for the classification system. Matlab was used to implement the SVM model mentioned in this work. The SVM classifier was modeled using the four inputs, representing the independent variables and the output is one of the six target class, as defined before. The complete description of this procedure can be consulted in [15].

In this work, the data were also interpreted according to the set of preestablished rules by an expert. The human knowledge was represented by a set of *IF-THEN* rules. In order to build these rules, human knowledge regarding characteristics of each flow pattern and the relation of each pattern with the four parameters measured with the experiment was determined by inspection of the dataset. For sake of comparing with the other data mining methods, the data were separated in the same training and test sets for evaluation with the expert rules.

All the experiments were obtained on a Centrino Duo PC (CPU 1.83 GHZ, RAM 2 GB) for the same datasets.

Table 1 indicates the classification accuracy of PSO/ACO2 algorithm for training and test sets, in the classification task of vertical flow patterns, using the settings parameters mentioned earlier and the discovered rules, compared with others data mining methods. The PSO/ACO2 algorithm was able to achieve a success rate of 68.56% in the training set, the worst performance among the other methods, and 70% in the test set, one of the best performances among the other classifiers. The hybrid particle swarm optimization/ant colony optimization algorithm produced in the training phase 13 classification rules used to identify the vertical flow patterns as indicated in Algorithm 1. All patterns were covered with this rule set.

Table 1: Comparing accuracy of flow classifiers.

Classifier	Training set	Test set	Full set
PSO/ACO2	68.56%	70.0%	68.92%
JRIP	78.65%	56.67%	73.11%
J48	86.52%	56.67%	78.99%
RepTree	77.53%	70.0%	75.63%
LMT	79.78%	63.33%	75.63%
MLP-BP NN	89.89%	60.0%	82.35%
RBF NN	91.01%	66.67%	84.87%
Linear SVM	85.39%	70.0%	81.51%
Polynomial SVM	100%	73.33%	93.28%
RBF SVM	100%	70.0%	92.44%
Expert rules	73.03%	80.0%	74.79%

In contrast with JRIP, which is another classifier based on discovering rules, PSO/ACO2 presented a best performance for unseen data (test set). JRIP produced only six simple rules for classifying the flow patterns, as detailed in Algorithm 1. The methods based on decision trees, J48, RepTree, and LMT demonstrated better results to learn the patterns during the training, notwithstanding the scores for testing do not outperformed PSO/ACO2, except RepTree, which had the same efficiency (70%).

Figure 3 displays the knowledge tree provided by RepTree, containing 17 decision leaves. The variable J_w is referred only to distingue the Ig-Ao and Bg-Ao classes. In the tree we can observe that no rules for Bg-Bo and Ig-Io were created. Thus, the recognition of these patterns is completely wrong using this solution.

Expert rules have found 80% of success in identifying the test samples, with a lower rate for the training set (73.03%), which was the worst among all the other techniques, apart from PSO/ACO2 algorithm. The expert has used nine rules to represent his knowledge about de flow patterns. According to those rules, the variable J_w is used only to aid detecting the Bg-Bo class. A comprehensive observation of the rules appoint that a pressure drop (ΔP) superior to 9300 Pa/m causes almost always a bubble regime in the gas phase, where the oil regime is determined by the superficial velocities. Otherhand, a pressure drop lower than 8700 Pa/m originates certainly an intermittent regime in the gas phase, so that the oil regime is determined by the others variables. In the intermediate range of drop pressure, between 8700 and 9300 Pa/m, depending on the values of the four measured variables, the possible flow patterns are only Bg-Bo, Ig-Io and Ig-Ao.

Despite their ineptitude to generate descriptive information about the classification process, which could useful to better understand the physical phenomenon under study,

PSO/ACO2 rules:

R1: If ($J_o \leq 0.639$) and ($J_g \geq 2.858$) and ($\Delta P \geq 4033.14$) then *class* = Ig-Bo
R2: If ($J_o \geq 0.148$) and ($J_w \leq 0.292$) and ($J_g \leq 5.351$) and ($6461.45 \leq \Delta P \leq 9831.71$) then *class* = Ig-Ao
R3: If ($J_o \geq 0.673$) and ($J_w \geq 0.108$) and ($J_g \leq 4.108$) and ($\Delta P \leq 10003$) then *class* = Ig-Io
R4: If ($J_o \geq 0.251$) and ($J_g \leq 0.293$) and ($\Delta P \geq 9714.49$) then *class* = Bg-Ao
R5: If ($J_o \leq 0.069$) and ($J_w \leq 0.071$) and ($J_g \leq 4.645$) and ($\Delta P \leq 8329.8$) then *class* = Ig-Ao
R6: If ($J_o \leq 0.075$) and ($J_w \leq 0.32$) and ($\Delta P \geq 9364.72$) then *class* = Bg-Io
R7: If ($J_o \leq 0.241$) and ($J_w \geq 0.323$) and ($8719.27 \leq \Delta P \leq 9964.14$) then *class* = Bg-Bo
R8: If ($0.12 \leq J_w \leq 0.675$) and ($6329.42 \leq \Delta P \leq 9332.34$) then *class* = Ig-Ao
R9: If ($J_o \leq 0.144$) and ($J_w \geq 0.094$) and ($\Delta P \leq 8801.66$) then *class* = Ig-Bo
R10: If ($0.045 \leq J_o \leq 0.1489$) and ($J_w \leq 0.084$) and ($\Delta P \geq 9224.91$) then *class* = Bg-Ao
R11: If ($J_o \leq 0.04$) and ($0.1276 \leq J_w \leq 0.322$) and ($J_g \leq 0.626$) and ($\Delta P \leq 8619.74$) then *class* = Ig-Io
R12: If ($J_o \geq 0.055$) and ($J_w \leq 0.253$) and ($\Delta P \geq 6282.8$) then *class* = Ig-Ao
R13: Otherwise, *class* = Ig-Bo

JRIP rules:

R1: If ($\Delta P \geq 8786.5$) and ($J_o \leq 0.069$) and ($J_w \geq 0.327$) then *class* = Bg-Bo
R2: If ($J_g \leq 0.06$) and ($J_o \leq 0.069$) then *class* = Bg-Io
R3: If ($J_g \leq 0.051$) and ($\Delta P \geq 9758$) then *class* = Bg-Ao
R4: If ($J_o \geq 0.999$) and ($J_w \geq 0.316$) then *class* = Ig-Io
R5: If ($J_g \leq 3.592$) then *class* = Ig-Ao
R6: Otherwise, *class* = Ig-Bo

Expert rules:

R1: If ($J_o > 0.2$) and ($3.4 < J_g < 8.6$) and ($\Delta P < 7900$) then *class* = Ig-Bo
R2: If ($J_o < 0.1$) and ($0.3 < J_g < 0.6$) and ($6500 < \Delta P < 8500$) then *class* = Ig-Bo
R3: If ($J_o < 0.1$) and ($J_g > 3.5$) and ($\Delta P < 6500$) then *class* = Ig-Bo
R4: If ($0.33 < J_g < 0.38$) and ($6300 < \Delta P < 9800$) then *class* = Ig-Io
R5: If ($J_o > 0.5$) and ($3.1 < J_g < 4.1$) and ($5700 < \Delta P < 9000$) then *class* = Ig-Io
R6: If ($J_g < 0.055$) and ($\Delta P > 9300$) then *class* = Bg-Ao
R7: If ($J_o < 0.25$) and ($J_g < 0.07$) and ($9300 < \Delta P < 10000$) then *class* = Bg-Io
R8: If ($J_o < 0.3$) and ($J_w > 0.3$) and ($J_g < 0.4$) and ($8700 < \Delta P < 9860$) then *class* = Bg-Bo
R9: Otherwise, *class* = Ig-Ao

Algorithm 1: Generated rules from PSO/ACO2 and JRIP algorithms.

