When should a Machine Learning model be retrained? A case study with digital twins of a wet gas flow meter

Marcelo Ferreira de Souza Alves [a], Gildeir Lima Rabello[a], Diego Queiroz Faria de Menezes[b], Thainá Menezes de Melo[b], Luiz Octavio Vieira Pereira[c], José Carlos Pinto[a]*

[a] Programa de Pós Graduação em Engenharia de Processos Químicos e Bioquímicos / Escola de Química / Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro / RJ, Brazil
[b] Programa de Engenharia Química / Coppe / UFRJ, Rio de Janeiro / RJ, Brazil
[c] Petróleo Brasileiro S.A., Rio de Janeiro / RJ, Brazil
**marcelo.alves@eq.ufrj.br*

ABSTRACT

Data evolves continuously and, consequently, a previously trained Machine Learning model might become outdated, with its performance severely worsening. In this context, it is of utmost importance to monitor when a deployed models needs to be retrained. In the present study, a one-year operational dataset of a wet gas flowmeter is considered to develop eleven analogous regression neural networks, but with an increased number of training instances, to act as digital twins for the device. Six approaches are then tested to alarm the need for retraining. In a scenario where a ground truth is not available, heuristics and inputs from the user prove themselves crucial.

Keywords: Retraining; Concept Drift; Data Drift; Data-driven; Machine Learning.

1. Introduction

Machine Learning models are typically trained on available data and subsequently used to generate predictions on previously unseen data (Lu et al., 2018). However, Machine Learning models can lose predictive performance over time due to either data drift or concept drift (or both); the former refers to the distribution of the data changing between training and usage while the latter refers to the relationships between features and outputs changing (Lu et al., 2018; Pham et al., 2025). It should be noted that, although with different meanings, it is common to see only the terminology 'concept drift' being used to refer to both phenomena (Pham et al., 2025) or, in general, to the statistical properties of a target domain changing while the specific sources of the changes (i.e., changes in distributions, relationships or both) are defined separately (Lu et al., 2018).

In this context, it is important that users try to detect eventual drifts, understand when they begin, how severe are its impacts and where they interfere the most in terms of variable space; if needed, a final important understand is how to adapt to change (Lu et al., 2018). Furthermore, it is worth noting that the changes in concept might be sudden, gradual, incremental or reoccurring (Lu et al., 2018). Aiming on maximal performance, Florence et al. (2025) state that such adaptation to drifts may be found in the literature by models that have a (limited) resilience to drifts, approaches where a continuous learning is considered for the models (which may include transfer learning) and approaches where retraining of the model is considered whenever a significant drift is detected, which is the interest in the present work as we consider this the more general case and where the user has greater control over the process. Then, the identification of when a model that was deployed in an application needs retraining constitutes a significant challenge; particularly, as data and data distributions typically evolve continuously (Florence et al., 2025; Pham et al., 2025). However, we emphasize that the retraining costs (computational and financial), an important concern highlighted by Florence et al. (2025), is not explicitly considered in the present paper in the scheduling of retraining as they are small, as discussed in the results Section; then, a good approach might be, indeed, to retrain the models whenever a shift is detected (Florence et al., 2025).

Lu et al. (2018) state that there are three main classes of drift detection. The authors argue the most popular is the error-based drift detection, which is based on checking if there are significant differences in the error of a prediction model along time, typically by using statistical testing. The second most popular class of methods, typically related to higher computational costs, is argued as the distribution-based drift detection, which typically measures the statistical significance of the dissimilarity between the distribution of historical data and some new data, usually contained in predefined windows. Finally, some algorithms rely on multiple hypothesis testing, which

can be performed in parallel or hierarchically. Pham et al. (2025) further state that drifts may be identified by supervised, semi-supervised or unsupervised algorithms. The authors highlight that unsupervised techniques typically only identify data drifts, assuming that concept drifts would occur simultaneously, and highlight that fully supervised techniques may be unfeasible in real industrial settings as not all data may be labeled.

In the present study, we deal with wet gas flows, that is, streams with over 95% in volume of gas, but that also present liquids (Salehi et al., 2023; ISO, 2012). Particularly of interest is a wet gas flow meter (WGFM) installed near the X-mas tree of an actual wet-gas-producing well located off-shore. The WGFM is a multiphase flow meter device whose technology has been gaining growing acceptance as an alternative to assess the flowrates produced by each individual well (Niblett and I. Robertson, 2011) in a context in which several wells are typically operated simultaneously in a platform with only the comingled production being measured, while the individual assessment is of utmost importance for regulatory, environmental, safety and optimization purposes (Niblett and Robertson, 2011; Bikmukhametov and Jäschke, 2020).

