SUPPORT VECTOR MACHINES FOR IDENTIFICATION OF THREE-PHASE FLOW PATTERNS OF HEAVY OIL IN VERTICAL PIPES

1 F. Pacheco, 1 A. C. Bannwart*, 1 J. R. P. Mendes, 2 A. B. S. Serapião

1 State University of Campinas - UNICAMP - Campinas, SP – Brazil
2 São Paulo State University - UNESP - Rio Claro, SP – Brazil

* To whom all correspondence should be addressed.
Address: DEP/FEM/UNICAMP, P.O. Box 6122, Campinas, SP, Brazil, CEP 13083-970
Telephone / fax numbers: +55 19 3289-4999 / +55 19 3521-3202
E-mail: bannwart@dep.fem.unicamp.br

Abstract. The purpose of this study is to investigate the relationship between phase flow rates, pressure drop and flow pattern identification in a three-phase water-assisted upward flow of heavy oil in the presence of a gaseous phase in vertical pipes. An automatic flow pattern classification tool is proposed using an artificial intelligence technique known as the 'support vector machine' (SVM). Real records of previously obtained laboratory data set of three-phase flow of heavy oil with gas and water in a vertical pipe were used to train and to evaluate the SVM classifier. Several tests with different parameters and training data were performed. Results indicate the most relevant model to be used in the flow pattern identification, revealing the enhanced prediction capability of the SVM technique. The SVM performances were compared with others classification methods.

Keywords: production; heavy oil; multiphase flow; flow pattern identification; artificial intelligence; support vector machines

1. INTRODUCTION

Petroleum characterized by high density (10-20°API) and viscosity (100-100,000 cP) is usually known as ‘heavy oil’ (Tissot and Welte, 1984). The importance of its recovery and productions is justified by the magnitude of their reserves - 4.6 trillions of barrels in place (Briggs et al., 1988). Production of heavy oil reservoirs has been generally successful in onshore and some offshore shallow water fields, but technologies for deep-water recovery and production of heavy oil are still under development (Wehunt et al., 2003).

The pipe flow of an oil-gas-water mixture such as that involved in heavy oil production is a rather complex thermo-fluid dynamical problem. The local pressure and temperature in the pipeline govern phase equilibria as well as the thermo-physical properties of each phase. Temperature and pressure drop control in production pipelines become crucial when viscous oils are involved. Pressure drop control can be achieved by injecting water in the pipeline so as to create a water-continuous flow pattern known as ‘water-assisted flow’. In view of the complexity of three-phase flows, the development of an objective flow pattern identification scheme is fundamentally important to obtain useful information about the flow nature and to establish the best strategy for the pressure drop control.

In a previous paper (Bannwart et al., 2005), the three-phase water-assisted flow of heavy crude oil with gas (air) in a vertical pipe was investigated in a low-pressure-ambient-temperature 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 the total pressure drop was measured for comparison with existing correlations.

In the present work, we investigate the possibility of detecting the three-phase flow patterns previously described (Bannwart et al.,...
from the correspondent flow rates and pressure drop data. This issue may be relevant when direct visualization or tomography of the flow is not possible or is unfeasible, such as in a real heavy oil production pipeline. For that purpose we use a 'support vector machine' tool (SVM), which is well-known in the field of Artificial Intelligence (Cristianini and Shawe-Taylor, 2000). The available data set is then analyzed with the aim of determining which properties are decisive for flow pattern classification.

Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. The formulation of SVM embodies the Structural Risk Minimization (SRM) principle. SRM minimizes an upper bound on the expected risk, so that this approach equips SVM with a greater ability to generalize, which is the goal in statistical learning. SVMs were primary developed to solve the classification problem, but recently they have been extended to the domain of regression problems.

In the classification problem by the SVM approach, the goal is to separate the classes by hyperplanes (support vectors) which are induced from available examples. The purpose is to produce a classifier that generalizes well for both trained and unseen examples. A separating hyperplane is optimal if it separates without error and maximizes the margin (i.e., maximizes the distance between it and the nearest data point of each class).

The results of the implemented SVM classifiers (SVC) are compared with two models of neural networks for the same classification task. Neural networks (NNs) have been applied in a variety of scientific areas in the past decades, solving problems such as pattern recognition, image treatment, time-series prediction, etc. The most important characteristic regarding NNs is their capability of generalizing a result based on knowledge previously acquired. When a neural network is trained, it is presented to a set of data constituted of inputs and expected outputs. The NN memorizes a set of internal weights that will bring the training input set to the expected output. One of the great advantages of this process is that a neural network does not need to be familiar with the mathematical model in order to present a generalized expected output. The mapping between the input and output is provided through its own internal mechanisms of weighting and summing.

