

A Statistical Feature-Based Approach for Operations Recognition in Drilling Time Series

Bilal Esmael¹, Arghad Arnaout², Rudolf K. Fruhwirth² and Gerhard Thonhauser¹

¹University of Leoben
8700 Leoben, Austria
Bilal@stud.unileoben.ac.at
Gerhard.Thonhauser@unileoben.ac.at

²TDE GmbH
8700 Leoben, Austria
Arghad.Arnaout@tde.at
Rudolf.Fruhwirth@tde.at

Abstract- Recognizing patterns in time series has become a necessary machine learning task in many fields including medicine, finance, business and oil and gas industry. In this paper we propose a feature-based approach to recognize patterns in drilling time series data. Our approach consists of four phases which are: data preparation, feature extraction, feature selection and classifier training. In the first phase, the sensor-generated data required for building the recognition models are collected and prepared. In the second phase, the prepared time series data are transformed into a compact representation. The compact representation of the data consists of a set of statistical features extracted by sliding a window across the time series. In the third phase, numerous feature selection algorithms are applied to select a subset of most informative features from the statistical features set. Finally, the selected features are exploited to train a classifier that is used for final pattern recognition. Numerous feature weighting and selection algorithms were tested to find which statistical measures clearly distinguish between several different patterns. In addition, many classification techniques were employed to find the best one in terms of accuracy and speed. Experimental evaluation with real data shows that our approach has the ability to extract and select the best features and build accurate classifiers. Four different real-world drilling scenarios were used in the experiments. The performance of the classifiers was evaluated by using the cross-validation method.

Keywords: Operation recognition, Time series classification, Statistical Features, Features Selection.

I. Introduction

Time series data are omnipresent and broadly available in industrial applications. In the oil and gas industry, it is very common to monitor the basic drilling actions such as moving the drill string, rotating the drill string and circulating the drilling mud. Many mechanical parameters, such as hook load and block position, are continuously measured during drilling oil wells. These parameters are measured by a group of sensors located around the drilling rig and wired to a measurement system called a mud-logging system. Fig. 1 shows sensor-generated time series data of eight hours of drilling. This time series is

multivariate which means that many variables - eight in our case - will be measured at each time point.

After collecting the data at the rig site, data transferring systems and data storing systems can be employed to transfer and store these data anywhere in the world. Although the sensor measurement and transferring systems are being developed rapidly, the techniques of data interpretation and analysis have not developed at the same speed. There is a lack of systems able to make efficient use of all the data available to improve the drilling process.

Improving the drilling process relies on performance analysis that is primarily based on daily activity breakdowns [1]. Drilling operations recognition systems break down the total drilling time into a list of well-defined operations e.g. drilling, rotating and make connection. These systems provide the engineers with detailed information about what is happening on the rig site. In the last decade, numerous operation recognition approaches have been proposed. Some of these approaches take as input the sensors data themselves, and recognize the drilling operations. Adriane et al. [2] present a drilling operations classification system using Support Vector Machine (SVM). The input of this system is five sensors values with a specific timestamp, and the output is one of six predefined operations.

Drilling time series data have a very high dimensionality. The high dimensionality of the data makes the access time very slow and the total computation time more expensive [3]. That means, applying machine learning techniques directly on raw time series data is not practical. What is needed is a higher-level representation of the raw data that allows efficient computation, and extracts higher order features. Esmael et al. [4] propose a new representation of drilling time series data which combines trend-based and value-based approximations. The proposed compact representation consists of symbolic strings that represent the trends and the values of each variable in the time series.