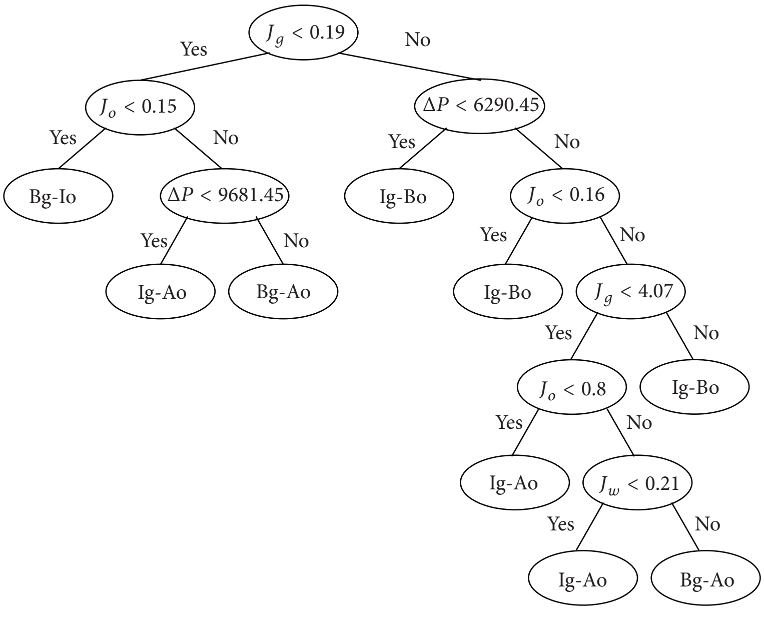


Figure 3: Knowledge tree generated by RepTree method.

neural networks and SVMs have presented superior prediction accuracy for both training and test sets than the metaheuristic methods and expert rules, as observed in Table 1. Polynomial SVM presented the best recognition score among all methods, classifying correctly all samples of the training set and obtaining 73.33% of recognition score for the test set.

With regard to the moderate accuracy rate for the vertical upward flow pattern classification of the investigated methods, it is supposed that this survey is due to the intermittent phase, which makes complex a correct prediction as reported in [16]. In all runs with every classifier, the patterns composed with a part of the intermittent regime had an expressive misrecognition, principally the Ig-Io class.

5. Conclusion

Flow pattern prediction in industrial systems that rely on complex multi-phase flows is essential for their safety, control, diagnostics, and operation. In oil extraction and oil-gas-water mixture transporting processes, the identification of flow regime is one of the essential tasks for the purpose of maintaining optimal operating and improving the performance of equipment. Usually, different flow regimes produce distinct performances of the system. However, identifying/classifying multiphase mixture continuously and precisely is still a significant and unsolved problem, due to the highly nonlinear nature of the forces which rule the flow regime transitions.

In this paper, we use the PSO/ACO2 algorithm for data mining, a hybrid method for mining classification rule. We have compared the performance of PSO/ACO2 with others data mining methods. Experimental results show that PSO/ACO2 has a predictive accuracy for untrained data (test set) greater than the others evaluated heuristic methods. Nevertheless, the *black-box* methods (neural networks and SVM) have shown a higher precision for predicting patterns than PSO/ACO2 and the others heuristic algorithms. Because the application of swarm intelligent algorithm in data mining, especially in classification rule mining, is still in infant periods, we believe that improvements in this methodology could increase the accuracy for the pattern detection, so that this kind of approach could be suitable for real world applications in Petroleum Engineering area.

Rule base methods are particularly recommended in applications where rule comprehensibility is very important, such as in engineering processes—where discovered rules should be carefully interpreted by experts before they are actually used to recognize a pattern.

Although the generation of classification rules by an expert is a feasible process, producing good results, it is an exhaustive and tedious task, mainly when a great number of parameters and samples must be analyzed. The automated classification of the data is vital when a huge volume of information is involved. The approach and methods used in this work contribute to an in-depth understanding of flow patterns dynamics, especially in the cases of multiphase flows, aiming extraction of simple rules for identifying flow pattern features. We intend to encourage researchers of the Petroleum Engineering area to use data mining to analyze real data in different domains as well.

A future research direction consists of exploring others techniques based on rule mining. Further work includes also dealing with a deeper study of the important features of three-phase flow of heavy oil in vertical pipe in order to obtain better classifier systems.

References

[1] A. Wegmann, *Multiphase flows in small scale pipes [Doctoral Dissertation]*, Federal Institute of Technology Zurich, 2005, ETH Nr. 16189.

[2] A. C. Bannwart, F. F. Vieira, C. H. M. Carvalho, and A. P. Oliveira, "Water-assisted flow of heavy oil and gas in a vertical pipe," in *Proceedings of the SPE International Thermal Operations and Heavy Oil Symposium (ITOHOS '05)*, Alberta, Canada, November 2005, Paper PS2005-SPE-97875-PP.

[3] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, "Data mining with an ant colony optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 321–332, 2002.

[4] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 2nd edition, 1999.

