North Sea FLOW Measurement Workshop

Validity and consistency of MPFM data through a Machine Learning based system

T. Barbariol, Università degli Studi di Padova
E. Feltresi, Pietro Fiorentini s.p.a.
G.A. Susto, Università degli Studi di Padova

1 INTRODUCTION

Despite the increased interest on Multiphase Flow Meters (MPFM) in the last decades [1], the overall trust in the MPFM is still quite small, as the matter of fact less than a few percent of the oil fields worldwide employs MPFM to monitor the production [2]. However, monitoring the well flow and its composition has become very important as fields become economically marginal and reservoirs deplete [3]. Traditional methods, like oil separators, continue to be adopted despite their obvious disadvantages: they are expensive, bulky and do not allow continuous monitoring of the well performances. The reason behind this choice mainly relies on the lack of confidence in the MPFM. The separator is a conceptually simple instrument that does not require an act of faith to be trusted. On the contrary the MPFM is a complex system made up of different sensor modules that return their measurements to a flow model. The flow model processes the input data and gives back the estimated individual flow rates. Both these two stages can be prone to errors and malfunctions, but the sensor level is the most crucial and delicate. In order to overcome this mistrust, the MPFM manufacturer has to guarantee the customer the highest reliability.

In this work we will address the sensor data validation, developing a Machine Learning-based System for Self-diagnosis in order to increase the overall confidence in MPFMs. More specifically, in this paper we will extend a on board Anomaly Detection (AD) system for MPFM, previously described in [4]. This system will be used to assess the quality of the sensor measurements and to detect the module that is eventually malfunctioning. Detecting sensor anomalies in MPFM is challenging mainly because:

- we must decouple data related to metrology anomalies from data associated to new operating conditions;
- MPFM generated data are strongly affected by the flow and the well conditions;
- MPFM data are multivariate.
- detecting the root cause of a possible fault can be difficult due to the complexity of the MPFM system.

The developed system aims at overcoming those challenges, providing a reliability score associated with the measurements and a *guilty index* associated with the module responsible for the anomaly.

The contribution of this work can be summarized as follows:

- This is one of the first Machine Learning-based technologies for MPFM in literature;

- The work employs state-of-the-art unsupervised AD approaches like Isolation Forest [5] and Cluster-based Local Outlier Factor [6];

The rest of the paper is organized as follows: MPFM and its principles are described in Section 2. In Section 3, the data-driven approach is compared with respect to the model-based approach. The proposed Self-Diagnosis solution is explained in Section 4 with a brief review of the employed unsupervised AD methods. Experimental settings and results are shown in Section 5; finally, conclusive remarks and future works are discussed in Section 6.

2 MULTIPHASE FLOW METER

The MPFM is non-intrusive, in-line meter for measuring flow rates of oil, water and gas in the dispersed phase of the flow. In this work data have been collected using a MPFM [7] (an example of a MPFM can be seen on Figure 1) which combines the input of five sensors to compute the flow rate of oil, water and gas:

- *Venturi Module*: it measures the pressure drop across the inlet and the throat of a Venturi tube.
- *Impedance Module*: Several couple of electrodes, depending in the flow conditions, measure either the permittivity or the conductivity of the mixture. Moreover, the velocity of the flow is measure cross correlating the impedance measurements.
- *Gamma module*: the attenuation of a Caesium $_{137}$Cs radioactive source is used to measure the mixture density.
- Absolute pressure and temperature.

Figure 1 Skid mounted MPFM

3 DATA-DRIVEN ANOMALY DETECTION

Despite the importance of detecting malfunctions in the MPFM, the literature on fault diagnosis applied to multi-phase instruments is still not very broad [8].

The most natural way to tackle this problem is by employing a model-based strategy. However, the performance of the model-based approaches is strongly dependent on the quality of the model they rely on. Finding a model for the MPFM, in other words modelling all the modules and their interaction, can be particularly challenging.
In addition, the behaviour of the oil well where the MPFM is placed can be significantly different between other oil wells. Even the same well's flow conditions change over time. This makes the usage of fixed models without customization and periodic updates, hardly viable.

Another strategy is by employing data-driven approaches exploiting the availability of historical data. These model-free methods are usually divided into two main categories: supervised and unsupervised. Although powerful, supervised methods need data where the information 'Fault'/'Non-Fault' is present. Unfortunately, labelled datasets are hardly available: human domain experts have to manually label the dataset with time-consuming a posteriori visual inspection. Considering the large number of modules combinations, electronics and geometries, it's easy to understand the difficulties to use these methods.