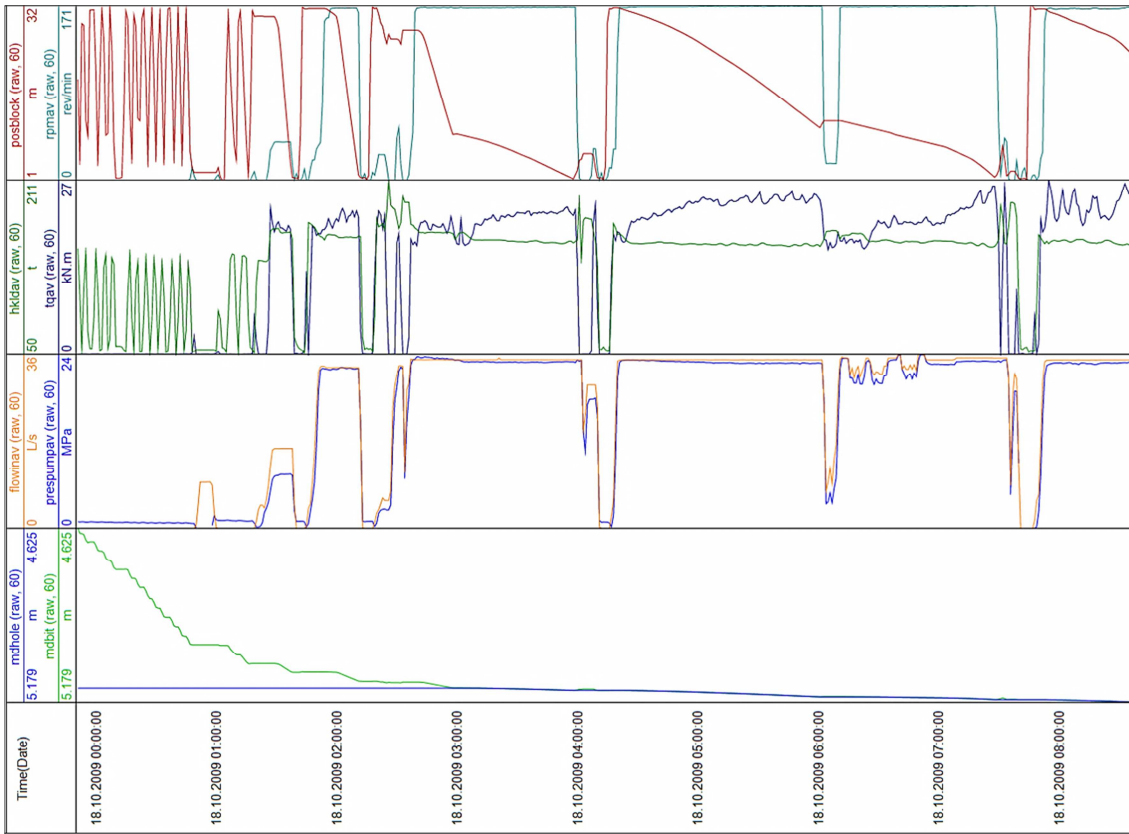


Figure 1. Drilling time series data

In this work, we improve the approach proposed in [5]. This approach is based on creating a compact representation of the raw sensor data in a given time range. The compact representation contains a set of statistical features calculated from the raw data.

Many papers suggest using statistical features to recognize patterns in time series. Lambrou [6] uses mean, variance, skewness, kurtosis and entropy as statistical features to classify audio signals. In [7], visual analytics techniques are used to explore the statistical features of sensors measurement. The results show how the statistical features are important in detecting different situations in underlying drilling process. Moreover, monitoring few features such as Skewness and Entropy can be considered as powerful tool to observe very critical situations (e.g. Stuck Pipes) in drilling process.

In this work, we focus on developing an approach that is not custom-made to solutions of specific application areas, but that will be applicable to other fields as well.

The remainder of the paper is organized as follows: Section II presents the general framework of our approach. Section III introduces the data preparation phase. Section IV shows the details of statistical features extraction phase. Sections V, VI and VII introduce the details of features ranking and feature selection phases. Section VIII shows the details of the classification task, and the last section IX displays the experimental results of the approach.

II. The General Framework

The proposed approach is simple but efficient; it uses the classical steps of data preparation, feature extraction, feature selection and classifier training. The general steps of the approach are sketched in Fig. 2 and described in more details later.

The input of the approach is the raw sensor-generated data which called “channels“. Most of the mud-logging systems provide 10 time series channels that represent the most important mechanical parameters. Table 1 describes the commonly-used channels.

TABLE 1. STANDARD DATA CHANNELS

Channels	Description
flowinav	Average mud flow-rate
hkldav	Average hook load
mdbit	Measured depth of the bit
mdhole	Measured depth of the hole
posblock	Block position
presumpav	Average pump pressure
ropav	Average rate of penetration
rpmav	Average drill string revolutions
tqav	Average torque
wobav	Average weight on bit

In other words, the input is a multivariate time series with ten variables (channels) $\{T_1, T_2, \dots, T_{10}\}$, where T_i is a series of real numbers $\{x_1, x_2, \dots, x_n\}$ made sequentially through time.