Recently, Alves et al. (2025a) and Alves et al. (2025b) presented for the first time a detailed evaluation of data preprocessing, feature selection, and development of data-driven models (neural networks and principal component analysis) using data from a real wet gas production site to develop a digital twin for a WGFM device, a development important for enhancing the technical and economic feasibility of WGFM at plant sites as virtual sensors can address hard-sensor failures, filter random errors, replace gross-error measurements, and be applied on data reconciliation. Particularly for neural network models, the main interest in the present paper, Alves et al. (2025a) reached 95th percentile relative prediction errors between 0.4 to 2.2 for the three flowrates of interest in an independent testing set sampled randomly from a range of the values of variables seen in training while Alves et al. (2025b) showed that even the simple models developed could be used in a data drift scenario, with 3.0 to 6.8% 95th percentile relative prediction errors across a five-month period. However, despite the good results, two open questions are yet how long the models can be confidently used before significant performance degradation and how small can the training set be, both of which the present study aims on understanding.

For that, we consider a one-year operational dataset of a WGFM device and develop 11 neural-network based digital twin with increasing amounts of training instances. Besides performance evaluation on training and validation sets for each trained model, the performance on unseen data, simulating an online deployment of the digital twin, is monitored over time. Particularly, six drift detection approaches are investigated to determine how long it takes for the performance to degrade upon deployment of the digital twin.

2. Methodology

A one-year operational dataset, sampled at 5-minute intervals, from a WGFM device installed in an offshore producing well is considered in the present study. The device (model Roxar Subsea Wetgas Meters, Emerson) integrates multiple sensors with a proprietary data-processing scheme to provide measurements of gas, condensate oil and water flowrates. A v-cone sensor provides differential pressure measurements, which are used to estimate the overall flow. A microwave sensor measures amplitude, frequency and Q-factor of waves interacting with the sample, which help assess the water content. In addition, pressure and temperature sensors, together with user-input fluid composition, are incorporated into thermodynamic calculations within the device's black-box processing to estimate phase densities as well as gas and condensate phase fractions (Hellestø et al., 2020).

Based on phenomenological considerations, 13 variables in the dataset were selected as independent features for the development of data-driven neural network models to act as digital twins for the hard sensor. These features comprise the pressure drop across the V-cone inside the WGFM; amplitude of the wave, frequency of the wave and Q factor measured by the microwave sensor, considered both in the instantaneous readings and proprietary-calculates mean values; densities of gas, condensate oil and water phases; pressure and temperature measured at the device; and the opening of the choke valve of the well. The outputs of interest were the mass flowrates of gas, condensate oil, and water.

With a Dell Precision 3581 workstation, with a 13th Gen Intel(R) Core(TM) i9-13900H processor, used throughout all the study, eleven different neural network models were developed with increasing amounts of historical data for training. The dataset was sequentially split in time: data from the first month up to the eleventh month were considered for training in one-month increments. For each configuration, 10% of the corresponding initial time window was reserved, with random sampling, for validation (referred to as the "interpolation test set"). The maximum evaluation period for "extrapolation testing" was therefore limited to (12 − T) months, where T denotes the number of months used for model development. Before model development, the data were subjected to preprocessing. First, considering the entire dataset, samples corresponding to non-physical conditions (negative pressure drops or flow rates) were discarded. As the focus of this study is the predominantly quasi-steady-state operation of the well, the first 20 samples after each well reopening - identified heuristically by inspection and

characterized by strong transient behavior - were also removed. Occasional NaN ("Not a Number") values, possibly associated with sporadic sensor failures, were imputed by carrying forward the last valid observation. Second, normalization parameters were computed exclusively from the training set of each model to avoid data leakage. The input variables were normalized using the standard Z-score. To mitigate the influence of potential outliers, the output variables were also normalized using a Z-score approach; however, their mean and standard deviation were calculated considering only data between the 25th and 75th percentiles. Additionally, for the training sets, a trimming procedure was applied to the normalized outputs, removing extreme values beyond ± 20.