This article is organized as follows. Section 2 gives a brief overview of the physical experimental set up, data acquisition and the upward flow patterns in vertical pipe. Section 3 reviews the foundations and models of the SVM technique used for implementing a classification machine. The next section focuses on the implementations and results of the SVM classifier models, comparing their performances with traditional NN approaches for the classification task. We conclude, in the last section, with remarks of this work and directions of future researches.

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

In this part, we summarize the test conditions and the main results obtained in the experiments described previously (Bannwart et al., 2005).

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 $\rho_o = 971$ kg/m$^3$ at 25°C. The oil phase was observed to be a water-in-oil (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 \([U_{i,s}, \text{where } i \text{ refers to either oil (o), gas (g) or water (w)}]\) varied within the following ranges:
- oil: \(0.02 < U_{o,s} < 1.2 \text{ m/s}\)
- gas (air): \(0.04 < U_{g,s} < 9 \text{ m/s}\)
- water: \(0.04 < U_{w,s} < 0.5 \text{ m/s}\)

The experiments took place at ambient temperature and near atmospheric pressure. In all runs, water was always injected first (in order to ensure that it would be the continuous phase), followed by oil and air. The glass pipe was never observed to be fouled (hydrophilic behavior).

Figure 1 illustrates the six identified flow patterns, all of them water-continuous, which were named according to the gas and oil distributions in water. The patterns are described below.

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 \(\text{m/s}\), 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. 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.
1.2. Support Vector Machines

SVMs have been successfully applied to solve a large number of classification tasks (Cristianini and Shawe-Taylor, 2000; Vapnik, 1998). They are based on the principles of structural risk minimization (SRM) and hence have good generalization ability. In the simplest and linear form, SVM is the optimal hyperplane that separates a set of positive samples from a set of negative samples, with a margin defined by the distance between the hyperplanes supporting the nearest positive and negative samples and at the same time reducing the empirical risk.

The power of SVMs lies in their ability to transform data to a higher dimension and construct a linear binary classifier in that higher dimension. The construction of the linear decision boundary is done implicitly and hence the sparseness of the data in the higher dimension is not an issue.

For the two-class pattern classification problem, SVM finds a hypersurface \( y(x) \) decision, where the vector \( x \) belongs to the space of samples (Burges, 1998). The analysis is performed with a set of \( N \) data points \( \{x_i, y_i\} \), where \( x_i \in \mathbb{R}^n \) is the \( i \)-th input data, and \( y_i \in \{-1, +1\} \) is the label of the data. The purpose of the SVM approach is to find a form classifier (Vapnik, 1998), as established by equation 1:

\[
y(x) = \text{sign} \left[ \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \right]
\]  

(1)

where \( \alpha_i \) are positive real constants and \( b \) is a real constant; in general, \( K(x_i, x) = \langle \phi(x_i), \phi(x) \rangle \), where \( \langle \cdot, \cdot \rangle \) is an inner product and \( \phi(x) \) is the nonlinear map from the original space to the high dimensional space.

In the high dimensional space, we assume the data can be separated by a linear hyperplane. This will cause:

\[
\begin{align*}
\begin{cases}
w^T \phi(x_i) + b \geq 1, & \text{if } y_i = +1 \\
w^T \phi(x_i) + b \leq -1, & \text{if } y_i = -1
\end{cases}
\]

(2)

which is equivalent to

\[
y_i \left[ w^T \phi(x_i) + b \right] \geq 1, \quad i = 1, \ldots, N
\]

(3)

If the separating hyperplane does not exist, a so-called slack variable \( \xi_i \) is introduced such that:

\[
\begin{align*}
\begin{cases}
\begin{ cases}
y_i \left[ w^T \phi(x_i) + b \right] \geq 1 - \xi_i, & \text{if } y_i = +1, \ldots, N \\
\xi_i \geq 0, & \text{if } i = 1, \ldots, N
\end{cases}
\end{cases}
\]

(4)

According to the structural risk minimization principle, the risk bound is minimized by the following minimization problem:

\[
\min_{w, b} J_1(w, \xi) = \frac{1}{2} w^T w + c \sum_{i=1}^{N} \xi_i
\]

subject to (4). One constructs the Lagrangian function as:

\[
L(w, b, \xi, \alpha, \beta) = J_1(w, \xi) - \sum_{i=1}^{N} \alpha_i \left[ w^T \phi(x_i) + b \right] - \sum_{i=1}^{N} \beta_i \xi_i
\]