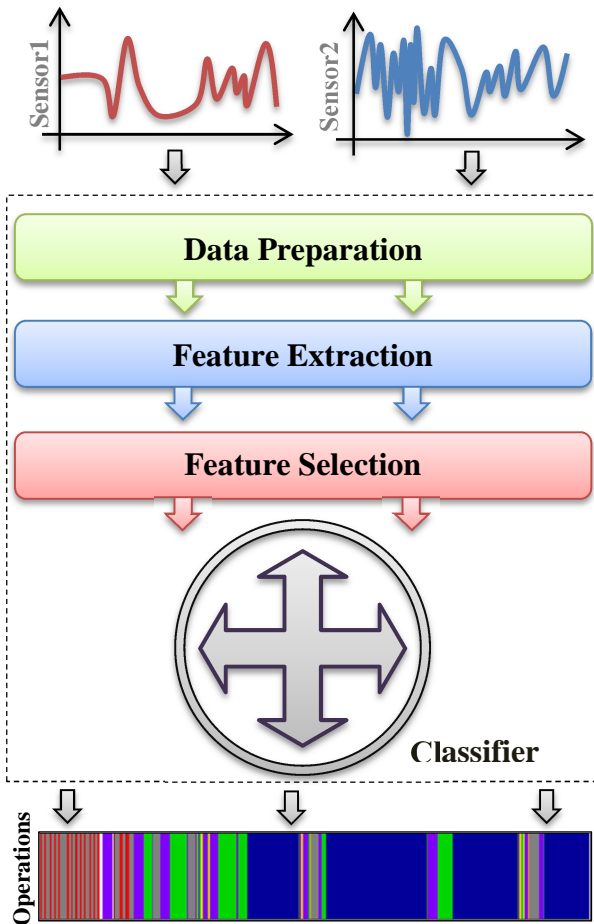


Figure 2. The general framework.

After receiving the input data, these data will be cleaned and prepared, a set of statistical features will be calculated, and then a subset features will be selected, and finally a classifier will be trained. Several techniques were used in the third and fourth phases in order to get the best performance in terms of speed and accuracy. The final output of the approach is a sequence of drilling operations that have different durations.

III. Data Preparation

The sensor-generated data are not directly ready for building the classification models. These data contain, in most cases, outliers and missing values that will influence the accuracy of the features calculation. Data cleansing is an elementary phase that should precede all others machine learning phases. In data cleansing task, two subtasks were executed which are:

- Identification and handling of missing values
- Identification and handling of outliers

Outlier is a numeric value which has an unusually high deviation from either the mean or the median value. Although there are numerous sophisticated algorithms for outlier detection, a simple statistical method is used in this work. This method is based on interquartile range (IQR)

which is a measure of variability of the data. IQR was calculated by this equation:

$$IQR = Q_3 - Q_1 \tag{1}$$

Here Q_1, Q_3 are the middle values in the first and the third half of the data set respectively. An outlier is any value x that is at least 1.5 interquartile ranges below the first quartile Q_1 , or at least 1.5 interquartile ranges above the third quartile Q_3 . One of these equations should be satisfied:

$$x < Q_1 - 1.5 \times IQR \tag{2}$$

$$x > Q_3 + 1.5 \times IQR \tag{3}$$

Boxplot (Box-and-Whisker plot) was used as a graphical representation of dispersion of the data. In other words, it was used to display the outliers graphically. Fig. 3 shows that there are no outliers in the “mdbit” and “mdhole” data taken from one drilling scenario. Fig. 4 shows the outliers in “Hook load” and “Block position” data taken from the same drilling scenario.

The length of the box equals to the difference between Q_3 and Q_1 which is IQR. The red line drawn inside the box represents the median value. All data points appear above the top horizontal line or below the bottom horizontal line will be considered as outliers.

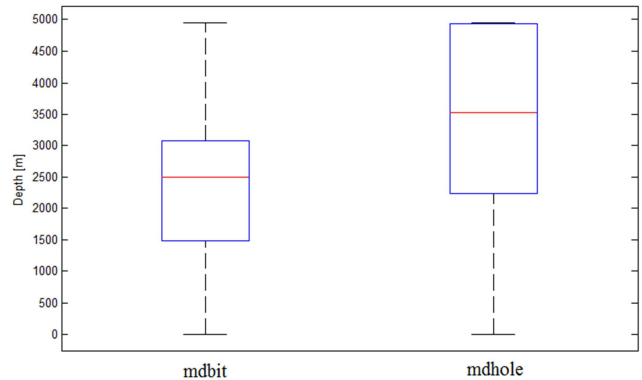


Figure 3. Boxplot for mdbit and mdhole channels

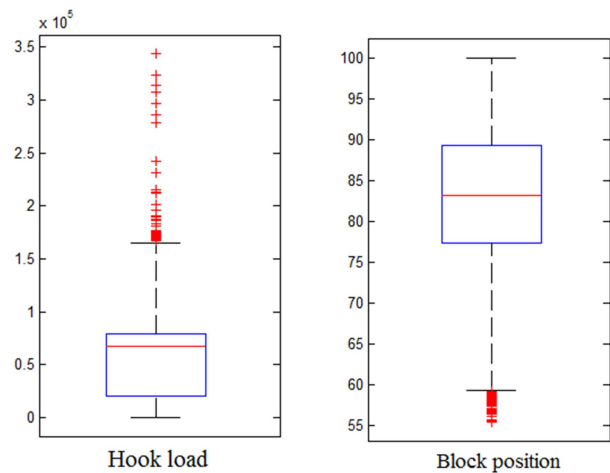


Figure 4. Boxplot for hkldav and posblock channels

After identification the outliers, these outliers as well as the missing values were substituted by the mean value. If x_i is considered as an outlier or missing value, then this value will be substituted by:

$$x_i = \frac{x_{i-1} + x_{i+1}}{2} \quad (4)$$

Although this data cleansing procedure is simple, it is very efficient and can remove most erroneous values from the data.