Each preprocessed training dataset was then used to develop feedforward neural networks with multiple inputs and multiple outputs (MIMO) using the MLPRegressor function from the Scikit-learn library (Pedregosa et al., 2011). The networks consisted of one input layer with 13 neurons, one hidden layer with a number of neurons optimized between 3 and 30, and one output layer with 3 neurons. Hyperparameter optimization was conducted with the help of Optuna library (Ozaki, Watanabe and Yanase, 2025) and, beyond the number of neurons, the L2 regularization parameter (alpha, searched within the range 0.0001 to 0.01) and the activation function (ReLU or sigmoid) were also investigated. Early stopping was enabled to mitigate overfitting, while all other hyperparameters were kept at their default values.

After training, predictions of the output variables were generated over the extrapolation testing period. Since the models operate as digital twins of an already deployed WGFM device, the reference flow rate values are continuously available, allowing error evaluation without the need for manual labeling. The adopted error metric was the absolute difference between predicted and measured values in the original physical units (after de-normalization), reported in relative terms with respect to the measured value. Importantly, no single aggregated performance metric is computed for the entire testing period; instead, the prediction error is monitored over time, a strategy that is suitable both for the present analysis and for online performance monitoring in real-world deployment scenarios. These time-dependent relative errors, particularly over the testing subset, were used to estimate the duration for which each model could operate without retraining. Five strategies were initially tested, which were based either on the ADWIN technique, an adaptive windowing algorithm widely used for concept drift detection (Lu et al., 2018) that identifies distribution changes through statistical testing (Bifet and Gavalda, 2007), or on a fault detection-inspired approaches. In the latter, a performance metric computed during training is considered to define control limits for monitoring during operation; if the monitored performance consistently exceeds such limit, a fault is detected (Spina et al, 2024) or, in the present case, a retraining alarm is triggered. Additionally, a practical criterion was evaluated in which retraining was triggered when the daily relative prediction error exceeded 5% for three consecutive days, which is referred to as "Threshold (3X)" in Table 1.

For the ADWIN-based strategies, the implementation available in the River library for Python (Montiel et al., 2021) was employed with default settings unless otherwise stated. Three scenarios were tested: (i) monitoring the average daily prediction error over time; (ii) monitoring the prediction error at each 5-minute sampling interval; and (iii) monitoring the 5-minute prediction error after initially feeding the model with the training errors (without drift assessment during this phase, by setting the "grace_period" parameter equal to the size of the training dataset). In Table 1, such strategies are referred as "ADWIN#1", "ADWIN#2", and "ADWIN#3", respectively. For the fault detection-based strategies, the monitoring threshold was defined as the mean training error plus three times its standard deviation. Two alarm logics were considered: triggering retraining upon the first threshold exceedance, and triggering retraining only if the threshold was exceeded in three consecutive evaluations, referred to as "SPE (1X)" and "SPE (3X)", respectively in Table 1.

It should be noted that due to expected ageing effects in the wet gas well operation, data distribution shifts are anticipated over time. Finally, for conciseness and due to space limitation, the intervals until a retraining alarm are summarized in tabular form, but time series are only shown for the model trained with 9 months of data.

3. Results and Discussion

Figure 1 shows the measured and predicted values for the normalized flowrates of interest; the splitting of dataset and predicted values refer to the model trained with the first 9 months of data available, which is considered a representative case. Across all the models developed, it was noted that a good representation of the training ("Train") and validation ("Test: int", referring to interpolation testing) sets were achieved; for the most extreme case, the model developed with the one-month (8640 instances) training set, the 95th percentile of errors were below 0.12% for all the predicted flowrates in the training and validation intervals. When evaluating the extrapolation test sets ("Test: ext"), however, the observed errors were higher; particularly, they typically increased over time during usage (as illustrated in Figure 2), which can indicate model degradation or predictions over new operational regimes that were not seen during the available training set. This justifies the question of how long the

models could be used for without an update. Additionally, it should be noted that the computational cost associated with hyperparameter optimization and model training was modest. Even for the most demanding scenario (11 months of data for model development), the total computational time remained below 8 minutes. Considering typical industrial deployment conditions, such processing times were regarded as small and do not represent a practical limitation for retraining scheduling.