(5)

where \( \alpha_i \geq 0, \beta_i \geq 0 \) \((i = 1, \ldots, N)\) are the Lagrangian multipliers of (Bannwart et al., 2005). The optimal point will rest in the saddle point of the Lagrangian function, i.e.:

\[
\max_{\alpha, \beta} \min_{w, b, \xi} L_1(w, b, \xi, \alpha, \beta)
\]

(6)

We then obtain:

\[
\begin{align*}
\frac{\partial L_1}{\partial w} &= 0 \quad \rightarrow \quad w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \\
\frac{\partial L_1}{\partial b} &= 0 \quad \rightarrow \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \\
\frac{\partial L_1}{\partial \xi_i} &= 0 \quad \rightarrow \quad 0 \leq \alpha_i \leq c, \ i = 1, \ldots, N
\end{align*}
\]

(7)

Replacing (8) with (6), we will get the following quadratic programming problem:

\[
\max_{\alpha} Q_1(\alpha) = -\frac{1}{2} \sum_{i=1}^{N} \alpha_i \sum_{j=1}^{N} \alpha_j y_i y_j K(x_i, x_j) + \sum_{i=1}^{N} \alpha_i
\]

(9)
where: $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ is called the kernel function. It is possible to use different types of functions as sigmoid, gaussian radial basis (RBF), polynomial and linear functions in order to implement the kernel function, as shown by equation 10:

$$
\phi = \begin{cases} 
  x_i \cdot x_j & \text{linear} \\
  (\gamma \cdot x_i \cdot x_j + \text{coefficient})^{\text{degree}} & \text{polynomial} \\
  \exp(-\gamma |x_i - x_j|^2) & \text{RBF} \\
  \tanh(\gamma \cdot x_i \cdot x_j + \text{coefficient}) & \text{sigmoid}
\end{cases}
$$

(10)

The RBF is by far the most popular choice of kernel types used in Support Vector Machines. This is mainly because of their localized and finite responses across the entire range of the real $x$-axis.

Solving this quadratic programming problem, subject to the constraints in (8), will place the hyperplane in the high dimensional space and hence the classifier in the original space as in (1).

To construct an optimal hyperplane, SVM uses an iterative training algorithm, which is used to minimize an error function. According to the form of the error function, SVM models can be classified into four distinct groups:

- Classification SVM Type 1 (also known as C-SVM classification)
- Classification SVM Type 2 (also known as ν-SVM classification)

For the C-SVM classification, training involves the minimization of the error function:

$$
\frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i
$$

(11)

subject to the constraints:

$$
y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i,
$$

(12)

with $\xi_i \geq 0$ $(i = 1, \ldots, N)$

where $C$ is the capacity constant, $w$ is the vector of coefficients, $b$ is a constant and $\xi_i$ are parameters for handling non-separable data (inputs). The index $i$ labels the $n$ training cases. Note that $y \in \pm 1$ is the class labels and $x_i$ denotes independent variables. The kernel $\phi$ is used to transform data from the input (independent) to the feature space. It should be noted that the larger the $C$, the more the error is penalized. The parameter $C$ controls what proportion of samples of a particular class can be wrongly placed with regard to the separation hypersurface and its margin. In other words, the soft margin parameter controls the overtraining/generalization trade-off when learning a particular classifier from data and should be chosen with care to avoid over fitting.

In contrast to C-SVM, the ν-SVM classification model minimizes the error function:

$$
\frac{1}{2} w^T w - \nu \rho + \frac{1}{N} \sum_{i=1}^{N} \xi_i
$$

(13)

subject to the constraints:

$$
y_i (w^T \phi(x_i) + b) \geq \rho - \xi_i,
$$

(14)

with $\xi_i \geq 0$ $(i = 1, \ldots, N)$ and $\rho \geq 0$

At first, SVM was originally designed to solve binary classification problems. Later they could also be extended to solve multiclass problems by the combination of binary SVM classifiers, where the data set is decomposed to several binary problems. Two binary classification approaches, the one-against-one and the one-against-all strategies, are typically used to solve the multi-class classification problem.

The ‘one-against-one’ approach trains a binary SVM for any two classes of data and obtains a decision function. Thus, for an $N$-class problem, there are $N(N-1)/2$ decision functions. In the prediction stage, a voting strategy is used where the testing point is designated to be in a class with the maximum number of votes.

The ‘one-against-all’ approach decomposes the $n$-class classification problem into $N$ binary classification sub-problems. Each classifier separates one class $k$ from the remaining $N-1$ classes and constructs a binary classifier for this class versus all others, in order to find the
decision function. The resulting combined decision function chooses the class for a sample that corresponds to the maximum value of \( k \) binary decision functions (i.e. the furthest ‘positive’ hyperplane).