Data Normalization

Data were normalized to reduce unwanted variation between datasets as well as to allow data on different scales to be compared by converting them to a common unified scale.

Since the total depth of the drilling wells differ from a well to another, all channels that are related to the depth (e.g. "hklдав", "mdbit" and "mdhol") were normalized by dividing by the total depth of the selected well. The unrelated channels (e.g. "posblock" and "ropav") were used without normalization.

IV. Statistical Features Extraction

The second step of the approach is feature extraction, which is the transformation of patterns into features that are considered as a compressed representation.

Drilling time series data have a very high dimensionality, therefore mining such data is a challenge because a huge number of features can be extracted from the raw data [8]. To reduce the dimensionality of the data, a high-level representation is built where a set of significant features are calculated. These features provide an approximation of the original time series data.

For each time series variable $T_i = \{x_1, x_2, \dots, x_n\}$, $i = 1..10$ many statistical features were calculated to measure different properties of that variable. Fig. 5 shows the main groups of the calculated statistical measures, and below is some details about these measures.

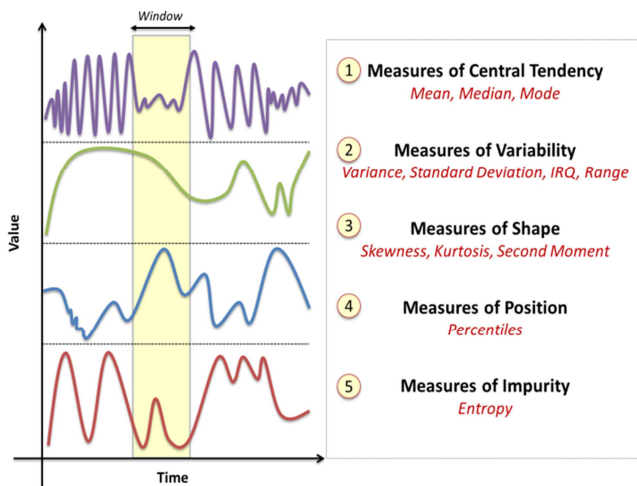


Figure 5. Statistical Features Extraction

A. Arithmetic mean (AM)

The arithmetic mean μ is the average of the values $\{x_1, x_2, \dots, x_m\}$ located within a time window. It was calculated by equation (5):

$$\mu = \frac{1}{m} \sum_{i=1}^m x_i \quad (5)$$

B. Standard Deviation

The standard deviation σ was calculated by equation (6) to measure how the values $\{x_1, x_2, \dots, x_m\}$ are spread out.

$$\sigma = \sqrt{\frac{1}{m} \sum_{i=1}^m (x_i - \mu)^2} \quad (6)$$

C. Standardized moment

It is the normalization of the k^{th} moment with respect to standard deviation. It was calculated by the equation:

$$\frac{\mu_k}{\sigma^k} \quad (7)$$

Here μ_k is k^{th} moment about the mean. The third standardized moment (skewness) and fourth standardized moment (kurtosis) were calculated and used as features.

D. Kurtosis

Kurtosis was calculated by equation (8) to measure the peakedness of the probability distribution of the data.

$$Ku = \frac{\mu_4}{\sigma^4} \quad (8)$$

where μ_4 is the 4th moment about the mean, and given by:

$$\mu_4 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu)^4 \quad (9)$$

E. Skewness

Skewness was used to measure the asymmetry of the data. It was calculated by equation (10):

$$Sk = \frac{\mu_3}{\sigma^3} \quad (10)$$

Where μ_3 is the 3th moment about the mean, and given by:

$$\mu_3 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu)^3 \quad (11)$$

F. Entropy

Entropy was used to measure the impurity associated with a random variable. The entropy H of a discrete variable X with possible values $\{x_1, x_2, \dots, x_m\}$ and probability mass function $p(X)$ is given by:

$$H(X) = - \sum_{i=1}^m P(x_i) \cdot \log_2 P(x_i) \quad (12)$$

G. Root Mean Square (RMS)

Root mean square is a measure of the magnitude of a set of values. It is the square root of the arithmetic mean of the squares of the original values. It was calculated using the following equation (13):

$$RMS = \sqrt{\frac{\sum_{i=1}^m x_i^2}{m}} \quad (13)$$

H. Percentiles

A percentile is the value of a variable below which a certain percent of observations fall. In other words, the p^{th} percentile is a value Y_p such that at most $(100 \times p)\%$ of the measurements are less than this value and $100 \times (1 - p)\%$ are greater.