Figure 1. Measured and predicted values for the normalized flowrates of gas, condensate oil, and water predicted by the neural network model trained with 9 months of data. On the labels, the word 'train' refers to the actual training data considered, 'Test: int' to the validation dataset, and "Test: ext" refers to extrapolation testing, which represents online deployment of the model.

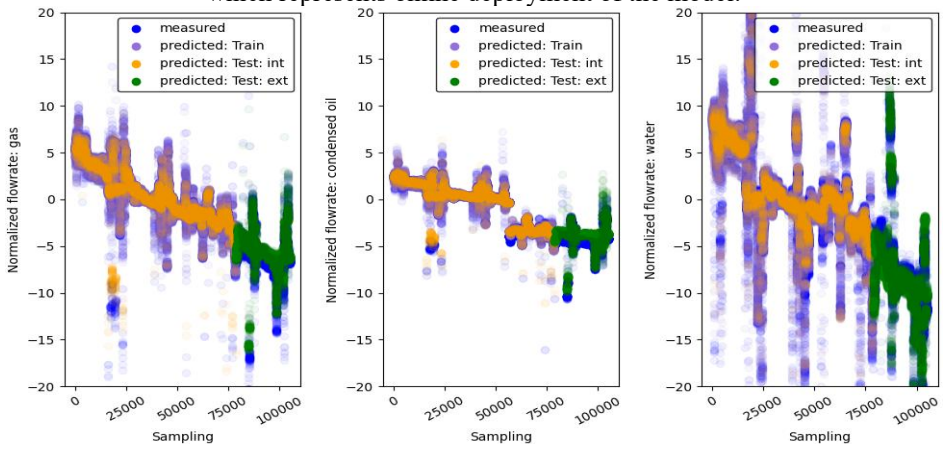

Figure 2. Squared Prediction Error (SPE) of the flowrate of gas predicted by the neural network model trained with 9 months of data.

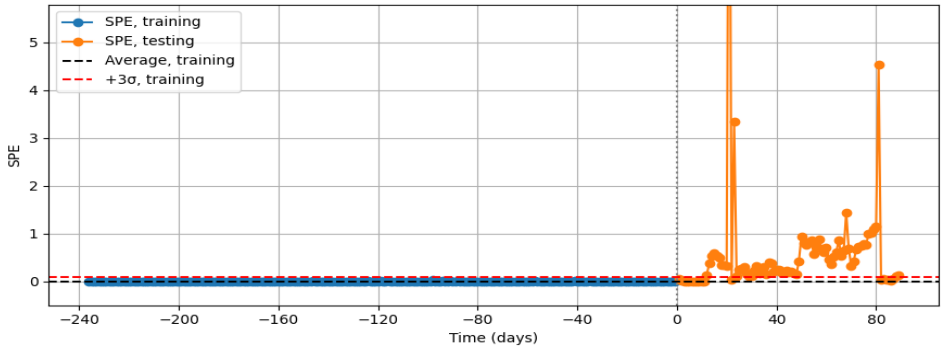

Figure 2 illustrates that the despite the expected and evidenced data drift (Figure 1), the trained model could still be deployed for usage with similar error metric as the obtained in the training set, but such similarity can change over time. This is systematically observed for all the trained models. In this context, Table 1 summarizes the formal evaluation of the need for retraining. Table 1 shows the number of days until the retraining alarm is triggered according to each of the 5 investigated approaches and evaluating each of the three flowrates of interest (gas, condensed oil, and water). However, it should be noted that the complete dataset available for both training and testing is 12 months (one year) long, such that, if "T" months are considered for training (and validation), only "12-T" months will be available for testing such that the retraining alarm can be triggered only up to the corresponding number of days. If not triggered during the available days of testing, a * is present, which should be understood as the model being reliably usable for at least the corresponding tested interval. It is relevant to state that the combined execution time of all detection algorithms in Table 1 was approximately 0.1 second per evaluated day.