In the one-against-all method, the similarity between the classes is not considered, so there is no guarantee that good discrimination exists between one class and the remaining classes; in the ‘one-against-one’ method, a multi-class classification problem is exhaustively decomposed into a set of \( N^2/2 \) classifiers, and the number of classifiers and computations are prohibitive. Using C-SVM classifier, various multi-class approaches give similar accuracy. However, the ‘one-against-one’ method is more efficient for training (Hsu and Lin, 2002).

2. IMPLEMENTATION AND RESULTS

We used multi-class SVM Type 1 as training algorithm for classification, considering the ‘one-against-one’ method for decomposing multi-class problem in binary sub-problems. The linear, polynomial and gaussian RBF functions were tested as the mapping function (kernel) for the classification system. The software Matlab®, v.7.4 by MathWorks Inc. was used to create the SVM model employed in this work.

The SVM classifier is modeled using the four inputs representing the independent variables (oil flow rate, water flow rate, gas flow rate and pressure gradient) and 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 2a.

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 SVM classifier. When training SVM network 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 for the Bg-Ao, Bg-Bo, Bg-Io, Ig-Ao, Ig-Bo and Ig-Io classes, respectively. The correspondent distribution for the test dataset is 2, 1, 3, 5, 13 and 6 samples, respectively.

The performance of the SVM classifier employed to identify the vertical 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 the three different types of kernels. In the linear SVM the capacity constant \( (C) \) was set as approaching infinity \( (C \rightarrow \infty) \). The implementation of the polynomial kernel has considered a 4 degree polynomial with \( C = 2 \), \( \gamma = 2 \) and \( \text{coefficient} = 1 \). The RBF kernel has adopted \( C = 300 \) and \( \gamma = 1 \). These
parameters were found by experimentation, after assessing which configuration would yield the least classification error. All the experiments were obtained on a Centrino Duo PC (CPU 1.83 GHZ, RAM 2GB).

Table 1 shows the results comparing the statistical performance of SVM classifier using the three different kernels for both training and test datasets using a vertical pipe. The results of our experiments can be summarized as follows. Concerning the classification accuracy, the polynomial and RBF kernels provided better results than the linear kernel in the training data set. Accuracy values are reported as percentage of success, comparing the real class with the predicted class. The polynomial and RBF kernel correctly classified all samples of the training set, with 100% of recognition score. For the linear kernel, 13 samples were misclassified, reaching a reliability of 85.39%.

The polynomial function provided somewhat better results (73.33% of success) than the other kernels during the testing phase. With regard to the other two variants of training (linear and RBF kernels), it is possible to notice that the performances of both models are virtually similar (70%). For both linear and RBF kernels, nine samples presented a false classification, whereas the polynomial kernel indicated eight incorrectly classified samples.

Table 1 also demonstrates that the misclassifications in the training and test data sets for all kernels involve mainly the false identification of the intermittent pattern either for the oil (Io) or gas phase (Ig). The pattern Ig-Io had the worst performance among all classes.

Because of this observation, we decided to split the classification system in two modules: one for gas pattern and another for the oil/water emulsion pattern identification, which is hereby named oil pattern. The system works as indicated in Figure 2b. Both modules receive the same input variables but each one emits an output relative to its commission, considering the same classes of Figure 1. In this way, gas SVM provides two possible classes (bubble and intermittent) and the liquid SVM caters to three classes (annular, bubble and intermittent). The final output is the aggregation of the partial solutions (gas+oil), forming the known six vertical flow patterns (Bg-Ao, Bg-Bo, Bg-Io, Ig-Ao, Ig-Bo and Ig-Io). The goal of this approach is to evaluate the role of each phase in the pattern classification task, in order to conceive the flow phenomena. We have considered the same parameters of the first approach for the related kernels.

The results of the second approach are very similar to the first flow identification system, with an error distribution as shown in the Table 2. For the gas module, the accuracy was almost 100% for all kernels in the training and test sets. It means that the bubble and intermittent regimes in the gas phase are easily differentiable. However, the oil phase obtained around 70% of correctness for each kernel in the test set. Therefore, the recognition rate for the combined pattern (gas+oil) is around 60% for the linear kernel, and 70% for the polynomial and RBF kernels. The RBF mapping provided the same performance of the polynomial one for both training and test sets. The main difference between both is that the
RBF kernel detected the intermittent class with accuracy less than the other classes in the test set, but by contrast the polynomial presented a balanced misclassification among the three classes.