Percentiles were used to measure the position. Five percentiles were calculated and used namely: p10, p25, p50, p75 and p90.

In addition to the above mentioned measures, basic statistical functions were calculated like sum, min, max, etc. Overall 22 statistical features were calculated for each channel namely: mean, median, mode, variance, standard deviation, root mean square, interquartile range IRQ, range, skewness, kurtosis, second moment, p10, p25, p50, p75, p90, min, max, sum, first, last and entropy.

The total number of calculated features equals: Number of channels times Number of features = $11 \times 22 = 242$ features. All these features were calculated using simple software written in Matlab. This software takes as input a list of channels and a time range (start timestamp and end timestamp) and returns the mentioned statistical measures.

V. Feature Ranking & Selection

High dimensional data, like our dataset, which has hundreds of features, can contain high degree of irrelevant and redundant information which might greatly reduce the performance of learning algorithms [9]. Therefore, feature selection becomes very necessary in our approach.

In the feature selection step, we seek to choose a subset of relevant features with high predictive value for creating robust learning models. Feature selection was implemented to improve the performance of our learning models by increasing the accuracy of the classifiers and speeding up learning and classification processes. In addition, feature selection improves model interpretability because it is much easier to tell an engineer that from hundreds of features these 10 are important to the classification task than to explain the influence of the hundreds features.

From a theoretical perspective, the best features can be selected using brute-force search, also known as exhaustive search of all possible subsets of features. For a dataset with n features, exhaustive search needs $(2^n - 1)$ possibilities. In our case we have 242 features yielding:

$2^{242} - 1 = 7.06 e^{72}$ possibilities to combine all the features. That means using exhaustive search is not feasible in finite time, and other selection algorithms should be considered.

The initial step in our feature selection phase is removing the correlated features in order to drop the dimensionality of the data and increase the computational efficiency.

A correlation matrix (242x242) was calculated to check the correlation strength between features, then we searched for highly correlated ones and removed one of them. In this step 24 features were removed.

VI. Feature Ranking

The fastest method for feature selection is ranking the features with some statistical test, and then selecting the k features with the highest score or those with a score greater than some threshold t . Such univariate filters do not take into account feature interaction, but they allow a first inspection of the data and most probably provide reasonable results [10].

We used 10 different feature ranking algorithms (described in table2) and measured the performance of them.

TABLE 2. FEATURE RANKING ALGORITHMS

Algorithm	Description
SAM	Calculates a weight according to "Significance Analysis for Microarrays"
PCA	Uses the factors of one of the principal components analysis as feature weights
SVM	Uses the coefficients of the normal vector of a linear support vector machine as feature weights
Chi Squared	Calculates the relevance of a feature by computing for each attribute the value of the chi-squared statistic with respect to the class attribute
Relief	Measures the relevance of features by sampling examples and comparing the value of the current feature for the nearest example of the same and of a different class
Gini Index	Calculates the relevance of the attributes based on the Gini impurity index
Information Gain	Calculates the relevance of the attributes based on the information gain
Correlation	Calculates the correlation of each attribute with the label attribute and returns the absolute or squared value as its weight.
Maximum Relevance	Selects Pearson correlation, mutual information or F-test depending on feature and label type (numerical/nominal).
Uncertainty	Calculates the relevance of an attribute by measuring the symmetrical uncertainty with respect to the class

Although the aforementioned algorithms did not produce identical results, there was about 70% of similarity between these results. For example most algorithms put flowin-p90, wobav-skewness, rpm-variance and prespumpav-range features in the top of the ranking list.

Feature number optimization

The resulting question now is: How many features should be used to get the best model in terms of accuracy? To answer this question, many tests were performed. We generated many models with different number of features and calculated the accuracy for each one. We started with

the top 150 features and then reduced this number to 100, 50 and 25. Table 3 shows the results. For most algorithms, models trained with 50 features have the best accuracy.

To select the best number of features accurately, PCA algorithm, which gives the best accuracy, was used to rank the features. We started with the top feature, and each time we added the next top feature until we finished all features.

TABLE 3. FEATURE RANKING COMPARISON (150, 100, 50 AND 25 FEATURES)

Algorithm	Accuracy [%]			
	150 F	100 F	50 F	25 F
SAM	80.29	81.19	75.12	66.06
PCA	83.29	81.38	85.72	80.74
SVM	80.59	81.09	76.33	66.06
Chi Squared	82.31	82.41	83.19	79.68
Relief	81.29	82.2	83.19	78.57
Gini Index	80.89	80.69	81.59	80.08
Information Gain	81.6	81.19	81.88	80.39
Correlation	80.89	84.22	83.3	80.21
Maximum Relevance	80.69	82	79.31	79.58
Uncertainty	80.89	82.61	85.51	82.91

Fig. 6 shows the accuracy curve as a function of the features number. It's clear that with 38 features we will get the most accurate classifier, but also with only 5 features we will get an acceptable result.

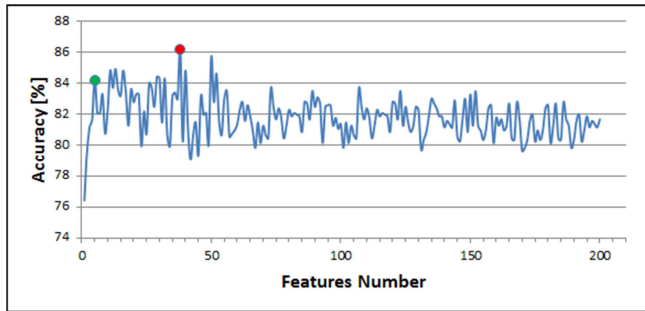


Figure 6. Accuracy curve as a function of the features numbers

VII. Forward Selection Methods

Forward selection method was used to bridge the gap between fast, but univariate filters, on the one hand, and slow, but multivariate exhaustive search, on the other hand.

Forward regression starts with creating models using exactly one feature. So we trained in the first step several networks using only the first feature as input, then the same procedure using the second feature as input and continued until the last feature was used as single model input [12]. The feature which yields the lowest error (ropav-p90) will be considered as the feature that has the most impact to the model.

In the second step we made new training runs with ropav-p90 as fixed input and adding exactly one of the remaining features as second input. We performed that procedure until all features were used as model input. Many networks were trained to obtain as result the ranking of the input with respect to the model error.

In Fig. 7 the results are sketched, ropav-p90 has the leading impact followed by wobav-skewness, mdhole-p75, etc.

The first error values in Fig. 7 give us the model errors using only ropav-p90 as input, the second values the errors using ropav-p90 & wobav-skewness as input, the third values the errors using ropav-p90 & wobav-skewness & mdhole-p75, etc.

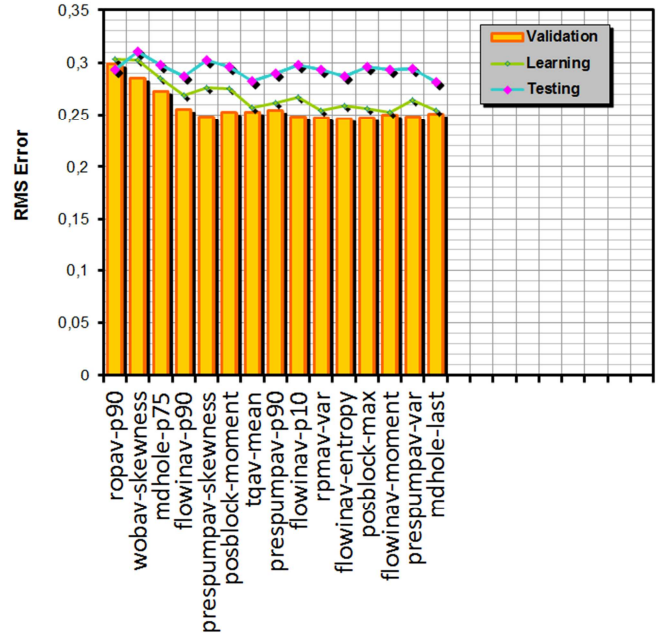


Figure 7. Forward Selection

VIII. Classifiers training

After extracting the features and selecting the most informative ones, we are ready to start classification process. Five classification techniques were used in this work. These techniques are: Support Vector Machine (SVM), Artificial Neural Network (ANN), Rule Induction (RI), Decision Tree (DT) and Naïve Bayes (NB).

Each one of these classifiers contains some parameters that can be tuned to improve the accuracy of the classification process. Numerous values and options of these parameters were tested to get the best results. Fig. 8 shows the structure of the neural network used in this work.

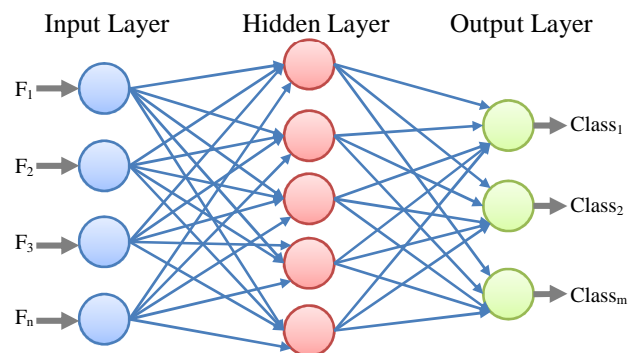


Figure 8. The structure of the neural network

A feed-forward neural network was trained by a backpropagation algorithm (multi-layer perceptron). The structure of the neural network has three layers: input, hidden and output. The input layer consists of input neurons which receive the input (statistical features). The output layer consists of output neurons which represented the classes (drilling operations).

The performance of the classifiers was evaluated by using the cross-validation method. We found that the worst classifier –in most cases – is Naïve Bayes, and the best one is support vector machine and rule induction.

IX. Experimental Results

To evaluate our approach, we collected data from four different drilling scenarios described in table 4. The time versus depth curve for scenario#1 is shown in Fig. 9, and the histogram of its operations is shown in Fig. 10.

TABLE 4. FOUR DRILLING SCENARIOS

Scenario	Instances	Duration [day]	Depth [m]	Classes
#1	991	95	7825	5
#2	1250	190	4402	9
#3	770	87	4863	7
#4	470	41	4004	4

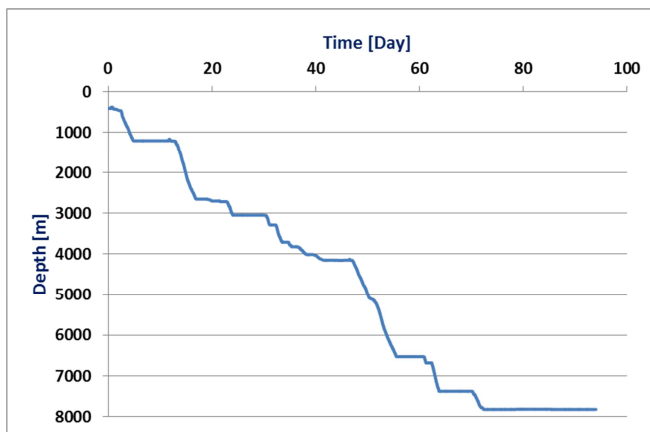


Figure 9. TxD curve of scenario#1

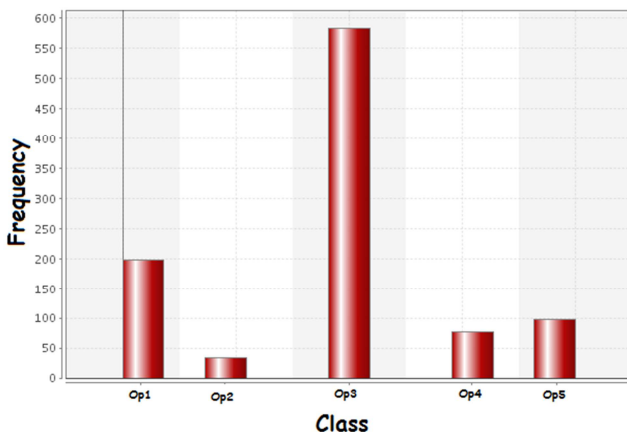


Figure 10. Histogram – operations of scenario#1

The proposed approach was applied to all scenarios. RapidMiner [13] was used to train neural network, rule induction, naïve Bayes and decision tree classifiers. LIBSVM [14] was used to train support vector machine classifiers which belong to the general category of kernel methods. The most important point was taken into account when using SVM is selecting an appropriate kernel, and determining the best parameters.

Most people randomly try a few kernels and parameters, and in most cases they cannot build an accurate classifier. In this work, the procedure proposed by Hsu et al. [11] was followed. This procedure consists of the following steps:

- Transform data to the format of an SVM package
- Conduct simple scaling on the data
- Consider the RBF kernel $K(x,y)=\text{Exp}(-\gamma|x-y|^2)$
- Use cross-validation method to find the best parameters C and γ .
- Apply the best C and γ to train the whole training set
- Test the classifier

To measure the improvement gained as a result of using feature selection, two groups of experiments were taken place. In the first group, all the classifiers were trained using the whole features set. In the second group, all the classifiers were trained only with the selected features.

Table 5 shows the results of the first group of the experiments. Table 6 shows the results of training the classifiers using only the top 38 features.

TABLE 5. CLASSIFICATION RESULTS (ALL FEATURES)

Scenarios	Accuracy [%]				
	ANN	RI	NB	DT	SVM
#1	78.2	79.08	65.02	72.55	79.3
#2	72.05	68.37	60.33	55.09	74.12
#3	78.12	78.90	63.95	75.26	79.10
#4	76.75	78.56	64.39	75.05	77.57

TABLE 6. CLASSIFICATION RESULTS (38 FEATURES)

Scenarios	Accuracy [%]				
	ANN	RI	NB	DT	SVM
#1	82.51	85.45	67.78	76.75	83.90
#2	70.84	70.51	63.33	54.12	72.4
#3	80.52	86.41	66.74	79.34	83.42
#4	81.90	85.96	67.45	78.59	82.70

The accuracy improvement rate is about 10%, and the classification and training process become much faster.