Unsupervised techniques come in handy to overcome these issues: they are flexible methods that do not need labelled dataset.
In the context of Smart Monitoring, Anomaly Detection (AD) tools are fundamental instruments to detect anomalous behaviour in data, more specifically in industry the primary application of AD methods is to detect faults [9]. In complex/multi-dimensional systems like the MPFM, these techniques can be used to detect abnormal behaviour that even domain experts cannot notice and explain.
In literature there are many definitions of anomaly depending on the application of interest: generally speaking, anomalies are patterns in the data that do not conform to an expected behaviour [10]. Alternatively, the term 'outlier' is used to describe anomalies, in order to remark the fact that anomalies are few in numbers and different from the inliers. In a book on AD [11] it is written: "An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism.".

Given the advantages stated before, in this work we have chosen the data-driven approach to AD, and more precisely the unsupervised setting. Otherwise modelling the dynamic and multi-dimensional MPFM system would be a costly, perhaps impossible challenge.

The univariate control charts are the most classical monitoring approach in the field of data-driven techniques: they are very simple and define control limits above which an alarm is triggered. The main drawback of these methods is their inability to capture multivariate patterns. Multivariate control charts try to overcome this issue, but due to their strict assumptions remain applicable to a small number of applications.
In recent years the employment of Machine Learning to Smart Monitoring has led to a wide technological improvement. New data-driven AD methods have been developed when unsupervised techniques have been applied to industrial problems. These new tools are much more flexible and effective than control charts. AD methods return as output an index measuring the degree of data abnormality, named Anomaly Score (AS).

4 ML-BASED MPFM SELF-DIAGNOSIS APPROACH

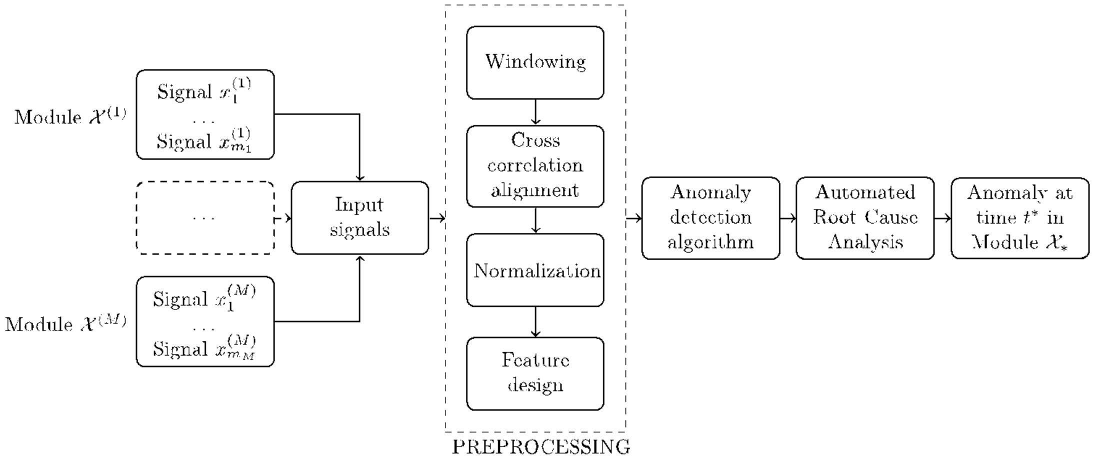

Figure 2 Flowchart of the algorithm.

As said above, in this article, we extend the algorithm proposed in [4]: in that paper a tool specifically suited to cope with the challenges described so far is presented. Such tool takes advantage of the MPFM structure: it exploits the presence of different metrology modules in order to have redundant and robust measurements. The algorithm (depicted in Figure 2) resorts on modern unsupervised AD approaches, preceded by an ad-hoc feature extraction (FE) and a windowing procedure. This strategy allows continuous monitoring and does not need a training process. The last part of the method, namely the Root Cause Analysis (RCA) block, is responsible to detecting which is the module responsible of the anomaly.

In the first part of this section the application peculiarities that led to the development of a new FE will be reviewed. In the second part, the implemented AD method will be described.