According to Table 1, ADWIN algorithm relying on daily error information tends to indicate the longest times of reliable usage of the models when considering the tests with ADWIN and SPE. For such ADWIN implementation, a curious behavior is observed for the models with one to five months of data in the training set: the reliable usage would shorten as more data is available for training, but the end of reliability coincides with an

abrupt change in the time series of condensate oil flowrate. Conversely, the other approaches using ADWIN tended to alarm retraining in less than a week, in general, despite the training size. Based of visual inspection of the model error over time (as the example depicted in Figure 2), the fault detection approach using SPE is argued more consistent with the comparative behavior of the models across training and testing. However, it should be highlighted that, given the regression nature of the investigated neural network models, there is no prior ground truth about when a retraining should be alarmed. In that context, inputs and choices of the user would be of utmost importance to such alarms; for example, by defining hyperparameters of the tested approaches, such as how many consecutive crossings of the threshold in the fault detection approach are tolerated before an alarm is triggered or the inputs of the ADWIN model. Additionally, as a ground truth time until retraining is not available, hyperparameter optimization might not be feasible without user influence. Furthermore, it was observed that, despite a tendency of an increase in the prediction errors over time during testing, the changes in error over time were not monotonically. In Figure 2, for example, after an increase in prediction error on day 12 of testing, which triggered a retraining alarm, the errors decrease around day 20. Finally, it should be highlighted that despite evident increase of error over time during testing, in general, the error metric was modest. For example, in the most extreme case, of only one-month training set, the 95th percentile of relative errors when considering the 11-month testing was 4.5%, 22.2% and 14.0% for gas, condensed oil, and water flows respectively. One additional possibility of indicating a need for retraining could be simply measuring when the error consistently exceeds a pre-defined threshold. Table 1 also shows the results of such procedure considering 3 days in a row crossing a daily-average 5% prediction error; in that case, the majority of the models, at least concerning the most-important gas flow, could be used along all the evaluated time interval, such again evidences the important of user inputs on final conclusions.

Table 1. Number of days until a retraining alarm is triggered according to different monitoring strategies. Each cell contains the values corresponding to gas, condensate oil, and water flowrates, respectively. * indicates that no retraining alarm was triggered within the maximum available testing period.

T	SPE (1X)	SPE (3X)	ADWIN#1	ADWIN#2	ADWIN#3	Threshold (3X)
1	23, 26, 14	23, 26, 14	192, 192, 192	23, 6, 5	10, 5, 3	*, 163, 163
2	1, 1, 1	1, 1, 1	160, 160, 160	3, 3, 2	1, 1, 1	167, 133, 133
3	8, 8, 8	74, 59, 60	128, 128, 128	41, 3, 2	8, 3, 2	*, 104, 104
4	21, 21, 27	52, 57, 61	96, 96, *	8, 8, 4	8, 2, 2	*, 73, *
5	36, 34, 5	43, 34, 32	64, 64, 64	14, 3, 2	4, 1, 1	*, 44, *
6	7, 3, 3	41, 3, 3	160, 64, 64	7, 3, 3	4, 2, 1	*, 15, 15
7	3, 3, 3	73, 3, 3	128, 96, 128	3, 3, 3	3, 1, 2	*, 121, *
8	14, 4, 4	*, 27, 30	*, *, *	5, 4, 4	4, 2, 1	*, *, *
9	12, 13, 13	12, 13, 13	*, *, *	3, 3, 2	1, 1, 1	*, *, *
10	38, 50, 8	*, 50, 8	*, *, *	3, 7, 2	1, 2, 1	*, *, *
11	8, 21, 20	*, *, *	*, *, *	7, 3, 3	4, 1, 3	*, *, *

4. Conclusions

In the present study, eleven feedforward neural networks were trained with increasing historical data to act as digital twins of a wet gas flowmeter installed in an offshore well. The study aimed to understand how few training instances could be enough to develop a reliable model and for how long such models could be applied in a data-drift reality until severe performance degradation. More importantly, the study proposes a framework that can be used for continuous online performance monitoring of actual deployments of models. For the present offline study, it was observed that even the model trained with the fewest data instances (one month) achieved small errors during training and validation (95th percentile of 0.12%); however, prediction errors generally increased over time during simulated deployment, reinforcing the need for performance monitoring and retraining strategies.

The simultaneous analysis of drift detection algorithms and time series of errors and flow rates indicate that abrupt changes in the process can alarm retraining. This is particularly true when considering the earliest alarm among the three predicted flow rates and the threshold metric, the one we could best interpret in light of the time series. Based on such criteria, the retraining alarm for the first six models would coincide with abrupt changes in the condensate oil flow rate, in Figure 1. Still based on such criteria, it should e noted that no more than six months

passed between a retraining alarm. Based on these findings, it would be suggested that models, on this particular case study, be used online for no more than six months or until a significant operational change before retraining. In particular, however, as the drift evaluation of daily data only took about 0.1 second, it is strongly recommended an online monitoring scheme. Furthermore, it should be emphasized that the triggering of alarms depends on the adopted monitoring heuristics and user-defined hyperparameters, which directly influence sensitivity to performance degradation. Finally, for multiple-output models, such as the present case, retraining decisions might consider the relative importance of each output for the intended application. And, in addition to automated alarms, inspection of the time series of predictions and errors is recommended to support informed decision-making.

Future work may involve the generation of synthetic and controlled datasets derived from the available real-world data to allow hyperparameter calibration under predefined concept drift scenarios and to assess models trained with different data volumes using a fixed extrapolation set.

Acknowledgments: this work was supported by the Petrobras – Petróleo Brasileiro SA (SAP #4600675539), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (grant #311418/2023-6), Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) (grant #E-26/201.150/2022), and Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) (grant #001).

References

A. Bifet and R. Gavalda: Learning from Time-Changing Data with Adaptive Windowing, in Proceedings of the 2007 SIAM International Conference on Data Mining, 443–448, Society for Industrial and Applied Mathematics: 2007.

A. Hellestø, B. Djoric, G. Farah, S. E. Monge, E. Undheim e L. A. Ruden: How Shell is Tackling Hydrate/Scale Formation and Optimizing Well Allocation and Production on the Ormen Lange Subsea Field Development, in 38th International North Sea Flow Measurement Workshop, 2020.

D. E. Spina, L. F. de O. Campos, W. F. de Arruda, A. Melo, M. F. de S. Alves, G. L. Rabello, T. K. Anzai and J. C. Pinto: Comparison of Autoencoder Architectures for Fault Detection in Industrial Processes, Digital Chemical Engineering 12, 100162, Elsevier: 2024.

F. Pedregosa, G. Varoquaux, M. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay: Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

ISO: TR 11583—Measurement of Wet Gas Flow by Means of Pressure Differential Devices Inserted in Circular Cross-Section Conduits, ISO Switzerland: 2012.

J. Lu, A. Liu, F. Dong, F. Gu, J. Gama e G. Zhang: Learning under Concept Drift: A Review, IEEE Transactions on Knowledge and Data Engineering (31), 2346–2363, 2018.

J. Montiel, J. Read, A. Bifet and T. Abdessalem: River: machine learning for streaming data in Python, Journal of Machine Learning Research 22 (2021) 1–8.

M. F. de S. Alves, G. L. Rabello, B. C. Manhães, R. M. Soares, D. Q. F. de Menezes e J. C. Pinto: Using Machine Learning for Wet Gas Flow Metering in an Actual Production Site: Detailed Methodology and Case Study, in AIChE Annual Meeting 2025 Proceedings, American Institute of Chemical Engineers (AIChE): 2025a.

M. F. de S. Alves, G. L. Rabello, R. M. Soares, D. Q. de Menezes, L. O. V. Pereira e J. C. Pinto: On Machine Learning Approaches for Wet Gas Flow Metering with Actual Data: From Handling Available Data to Developing Models, in OTC Brasil 2025, Offshore Technology Conference: 2025b.

M. Niblett and I. Robertson: A Solution to Measure the Flow Rate of Wet Gas, Measurement and Control 44 (2011) 49–52.

R. Florence, L. Schwinn, K. Sprague, M. Coates e T. Markovich: When to Retrain a Machine Learning Model, arXiv preprint arXiv:2505.14903, 2025.

S. M. Salehi, L. Lao, L. Xing, N. Simms e W. Drahm: Devices and Methods for Wet Gas Flow Metering: A Comprehensive Review, Flow Measurement and Instrumentation 84 (2023) 102518.

T. Bikmukhametov and J. Jäschke: First Principles and Machine Learning Virtual Flow Metering: A Literature Review, Journal of Petroleum Science and Engineering 184 (2020) 106487.

T. M. T. Pham, K. Premkumar, M. Naili e J. Yang: Time to Retrain? Detecting Concept Drifts in Machine Learning Systems, in 2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), 260–271, IEEE: 2025.

Y. Ozaki, S. Watanabe and T. Yanase: OptunaHub: A Platform for Black-Box Optimization, arXiv preprint arXiv:2510.02798, 2025.