Acknowledgment

We thank TDE Thonhauser Data Engineering GmbH for the support of this work and the permission to publish this paper.

References

- [1] G. Thonhauser, "Using Real-Time Data for Automated Drilling Performance Analysis", OIL GAS European Magazine, Edition 4, 2004
- [2] Adriane B. S. Serapiao, Rogerio M. Tavares, Jose Ricardo P. Mendes, Ivan R. Guilherme, "Classification of Petroleum Well Drilling Operations Using Support Vector Machine (SVM)". In Proc. of the International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA'06), 2006
- [3] E. Keogh, M. Pazzani, "An enhanced representation of time series which allows fast and accurate classification clustering and relevance feedback". In Proc. of the 4th International Conference on Knowledge Discovery and Data Mining. New York, NY, Aug 27-31. pp 239-243 (1998).
- [4] B. Esmael, A. Arnaout, R. K. Fruhwirth, and G. Thonhauser. "Multivariate time series classification by combining trend-based and value-based approximations". In Proc. of the 12th International Conference on Computational Science and Its Applications - ICCSA 2012, pages 392–403, 2012.
- [5] B. Esmael, A. Arnaout, R.K. Fruhwirth, G. Thonhauser, "Automated system for drilling operations classification using statistical features", In Proc. of the 11th International Conference on Hybrid Intelligent Systems, HIS 2011, Melacca, Malaysia, December 5-8, 2011
- [6] T. Lambrou, P. Kudumakis. "Classification of Audio Signals Using Statistical Features on Time and Wavelet Transform Domains", IEEE, 2002
- [7] A. Arnaout, B. Esmael, R.K. Fruhwirth, G. Thonhauser, " Diagnosing Drilling Problems Using Visual Analytics of Sensors Measurements", 2012 In Proc. of IEEE International Instrumentation and Measurement Technology Conference I2MTC, Graz, Austria, 2012
- [8] C.A. Ratanamahatana, J. Lin, D. Gunopulos, E. Keogh, M. Vlachos, and G. Das, "Data Mining and Knowledge Discovery Handbook 2010". 2nd Edition. Eds. O. Maimon, L. Rokach. Springer. Pages 1049-1077, (2010)
- [9] L. Yu, H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution", Twentieth international Conference on Machine Learning (ICML-2003), 2003
- [10] B. Schowe, "Feature Selection for High-dimensional Data with RapidMiner", Technical University of Dortmund, 2010
- [11] C. Hsu, C. Chang, C. Lin, "A Practical Guide to Support Vector Classification", National Taiwan University, Taipei 106, Taiwan, <http://www.csie.ntu.edu.tw/~cjlin>
- [12] R. Fruhwirth, G. Thonhauser, "Hybrid Simulation Using Neural Networks To Predict Drilling Hydraulics in Real Time", SPE Annual Technical Conference and Exhibition held in San Antonio, Texas, U.S.A., 24–27 September 2006
- [13] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, T. Euler, "YALE: Rapid Prototyping for Complex Data Mining Tasks", in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06), (2006)
- [14] C. Chih-Chung, L. Chih-Jen, "LIBSVM: a library for support vector machines". ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, (2011). Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

Author Biographies



Bilal Esmael is a research assistant in the University of Leoben (Montan University). He is also a member of Monitoring and Diagnosis Research Group at TDE - Thonhauser Data Engineering GmbH. Bilal has experience in data analysis and machine learning techniques. He holds a M.Sc. in information technology from Al-Baath University (2003, Syria) and he is a Ph.D. student in the university of Leoben (since 2008, Austria)



Arghad Arnaout is a Software Engineer (Al-Baath University, 2003, Syria). Currently he follows up his PhD at Automation Institute (since 2008, Montan University, Leoben, Austria). Arghad also works as a research and development engineer at Thonhauser Data Engineering GmbH. He is a member of Monitoring and Diagnosis Research Group at Thonhauser Data Engineering GmbH (since 2006, Leoben, Austria).



Rudolf K. Fruhwirth is heading the monitoring and diagnosis research group of TDE - Thonhauser Data Engineering GmbH. He has been gaining almost 30 years of experience in research in the fields of drilling, geophysics and data analysis and computational intelligence. Rudolf is author of numerous papers and associate lecturer at the University for Mining and Metallurgy in Leoben.



Gerhard Thonhauser is a professor of drilling and completion engineering at the University of Leoben. He is also founder and managing director of TDE Thonhauser Data Engineering in Leoben, Austria. Prior to the founding TDE, Thonhauser served at the Commonwealth Scientific and Industrial Research Organization. He holds degrees from the University of Leoben and the Colorado School of Mines, and received a Ph.D. in petroleum engineering from the University of Leoben.