The most complex challenge is set by the flow conditions: given an oil well, the evolution of the flow composition and the mass discharge can vary widely over time. As a matter of fact, these conditions can vary also between wells of the same area. The flow can be classified as *homogeneous*, *bubble*, *annular* or *slug flow* when the oil well evolves, also the relation between the measured quantities changes, forming new previously unseen manifolds. Due to this fact, the task of detecting anomalous behaviours in the measuring instrumentation is highly complex. The most natural way to handle this problem is by filtering out the flow dynamics, leaving to the AD algorithm the task of detecting anomalies in the instrumentation. This is exactly what we have done, obtaining a method that is independent on the well operating conditions.

Figure 2 shows the structure of the algorithm: the signals are continuously collected from the different MPFM modules. Later, the signals are pre-processed and used to generate the features. These features are the input for the AD algorithm and the RCA box.

4.1 PRE-PROCESSING

The selected AD algorithms apply on fixed size batches; for this reason, the signals need to be windowed. The windowing settings are the window size and the time-overlap.

The proposed feature extraction relies on the following intuition: a set of correlated modules looks at the same physical process from different points of view. A variation in the flow conditions can be detected simultaneously in all the signals. If the informative content of one module significantly differs from the others, the first module is likely to be faulty. Summarising, the employed feature design is based on these assumptions:

- all sensors continuously measure different properties of the same underlying physical phenomenon;
- the sensors measurements are approximately linearly correlated.

In order to obtain good performances, the modules need to see the same event at the same time. To ensure this, the signals have to be aligned in time and the correlation between them have to be sufficiently high. Time alignment algorithms like Dynamic Time Warping [12] are too computationally expensive for this application therefore a time alignment based on cross correlation was preferred.

4.2 FEATURE EXTRACTION

After alignment, the signals have been windowed in small overlapping intervals. Each signal has its own scale and to make a comparison between them they need to be rescaled. This step is very simple but quite delicate:

- Subtracting the mean and dividing by standard deviation can be dangerous, due to the possible presence of outliers. To overcome this issue, we decided to normalise each signal window using the median and median absolute deviation.
- In order to be consistent between different windows of the same signal, we have defined a *reference* batch, whose median and median absolute deviation have been used to normalize the other batches.

The intuition in behind the FE can be encoded in the following feature design:

$$z_{ij} = x_i - sign(K_{ij})x_j \qquad z_{ij} = -z_{ji}$$

The features z can be obtained as the difference between all the normalised time series x; K is the correlation matrix between the signals. The $sign(\cdot)$ function is needed when the considered signals are negatively correlated.

4.3 ANOMALY DETECTION ALGORITHMS

An AD algorithm takes as input the features and gives back flags for anomalous conditions and an Anomaly Score AS index. This index measures the level of anomaly of the considered sample.

Unsupervised AD algorithms can be divided in many families depending on their working strategy. In this paper we have chosen to compare four different AD algorithms belonging to the most important families: (a) Cluster Based Local Outlier Factor (CBLOF) [6], (b) Principal Component Analysis (PCA) [13], (c) Angle-based Outlier Detection (ABOD) [14] and (d) Isolation Forest (IF) [5]. Some AD approaches that require long evaluation time, like k-Nearest Neighbour, have not been considered since they are not suitable for the application at hand.

Proximity-based AD techniques [15] define a data point as an outlier, if its locality (or proximity) is sparsely populated. The notion of proximity can be defined on the basis of clusters, distance or density metrics. CBLOF mixes the cluster and density strategies.

Linear methods assume that normal data are embedded in a lower dimensional subspace. There, outliers behave very differently from other points and it’s much easier to detect them. PCA is the most famous member of this class.

Probabilistic and statistical methods are a class of very general techniques. They usually assume an underlying data distribution. After the parameter training, the model becomes a generative model able to compute the probability of a sample to be drawn from the underlying distribution. ABOD belongs to the *extreme values* class, which in turn belongs to the probabilistic family.

Outlier ensembles approaches combine the results from different models in order to create a more robust one. IF is based on the average behaviour of a bunch of special trees called isolation trees.

5. EXPERIMENTAL RESULTS

The discussed algorithm has been applied both to semi-synthetic and real datasets. In this section the results of the simulation on synthetic datasets will be shown and analysed.

Four types of sensor faults are described in literature [16] [17]: namely *bias*, *complete failure*, *drifting* and *precision degradation*. When a sensor is subject to a *bias* fault its measurements are systematically corrupted by an additive constant; on the contrary, a *completely failed* sensor measures a constant fixed value. A *drifting* fault increases the sensor measurements by a growing factor while a *precision degradation* behaviour consists in an added noise. Obviously real faults can be combinations of the, aforementioned, typologies, but studying these fundamental faults is important for understanding, which are the best AD methods.

In the following the drifting and the precision degradation cases (Figure 2 and Figure 3) will be analysed due to their relevance in practical situations.

5.1 SYNTHETIC DATA GENERATION

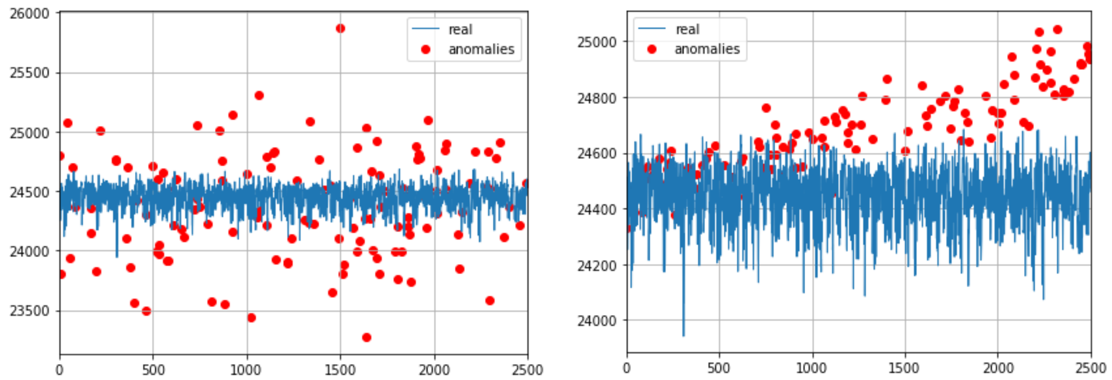

Figure 3 Example of anomalies injection (red circles) in real data (blue line). On the left hand-side precision degradation anomalies are simulated, whereas on the right-hand side drifting anomalies.

The real dataset employed in the synthetic fault generation is composed by three sets of data collected in flow loops. A set contains 150 operating points of 3 million samples each. Synthetic signals were made injecting anomalies inside real data in order to obtain a *sure* anomaly label. It's important to keep in mind that *natural* anomalies may already be present in the original signals.

5.2 FEATURE EXTRACTION

The proposed procedure is particularly suited for metrology instruments whose normal operating conditions are associated with high correlation between the modules.

The feature design process is able to decouple the normal process physical dynamics from the MPFM behaviour. By doing this, the AD algorithms are highly facilitated in detecting instrumentation anomalies. In fact, different flow regimes (e.g. mist or plug flows) show on data multiform structures that could hide anomalies inside, whereas the distribution, obtained once exploited the proposed method, shows a single cluster where anomalies stem radially.

5.3 COMPARISON OF ANOMALY DETECTION ALGORITHMS

In section 3 we have described the general operating principle of an unsupervised AD algorithm: after the computations it returns an index that measures the abnormality level of data, namely the anomaly score of each sample. Data with higher AS are more likely to be outliers. In order to assess which method performs better at detecting anomalies, the AS must be converted in a binary 'anomalous'/'not anomalous' with the aid of a threshold.

Note that in the real implementation of the algorithm, this threshold is the value above which the MPFM raises a malfunction alarm. This threshold setting can be quite challenging, and it is usually a compromise between *false positive* and *false negative* alarms.

In presence of an unbalanced dataset, as in the anomaly detection settings where outliers are much less than normal data, the *precision* (PREC) and *recall* (REC) metrics are more meaningful than *true positive* and *false positive rate* [18]. The precision is the number of true anomalies (i.e. the number of items correctly labelled as belonging to anomalies) divided by the total number of items labelled by the algorithm as anomalies. The recall is defined as the number of true anomalies divided by the total number of measures that are actually anomalies.

A common way to make a comparison between the performance of multiple classifiers is by using the F1-score or the AUC score: both of them summarize precision and recall measures. The F1-score (F1) is defined as the harmonic mean between PREC and REC: finding the classification threshold that maximises the F1 means finding the compromise between the highest precision and highest recall. Very roughly, when using the F1, the method that performs better is the one with highest possible F1, that is the one the has PREC and REC values closest to the (1,1). The AUC score is a global performance measure that does not depend on a specific threshold, like the one that maximizes the F1. AUC stands for Area Under the Curve and indeed it is the measure of the area enclosed by the curve in the PREC and REC diagram. Bigger is the AUC for a specific method, better this method will perform *on average*.

Since the threshold setting can be challenging because of the changing fault conditions, the most significative metric to make a comparison between different methods is the AUC score. In the following analysis we will use this metric, keeping in mind the importance of the F1-score.

5.3.1 SYNTHETIC FAULT: PRECISION DEGRADATION

The anomalies in precision degradation fault have been generated adding gaussian random noise. Five cases have been tested, changing the variance of the noise, in order to investigate if there exist any differences in the model responses. Figure 4 shows the REC-PREC curves computed in this experiment. As expected, the performance increases with the anomaly amplitude. In case of very small noise all methods but ABOD perform exactly the same. The interesting fact is that ABOD recovers quickly in terms of REC but not of PREC.

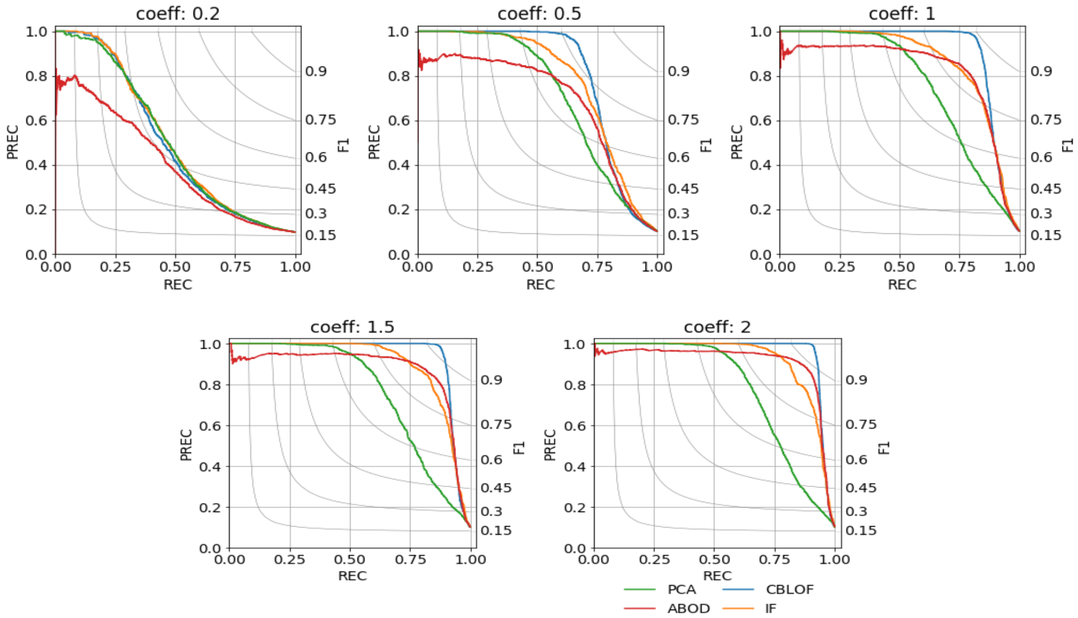

Figure 4 Precision and Recall curves. Performance of the AD methods for each degradation amplitude. 'Coeff' is a design parameter proportional to the variance of the synthetic noise. The grey contour lines on the plots represent values of F1-score, the value is reported on the right y-axis.

If we look at the same diagrams, but with a different perspective (Figure 5), we can observe more things. CBLOF performs bad in terms of REC when the anomalies are too small but improves its performance rapidly. It has large AUC and wins in every experiment. When the noise is not too high, IF and ABOD are similar in terms of AUC but IF outperforms the latter in terms of F1-score. On the contrary, PCA accumulates close to a limit curve.

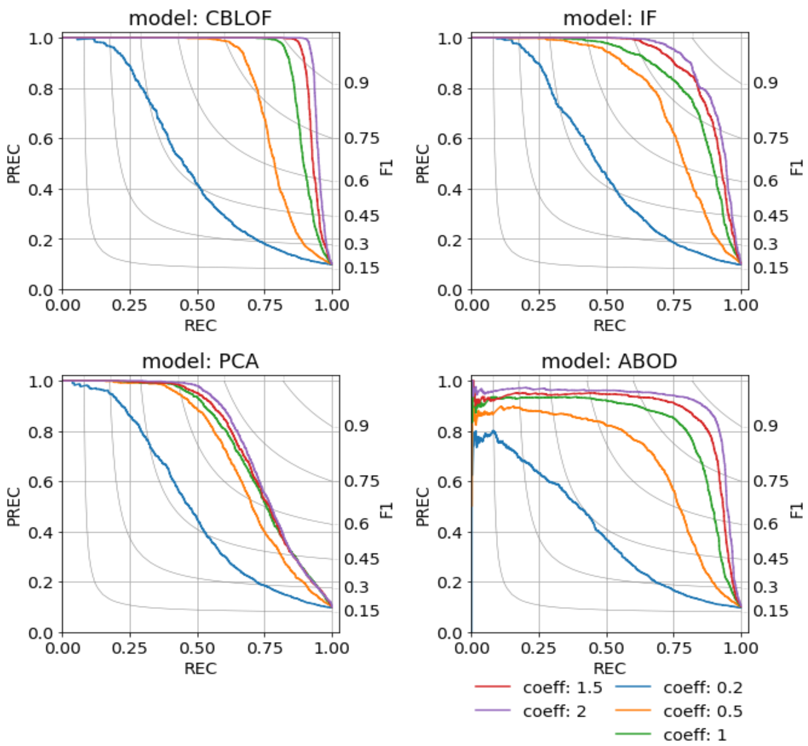

Figure 5 AD methods behaviour when changing the amplitude of the precision degradation fault. The grey contour lines on the plots represent values of F1-score, the value is reported on the right y-axis.

5.3.2 SYNTHETIC FAULT: LINEAR DRIFT

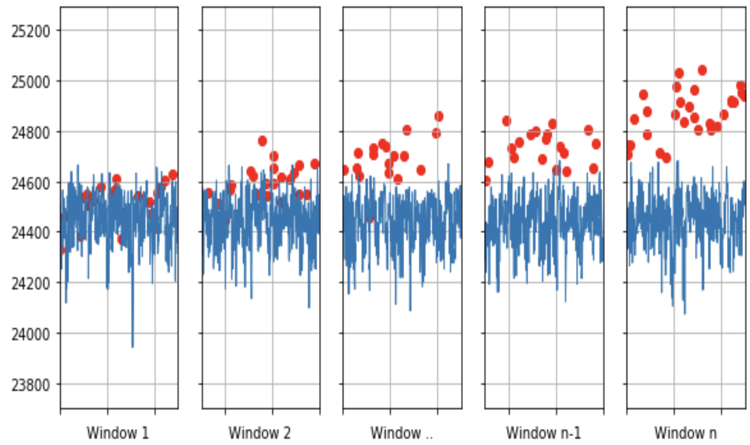

Figure 6 Windowing

In the linear drift case, the anomalies have been generated shifting the real sample by a value that increases in time. Unlike the previous case, this fault evolves, and every window has different anomaly settings (Figure 6).

The original signal has been windowed in 6 batches. The model response in different windows can be observed in Figure 7. In window-0 where the anomalies are too small, the methods are not able to distinguish them from the noise. In window-1 there is a big improvement, led by the CBLOF. In window-2 and 3 PCA and IF have similar F1-score but very different AUC.

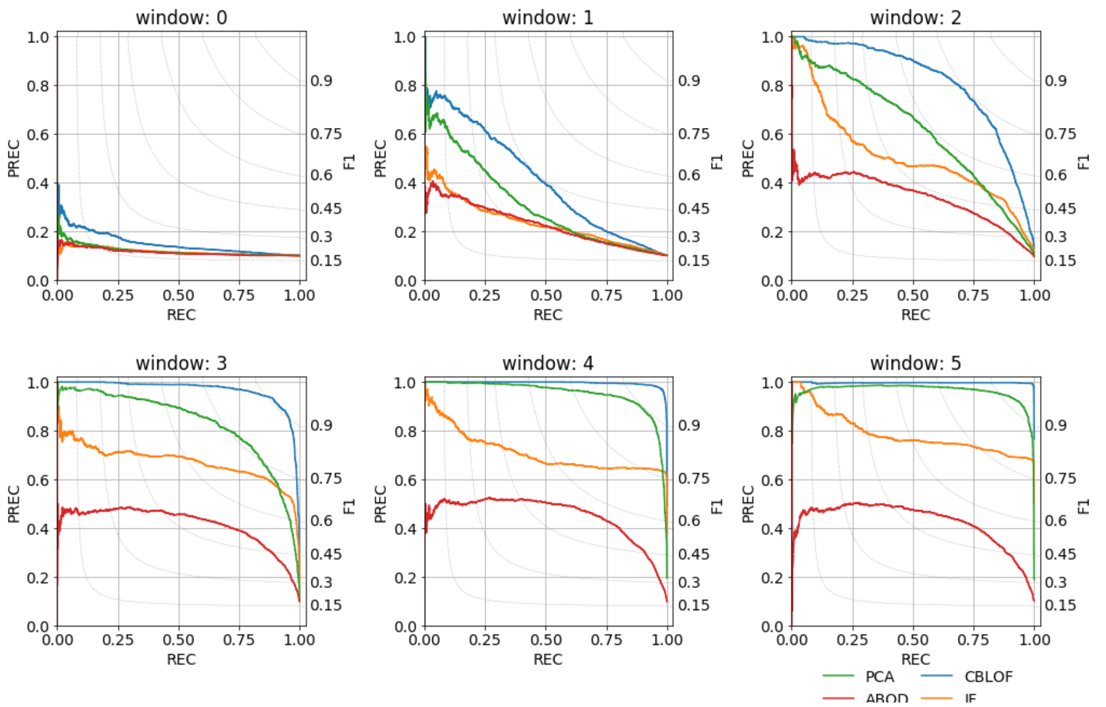

Figure 7 Precision and Recall curves. Performance of the AD methods for each window. The grey contour lines on the plots represent values of F1-score, the value is reported on the right y-axis.

Focusing on the methods (Figure 8), other important facts can be observed. CBLOF and PCA behave in a similar way, although CBLOF performs always better. IF has very good performance in terms of REC but has low PREC. ABOD always has awful results.

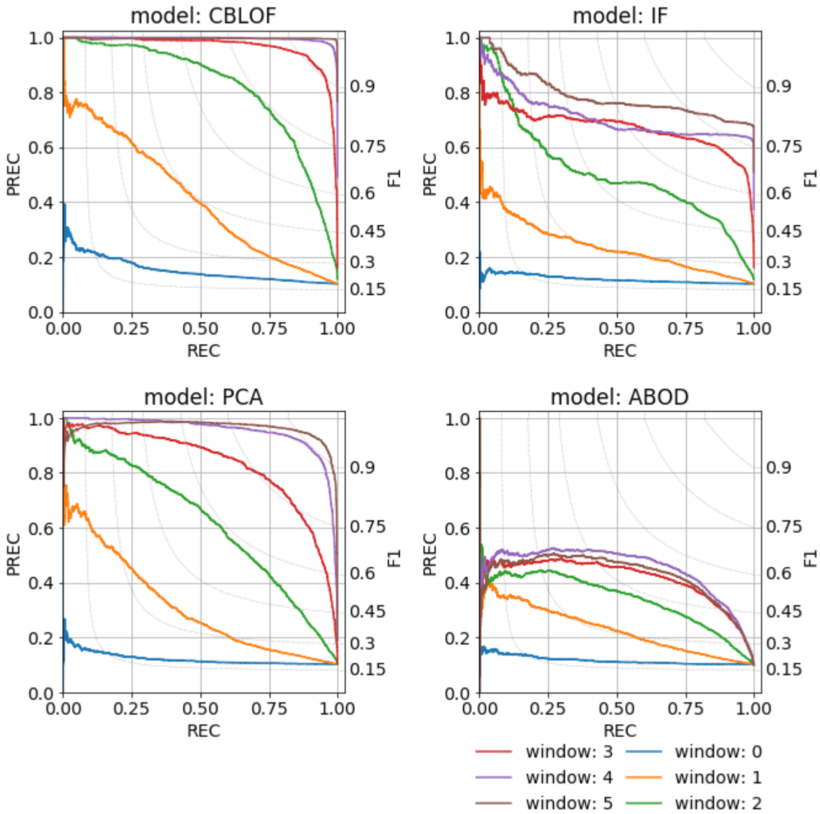

Figure 8 Precision and Recall curves. Performance of the AD techniques. The grey contour lines on the plots represent values of F1-score, the value is reported on the right y-axis.

	CBLOF	IF	PCA	ABOD
Computational time [seconds]	178.88	254.05	1.47	909.5

Table 1 Time complexity computed on a 2.7 GHz Intel Core 5 Processor with 8GB of RAM.

Three classes can be distinguished when looking at the computational time in Table 1. CBLOF and IF are 100 times slower than PCA, that is 1000 times faster than ABOD.

Despite the superiority of CBLOF, which method performs better between IF and PCA is not trivial: IF is good at detecting the precision degradation conditions, while PCA outperforms the former in the drifting case. As expected, ABOD performs well only when anomalies are extreme.

Summarizing, when the computational power is a constraint, the best compromise is the PCA. Otherwise CBLOF or a combination of the former has the best performances.

6. CONCLUSION AND FUTURE WORK

In this work we have tested our Machine Learning-based approach to Anomaly Detection for Multiphase flow meters with the most common types of failures.

We have proved that is possible to continuously monitor the MPFM signal acquisition, improving the reliability and the overall trust on this instrument. The system is able to promptly detect anomalous measures obtaining an indication of measure reliability.

Test were made, exploring two fault types and Anomaly Detection algorithms. Thanks to a very effective pre-processing procedure that allows to separate the well flow conditions and the instrument behaviour, the rest of the algorithm was greatly facilitated in finding the anomalies and detecting the faulty module.

Our approach does not need historical data to be trained and is ready for Plug & Play implementations.

The next step toward a fully trustable system is the development of an algorithm able to validate the data coming from the flow model. In future works we will investigate how to perform this task using supervised ML techniques.

7. REFERENCES

[1] S. Corneliussen, J.-P. Couput, E. Dahl, E. Dykesteen, K.-E. Frøysa, E. Malde, H. Moestue, P. O. Moksnes, L. Scheers and H. Tunheim, "Handbook of multiphase flow metering," *Norwegian Society for Oil and Gas Measurement (NFOGM), Revision,* vol. 2, 2005.

[2] R. Thorn, G. Johansen and B. Hjertaker, "Three-phase flow measurement in the petroleum industry," *Measurement Science and Technology,* vol. 24, 2012.

[3] T. A. AL-Qutami, R. Ibrahim, I. Ismail and M. A. Ishak, "Virtual multiphase flow metering using diverse neural network ensemble and adaptive simulated annealing," *Expert Systems with Applications,* vol. 93, pp. 72-85, 2018.

[4] T. Barbariol, E. Feltresi and G. A. Susto, "Machine Learning approaches for Anomaly Detection in Multiphase Flow Meters," in *ICONS - IFAC*, 2019.

[5] F. T. Liu, K. M. Ting and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD),* vol. 6, p. 3, 2012.

[6] Z. He, X. Xu and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognition Letters,* vol. 24, pp. 1641-1650, 2003.

[7] Pietro Fiorentini s.p.a., "MPFM FloWatch HS Datasheet," 2019. [Online]. Available: https://www.fiorentini.com/ww/en/product/components/mpfm_eng/flowatchhs/.

[8] Y. Yan, L. Wang, T. Wang, X. a. H. Y. Wang and Q. Duan, "Application of soft computing techniques to multiphase flow measurement: A review," *Flow Measurement and Instrumentation,* vol. 60, pp. 30-43, 2018.

[9] L. Meneghetti, M. Terzi, S. Del Favero, G. A. Susto and C. Cobelli, "Data-Driven Anomaly Recognition for Unsupervised Model-Free Fault Detection in Artificial Pancreas," *IEEE Transactions on Control Systems Technology,* 2018.

[10] V. Chandola, A. Banerjee and V. Kumar, "Anomaly detection: A survey.," in *ACM computing surveys (CSUR)*, 2009.

[11] D. M. Hawkins, Identification of outliers, vol. 11, Springer, 19080.

[12] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems,* vol. 7, pp. 358-386, 2005.

[13] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *Proceedings of International Conference on Data Mining*, 2003.

[14] H.-P. Kriegel, M. Schubert and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008.

[15] C. Aggarwal, Proximity-Based Outlier Detection BT - Outlier Analysis., New York: Springer, 2013.

[16] R. Dunia, S. J. Qin, T. F. Edgar and T. J. McAvoy, "Identification of faulty sensors using principal component analysis," *AIChE Journal,* vol. 42, no. 10, pp. 2797-2812, 1996.

[17] S. Narasimhan and C. Jordache, Data reconciliation and gross error detection: An intelligent use of process data, Elsevier, 1999.

[18] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PloS one,* vol. 10, no. 3, 2015.

[19] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," *KI-2012: Poster and Demo Track,* pp. 59-63, 2012.

[20] Y. Zhao, Z. Nasrullah and Z. Li, "PyOD: A python toolbox for scalable outlier detection," *arXiv preprint,* no. arXiv:1901.01588, 2019.