In view of all these facts, we have postulated that, generally, the intermittent regime is the critical element responsible for impairing an otherwise correct flow pattern prediction, substantially due to the oil phase. Nevertheless this remark is intuitive but not unique; as a matter of fact, its confirmation by means of computational tools is relevant.

Other hand, it is necessary to confirm the efficiency of the SVM classification technique. The results obtained using SVM for vertical flow pattern detection were compared with two well-known models of neural networks (NN): perceptron multi-layer (MLP) with back-propagation (BP) training algorithm (Haykin, 1999) and radial basis function (RBF) network (Poggio, 1994). Table 3 shows the accuracy rate for each flow classification method, considering the training and test datasets.

RBF neural network obtained a best result than MLP-BP neural network for upward flow pattern identification in vertical pipe, as reported in Table 3. However, comparing neural networks and SVMs classifiers, we can notice that, for the most part, the SVM carried out more thoroughly blanket flow pattern detection, especially on account of the test subset. The accuracy rate for unseen samples (test set) is for us the most important index to evaluate the classifier's efficiency, because it proofs its generalization ability. Polynomial SVM outperformed all implemented classifier systems.

In general, the results of the SVM classifiers can be considered satisfactory for this initial investigation of the flow pattern identification problem.

Computational time required to solve the problem classification is governed primarily by the time required for the whole data set training. In this study, the spent time for reaching the solution is very short, around 0.5 second in the worst case (linear kernel). The difference for training time among the kernels is so small that can be considered as insignificant for our experiments.

| Table 3. Comparing the accuracy of neural and SVM flow pattern classifiers |
|-----------------------------|-----------------------------|-----------------------------|
| Classifier                  | Training set                | Test set                    |
| MLP-BP NN                   | 89.89%                      | 60.0%                       |
| RBF NN                      | 91.01%                      | 66.67%                      |
| Linear SVM                  | 85.39%                      | 70.0%                       |
| Polynomial SVM              | 100%                        | 73.33%                      |
| RBF SVM                     | 100%                        | 70.0%                       |

RBF kernel detected the intermittent class with accuracy less than the other classes in the test set, but by contrast the polynomial presented a balanced misclassification among the three classes.

In view of all these facts, we have postulated that, generally, the intermittent regime is the critical element responsible for impairing an otherwise correct flow pattern prediction, substantially due to the oil phase. Nevertheless this remark is intuitive but not unique; as a matter of fact, its confirmation by means of computational tools is relevant.

Other hand, it is necessary to confirm the efficiency of the SVM classification technique. The results obtained using SVM for vertical flow pattern detection were compared with two well-known models of neural networks (NN): perceptron multi-layer (MLP) with back-propagation (BP) training algorithm (Haykin, 1999) and radial basis function (RBF) network (Poggio, 1994). Table 3 shows the accuracy rate for each flow classification method, considering the training and test datasets.

RBF neural network obtained a best result than MLP-BP neural network for upward flow pattern identification in vertical pipe, as reported in Table 3. However, comparing neural networks and SVMs classifiers, we can notice that, for the most part, the SVM carried out more thoroughly blanket flow pattern detection, especially on account of the test subset. The accuracy rate for unseen samples (test set) is for us the most important index to evaluate the classifier's efficiency, because it proofs its generalization ability. Polynomial SVM outperformed all implemented classifier systems.

In general, the results of the SVM classifiers can be considered satisfactory for this initial investigation of the flow pattern identification problem.

Computational time required to solve the problem classification is governed primarily by the time required for the whole data set training. In this study, the spent time for reaching the solution is very short, around 0.5 second in the worst case (linear kernel). The difference for training time among the kernels is so small that can be considered as insignificant for our experiments.
3. CONCLUSION

The experimental results show that the proposed SVM for the flow pattern identification task achieved an acceptable predictive accuracy. It was possible to understand the main difficulty for the flow pattern classification, which is associated with the identification of the intermittent pattern either for gas or oil phase. The polynomial function appeared to be the most suitable kernel for the SVM-classifier in the proposed problem, since such model got the best performance on classifying of non-training data.

Regarding the timing requirements, it may be noted that the SVM-classifier takes a small amount of time during training. Furthermore, the time taken during testing is much shorter, motivating its usage in real-time applications.

Directions for future work should include the application of this tool to classify larger flow datasets and the exploration of other pattern recognition methods for this same task.

ACKNOWLEDGEMENTS

The authors would like to thank the Brazilian Research Council (CNPq) and Petrobras – Petróleo Brasileiro S.A. for their support to this work.

REFERENCES