[5] T. Sousa, A. Silva, and A. Neves, "Particle swarm based data mining algorithms for classification tasks," *Parallel Computing*, vol. 30, no. 5-6, pp. 767–783, 2004.

[6] N. Holden and A. A. Freitas, "A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data," in *Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS '05)*, pp. 100–107, Pasadena, Calif, USA, June 2005.

[7] N. Holden and A. A. Freitas, "Hierarchical classification of G-protein-coupled receptors with a PSO/ACO algorithm," in *Proceedings of the 2006 IEEE Swarm Intelligence Symposium (SIS '06)*, pp. 77–84, Indianapolis, Ind, USA, 2006.

[8] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann, San Francisco, Calif, USA, 2001.

[9] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, Mass, USA, 2004.

[10] N. P. Holden and A. A. Freitas, "A hybrid PSO/ACO algorithm for classification," in *Proceedings of the 9th Annual Genetic and Evolutionary Computation Conference (GECCO '07)*, pp. 2745–2750, London, UK, July 2007.

[11] R. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, Calif, USA, 1993.

[12] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Machine Learning*, vol. 59, no. 1-2, pp. 161–205, 2005.

[13] W. Cohen, "Fast effective rule induction," in *Proceedings of the 12th International Conference on Machine Learning*, pp. 115–123, Lake Tahoe, Calif, USA, 1995.

[14] I. H. Witten, M. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tool and Technique with Java Implementation*, Morgan Kaufmann, San Francisco, Calif, USA, 3rd edition, 2011.

[15] V. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.

[16] F. Pacheco, A. C. Bannwart, J. R. P. Mendes, and A. B. S. Serapião, "Support vector ma-chines for identification of three-phase flow patterns of heavy oil in vertical pipes," *Brazilian Journal of Petroleum and Gas*, vol. 1, no. 2, pp. 95–103, 2007.

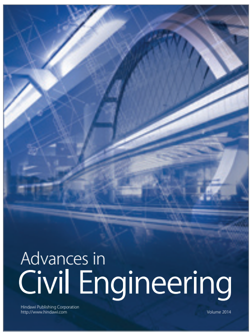
Journal of
Engineering

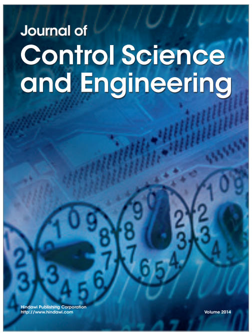
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

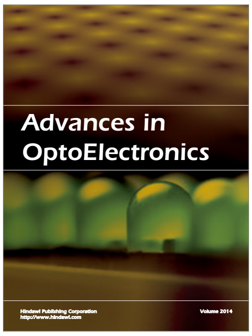
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

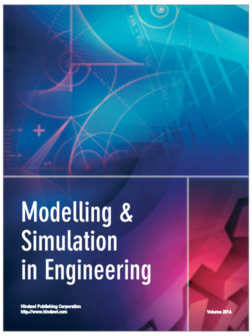
Journal of
Robotics

Journal of
Electrical and Computer Engineering

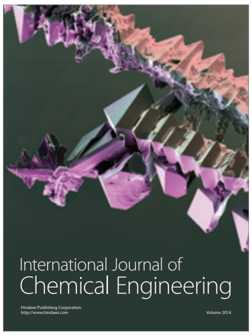

Advances in
OptoElectronics

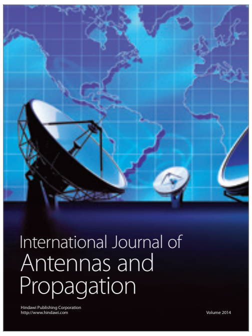
VLSI Design

International Journal of
Navigation and Observation

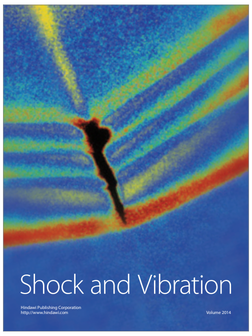

Modelling &
Simulation
in Engineering

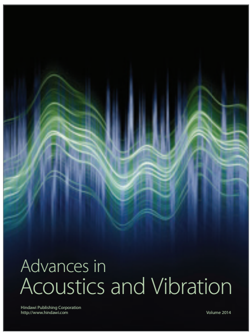
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration