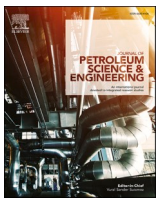
Journal of Petroleum Science and Engineering 218 (2022) 110982

Contents lists available at ScienceDirect

Journal of Petroleum Science and Engineering

journal homepage: www.elsevier.com/locate/petrol



Tagging and tracking oil-gas mixtures in multiphase pipelines

Riccardo Angelo Giro [a,*], Giancarlo Bernasconi [a], Giuseppe Giunta [b], Simone Cesari [b]

[a] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133, Milano, Italy
[b] Eni S.p.A., Facilities Technical Authority, San Donato Milanese, Italy

ARTICLE INFO

Keywords:
Multiphase propagation
Multiphase flow monitoring
Pipeline monitoring
Pipeline integrity
Machine learning

ABSTRACT

Pipeline transportation of multiphase products, such as gas and oil mixtures, exhibits complex and varying flow regimes: as a result, analytical approaches or conventional methods cannot accurately describe the composition or the propagation characteristics of the fluid mix inside the transportation system itself. We address such an issue by presenting a methodology, driven by the data and applied to a real case history, where basic pressure transients are used to tag and track, along a pipeline, different batches of a multiphase medium. Several statistical indicators, computed from the pressure data and on different window lengths, are employed to train a machine learning model, which learns to distinguish the characteristic behavior of two different oil-gas slugs: in practice, each different combination of fluid phases (in terms of gas/oil ratio in a given batch of product) and each different sequence of slugs (in terms of gas/oil ratio variability between successive batches) behaves like a coded tag linked to the flowing fluid. The key innovation consists in the possibility of tracking such multiphase slugs along the flowline and at each monitoring station: this allows one to determine in real-time the fluid composition entering/exiting the line, its position, and its movement along the pipe. As such, we obtain also a virtual metering system, able to provide estimates of the flow rate and phases ratio. Moreover, by having several recording stations accurately synchronized, one can also leverage real-time transmission and multichannel processing of the data, enabling the opportunity for online monitoring applications. The results on the test cases and the accuracy scores obtained for the metrics considered validate the tagging and tracking approach.

1. Introduction

Upstream hydrocarbons production and transportation must deal with multiphase flow, with a variable and often unpredictable ratio between gas and liquid phases. This ratio is fundamental in determining the flow regime along the pipe, as well as in tuning the appropriate fluid processing at the receiving terminal plant, in order to guarantee the efficiency and safety of the assets.

The need to model and predict the variability of multiphase propagation has been broadly addressed in the literature (Brennen, 2005; Henry et al., 1971; Hsu, 1972; Gudmundsson and Celius, 1999; Kumar et al., 2020). A fully theoretical approach, consisting in using mathematical equations and analytical models, cannot be employed in a real scenario, due to the impossibility of knowing the initial status and the governing equations with the required accuracy (Falcone and Alimonti, 2007; Agwu et al., 2022; Taylor, 1935). Other solutions make use of numerical simulations or experimental setups arranged in laboratories, which lack validation on real scenarios (Andrade et al., 2022; Chaves et al., 2022). The current frontier is instead represented by computational methods (Yan et al., 2018; Babakhani Dehkordi, Colombo, Guilizzoni and Sotgia, 2017), which employ models driven by the data, either available in the literature (Kanin et al., 2018; Alhashem, 2019; Al-Naser et al., 2016; Alhashem, 2020; Kanin et al., 2019; Gene et al., 2019), generated synthetically (Andrianov, 2018; Babanezhad et al., 2020) or collected from real transportation assets (Aziz AL-Qutami et al., 2018; Góes et al., 2021; Ye and Guo, 2013; Qiang et al., 2021).

What emerges from the research survey just discussed is that a lot of attention currently stands towards virtual flow metering systems (Bikmukhametov and Jäschke, 2020) or flow rate estimation in multiphase pipeline systems, yet (according to the best of the Authors' knowledge) the current literature is rather poor with regards to successfully tracking and monitoring the location of multiphase fluid batches/slugs: even though the latter is an equally important matter, it still remains a quite unexplored area of study. The work presented here fills such a research gap by proposing a data-driven methodology that allows to follow the position of diverse oil-gas mixtures moving along a pipeline at the

* Corresponding author.
E-mail address: riccardoangelo.giro@polimi.it (R.A. Giro).

https://doi.org/10.1016/j.petrol.2022.110982
Received 2 February 2022; Received in revised form 4 July 2022; Accepted 10 August 2022
Available online 20 August 2022

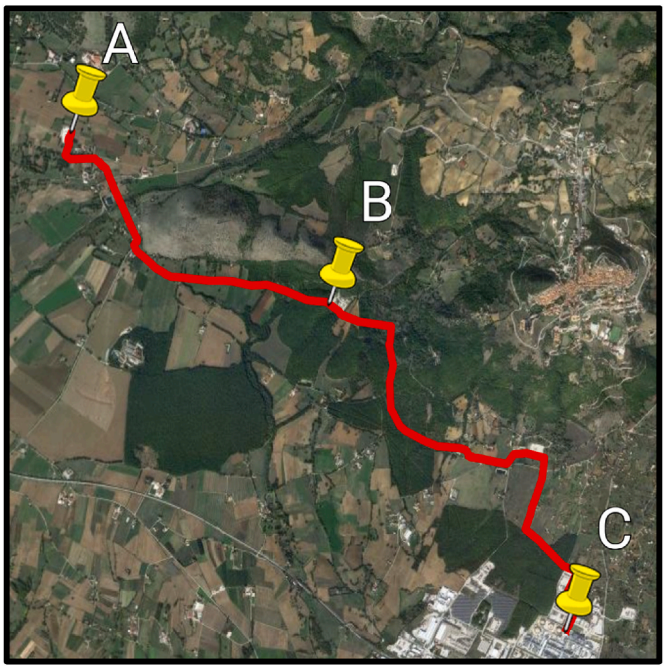


Fig. 1. Satellite map of the pipeline route (red line) and location of the sensing stations (yellow pins). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

velocity of the fluid flow. Our proposed method starts from a practical example to highlight its effective applicability on real scenarios: as such, we consider pressure signals collected by Eni for two full months (February 2020 and March 2020) in discrete locations along a 7.8 km oil-gas multiphase transportation asset located in Italy. We have designed and tested several features, directly derived from the collected pressure data, which are able to describe the variability of the multiphase batches flowing along the conduit. Such features are tuned for tracking a given sequence of mixtures along consecutive acquisition stations: the latter task is performed by a supervised learning model, based on Extremely Randomized Trees (Geurts et al., 2006). As said, the main technical novelty of our approach consists in an automated tagging and tracking process of the different multiphase slugs that enter/exit the pipeline, along with the opportunity to track their position and movement along the pipe. Besides that, our methodology is based on cheap and standard pressure measurements, which are typically already installed in many pipeline assets; it leverages on a simple machine learning algorithm; compared to most research works in the topic of oil/gas multiphase propagation, we have worked with an experimental dataset that is concurrently large enough (more than 15 billion examples) and collected from a real pipeline transportation system. Other key elements are an accurate synchronization of the measurements, the real time transmission and multichannel processing of the data.

The remainder of the paper is structured as follows. Section II provides an overview of the case study. Section III describes the processing steps applied to the data, whereas Section IV and Section V respectively explain how to tag and track oil-gas mixtures. Lastly, Section VI draws the conclusions.

2. Case study description

The work presented here makes use of pressure data collected for two months (from February 1st, 2020 to April 1st, 2020) by a proprietary vibroacoustic continuous monitoring system (e-vpms® technology (United States of America Patent No. US10401254B2, 2019)), installed on an upstream oil & gas network located in South Italy, which conveys a mixture of gas and oil products. The flowline has a length of approximately 7.8 km with altimetric variance, is characterized by 12" ID pipes and continuously operates with a service pressure comprised between 30 and 44 bar.

Table 1
Distance between each e-vpms® station and the pumping terminal positioned in station A.

Station	Distance with respect to station A (km)
B	3.6
C	7.8

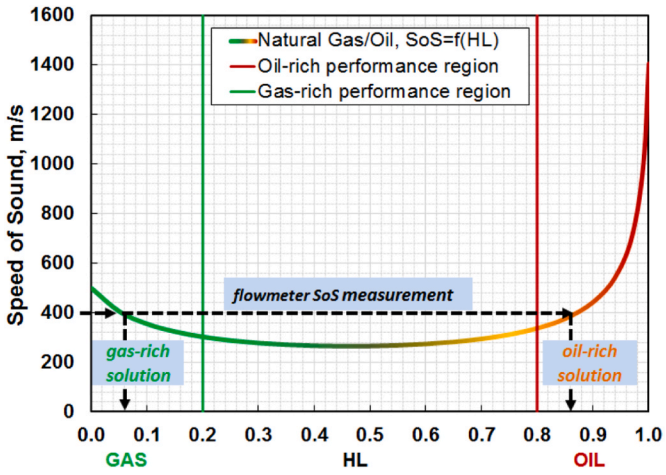


Fig. 2. Sound speed in a gas-oil mixture (Unalmis, 2015).

The gas-to-oil ratio (GOR) typically ranges between 50 and 250, with an average value of approximately 200, based on separator measurements collected on-site. Fig. 1 displays the satellite track of the conduit (red curve) and the location of the three acquisition stations installed on the line itself (highlighted by yellow pins and respectively labelled as A, B and C). The production well (A) is upstream with respect to the Oil Centre (C). Each e-vpms® sensing unit is equipped with a dynamic hydrophone, which records pressure transients within the multiphase fluid. The sampling rate has been set to 1 kHz: recalling that the acquisition time spans over two full months (e.g., 60 days), the total number of recorded samples per sensor is more than 5 billion. Table 1 reports the relative distance between each acquisition unit and the production terminal installed at station A.

In a pipeline fluid transportation system, every interaction with the pipe or with the flow generates acoustic signals that propagate as guided waves within the fluid itself. Considering multiphase pipelines, a sensor in contact with the fluid (e.g., an hydrophone) can measure pressure transients (i.e., sounds) having different characteristics according to the variable gas/liquid ratio of the moving mixture: during a normal operation regime, the related pressure fluctuations are neither constant nor periodic, yet their statistical features can be analyzed and exploited, using data-driven techniques, to define one or more propagation regimes.

Given such a context, two distinct acoustic propagation processes have been observed. The first one is related to pressure transients generated by the flow regulation equipment (e.g., pumps, valves, etc.): these transients propagate within the pipe at the sound velocity v_s depending on the type of fluid transported. More specifically, if the pipe was entirely filled with gas, one would observe a propagation velocity of a few hundred m/s; if one instead conveys liquid oil, v_s would be around 1000 m/s. Considering oil-gas mixtures, the sound speed exhibits the behavior shown in Fig. 2 (Unalmis, 2015): it is clear that sound velocity measurements can be exploited to estimate the gas-oil ratio. In principle, v_s is obtainable by a correlation analysis between the pressure transients recorded at different stations, as shown in (Bernasconi and Giunta,

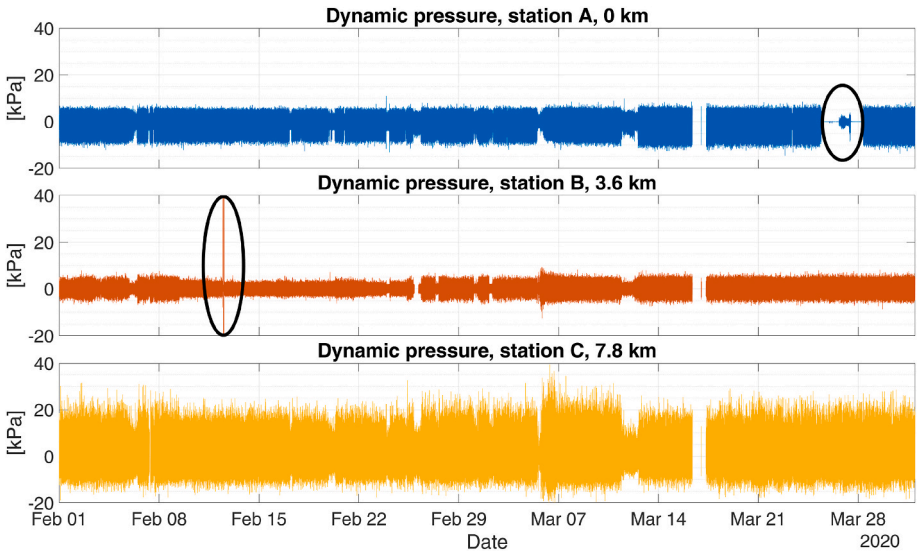


Fig. 3. From top to bottom: raw dynamic pressure measurements, respectively collected at stations A, B and C. The black ellipses highlight outliers and sensor errors.

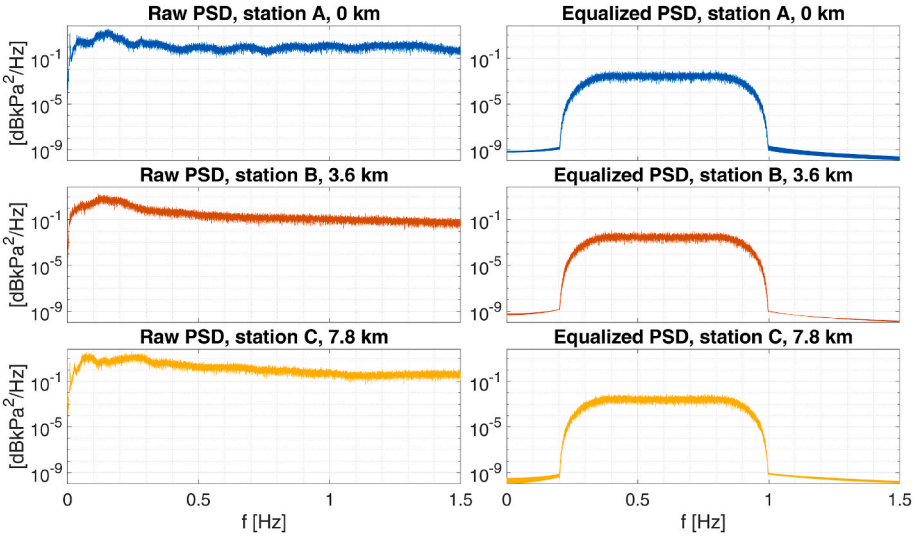


Fig. 4. Raw power spectral density of the dynamic pressure signals, respectively collected at stations A, B and C (left column, from top to bottom), and corresponding PSD after bandpass filtering and equalization (right column).

2020): a correlation peak happens at the relative propagation delay of the considered stations. Unfortunately, in high GOR regimes (like in our case, the average GOR is around 200) the propagation is very chaotic and turbulent due to the dominant presence of gas: as a consequence, pressure transients tend to decorrelate very rapidly (after a few hundred m) such that the aforementioned technique cannot be employed to estimate v_s. We have in fact verified this aspect on our experimental datasets by correlating the signals collected at the two closest stations (e.g., A and B), obtaining unsatisfactory outcomes.

The second propagation process is instead related to the fluid flowing within the line, which travels at the flow velocity v_f of some m/s: for instance, a batch of oil-gas mixture travelling at such a speed would be observed at two distinct sensing stations (e.g., A and C) with a time difference of several minutes or hours. In this case, the ability of tracking a given batch along the line permits to measure the flow velocity. The following Sections present an automatic procedure, fed with pressure data recorded in stations A, B and C along the pipeline, able to tag and track the batches of produced fluid travelling along the line.

3. Data processing

As a first step, raw pressure measurements (displayed in Fig. 3) undergo a number of preliminary operations to make them suitable for a machine learning workflow. For this specific application, we have performed the following tasks:

1) Removal of outliers and sensor errors (Giro et al., 2021a, 2021b; Giunta et al., 2020). More precisely, dynamic pressure measurements having absolute value greater than 100 kPa have been ruled out from the dataset. In addition, all the time intervals in which a sensor was not operational have been manually detected and the corresponding data points have been set to a null value. All those instances (highlighted in Fig. 3 with black ellipses) are related to sporadic electromagnetic disturbances affecting the power unit of the equipment;
2) Bandpass filtering pressure data between 0.2 and 1 Hz. This operation has a dual purpose: firstly, to remove the zero-frequency component, which would introduce an undesired bias in the

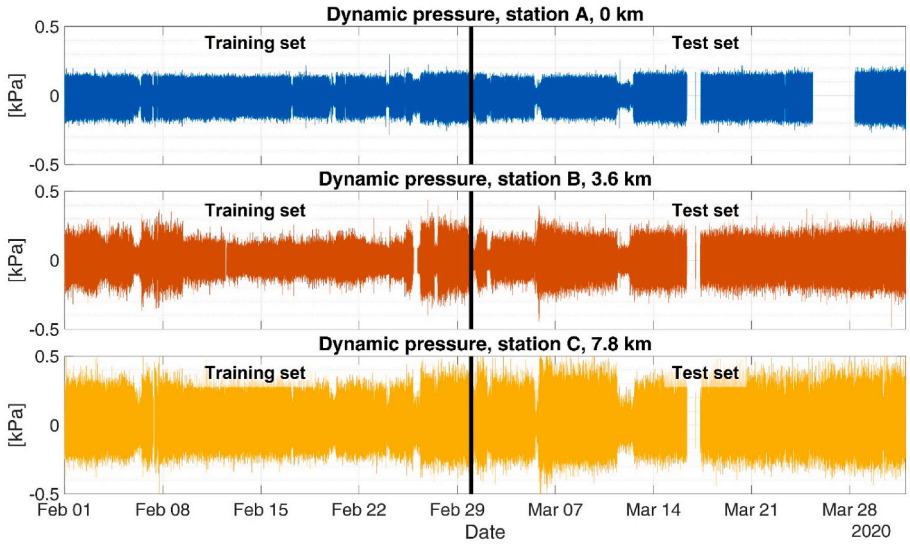


Fig. 5. From top to bottom: processed dynamic pressure measurements, respectively collected at stations A, B and C: the training and test regions are separated by a black vertical bar.

measurements; lastly, to keep the useful part of the observed signal (which is typically below 1 Hz) and to reduce the noise bandwidth. The useful bandwidth of the signal has been empirically derived by performing numerous correlations of the original signals within different frequency sub bands, and we concluded that the frequency range (0.2 Hz, 1 Hz) is the optimal one from a signal-to-noise ratio (SNR) point of view. We also recall that the attenuation of sounds increases with frequency, and for this reason it makes more sense to keep the low frequency component of the data, rather than its high-frequency counterpart: in this specific case, the SNR above 1 Hz is degraded to a point that it's not worth considering the corresponding frequency components;

3) Equalization. This operation is the reversal of distortion which occurs whenever a signal is transmitted through a channel: more specifically, equalizers are used to flatten the frequency response of such a signal within a desired frequency range; this means that the propagation channel (e.g., the pipe) progressively introduces distortion and degrades the quality of the target signal to be measured. In practice, if we were to use the raw pressure data, we would not be able to successfully track the multiphase slugs that propagate along the pipeline (the signal would get lost after a certain distance): for these reasons, equalization is a key step within the data processing phase. In this particular case, the pressure signals collected at the three different stations are each affected by local noise which alters their spectral content within the bandwidth of interest (0.2–1 Hz) in an undesired manner: this aspect can be observed by looking at the power spectral density (PSD) of the raw pressure data (Fig. 4, plots on the left column). Such an inhomogeneity is also visible in the time series (Fig. 3), which are characterized by non-uniform scales among each other: this aspect can become problematic during the training and testing phases of a machine learning algorithm (Danushka, 2017). To overcome this issue, we have whitened each PSD between 0.2 and 1 Hz using an equalizer based on Welch's method (Welch, 1967). The result of such an operation is displayed in Fig. 4 (plots on the right column).

In addition, it should also be noted that there are missing data points in the pressure time series represented in Fig. 3 (e.g., between March

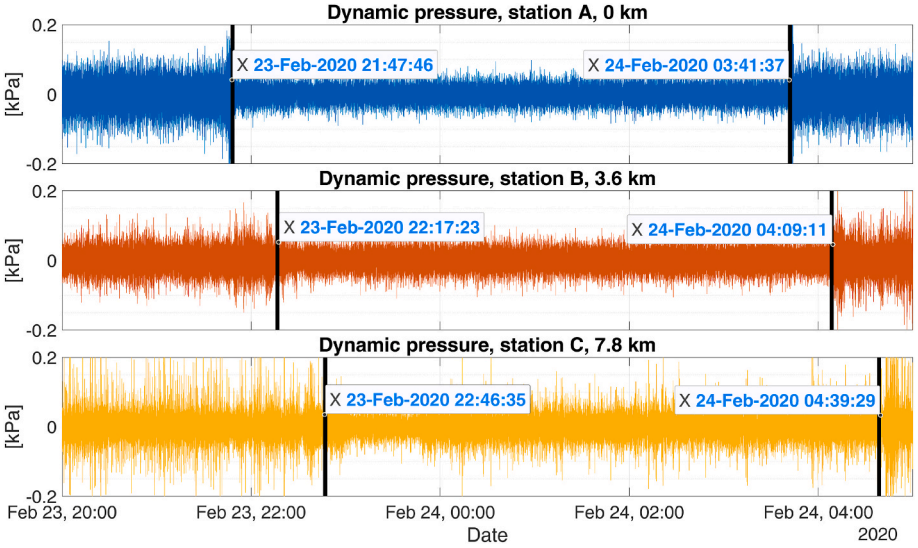


Fig. 6. From top to bottom: propagation of the first type (outside the black vertical bars) and of the second type of oil-gas slug (enclosed by black vertical bars) along the three recording stations A, B and C.

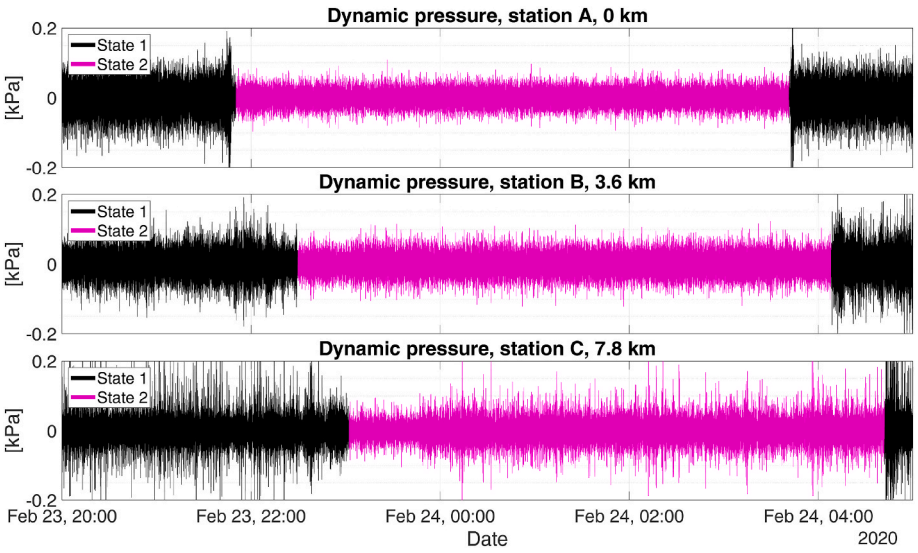


Fig. 7. From top to bottom: labelled pressure data for the three stations A, B and C.

14th and March 21st): the latter are related to the instances in which the acquisition units were temporarily not operational. Lastly, Fig. 5 displays the data collected by the dynamic hydrophones at their respective stations after the processing operations previously described. The historical data devoted to train and test the machine learning model are highlighted in Fig. 5 using text boxes and a black vertical line.

4. Tagging oil-gas mixtures

By looking at the time series in Fig. 5, two distinct multiphase regimes can be detected: the former and most frequent one (Fig. 6, outside two vertical bars) can be distinguished from the latter (Fig. 6, enclosed by black vertical bars) as the recorded pressure signals present a different acoustic energy content. We can qualitatively label such regimes as "high energy" and "low energy". It is also worth noting (by looking at the example in Fig. 6) that each transition from one state to another is observed at the three sensing stations with a relative delay τ of approximately 30 min between each other. Taking into account the relative distances reported in Table 1, it is reasonable to assume that these changes are due to variations in the composition of the flowing product: based on the on-site production logs, it is in fact expected to propagate on average with a velocity v_f of approximately 2 m/s.

The two different batches of oil-gas mixtures present therefore unique acoustic signatures, whose characteristics can be directly extracted from the recorded data: for instance, higher order statistical moments of pressure measurements (such as variance) are good candidates to characterize the flow regime along the entire pipeline. We have therefore exploited this observational knowledge to manually tag each of the time series displayed in Fig. 5 with a binary label, which univocally identifies one of the two oil-gas batches. An example of such an operation is depicted in Fig. 7, where every sample of the signals displayed in Fig. 6 has been assigned a tag: the datapoints corresponding to the high energy state (e.g., state 1) have been colored in black, while the remaining ones (state 2) have been represented in magenta.

4.1. On the need for supervision

Whenever this category of problems is tackled using a data-driven approach, a common argument might question the need for employing machine learning in the first place: since we have just said that the variance of the signal can be a valid indicator to distinguish between the two propagation regimes, one could simply evaluate the aforementioned

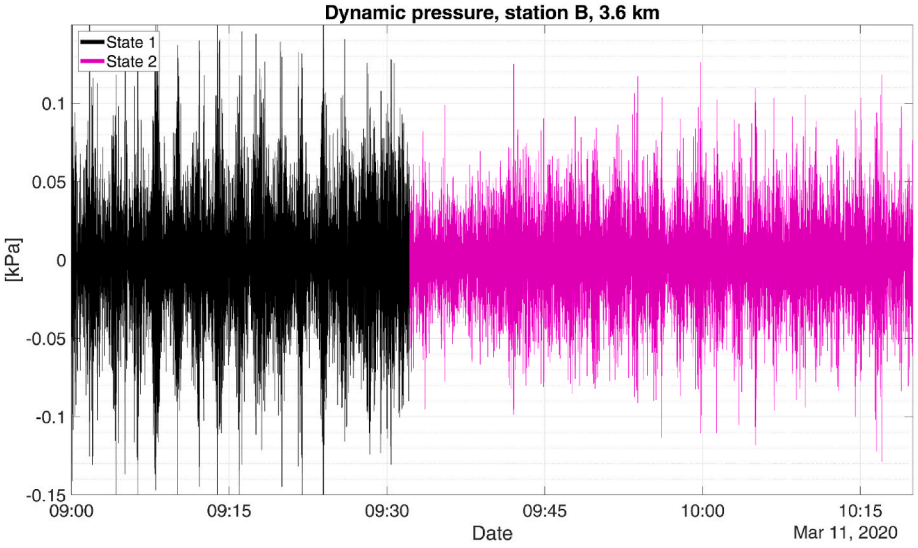


Fig. 8. Change of state where the boundary between the two multiphase regimes is not straightforward.

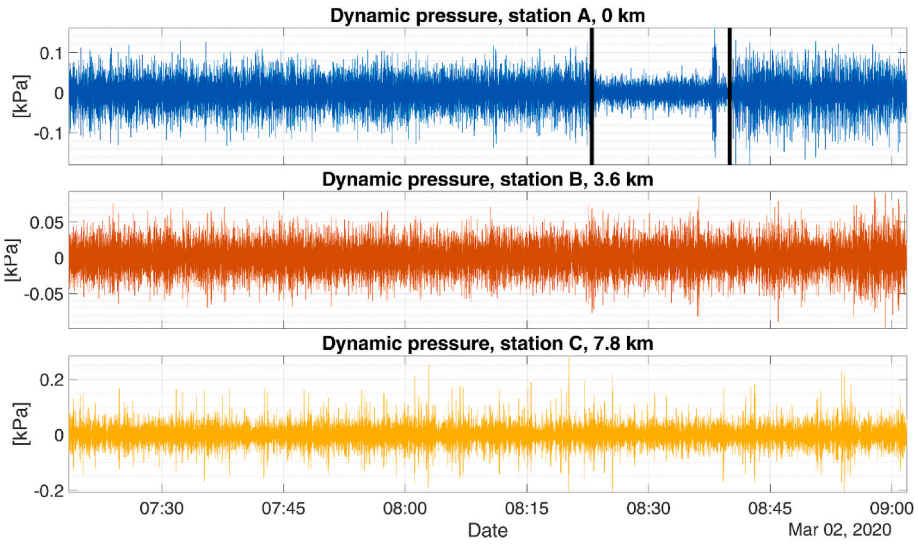


Fig. 9. Example of acoustic event (originated at station A and enclosed by black vertical bars) that is not visible at stations B and C.

quantity over two different windows, and the detection of a large variation in the data can be set as the discriminant between state 1 and state 2. However, deterministic methods (such as this one) tend to be less reliable the more uncertain the decision threshold becomes (what is exactly a large enough variation in the data to be eligible for tagging?): this would either result in classification ambiguity or in the requirement for human supervision. Let us consider the example displayed in Fig. 8: in this case, it is not trivial to decide for state 1 or state 2 using a simple moving variance, because it is not clear where to put the threshold on the variance level. The machine learning method presented in Section V overcomes this limitation: it will be demonstrated that its capability to generalize well enough can make the data-driven model more likely to be deployed in other assets. A moving variance-based solution, instead, would certainly require specific calibration for each new scenario.

Another problem that cannot be solved by basic techniques regards the presence of acoustic events that do not propagate with the same velocity of the flow: these processes still generate significant amplitude variations in the measured signals, yet they do not have any relation with multiphase batches travelling through the pipe. In Section II, we have also stated that such pressure transients decorrelate very rapidly after a few hundred m and are not visible at successive stations: as a result, a basic technique that blindly looks at the moving variance would trigger a false event at a given station, and would remain silent for all the others. For all the reasons just described, it is therefore of paramount importance to correctly tag the data and feed them to an expert system: the aim is to automatically detect and track only coherent events, ruling out all the false alarms. Deterministic techniques, instead, cannot simply provide the required level of abstraction typical of machine learning algorithms.

Fig. 9 shows an example instance of the problem just discussed: an acoustic event (enclosed by black vertical bars) occurs at station A on March 2nd, 2020 before 8:30 in the morning, yet it is not visible at any of the successive stations; this event was caused by the flow regulation equipment (located at station A) and was not related to changes in the gas/oil fraction. It is clear that blindly looking at the amplitude of the signal would erroneously highlight a variation in the multiphase composition: a machine learning model can instead learn to detect the propagation phenomena of interest (e.g., the ones related to multiphase propagation), while concurrently discarding false events (such as the one depicted in Fig. 9) if trained appropriately.

5. Tracking oil-gas mixtures

By having labelled data at disposal, one can employ such tags to train a supervised learning classifier, having two practical applications in mind: firstly, the real time tracking of the position of different oil-gas batches moving along the conduit; lastly, the automatic identification of the type of mixture currently entering/exiting the line. To perform these operations, we have employed a multi-output Extremely Randomized Trees Classifier (ERTC) (Geurts et al., 2006): the latter is a data-driven algorithm which provides (at each sampling instant) three discrete outputs by analyzing several characteristics of the input signal. For this specific application, the classifier has been designed to provide binary labels, which respectively identify (for each of the three stations) one of the two flow regimes previously discussed. We have chosen to employ the ERTC because it can concurrently satisfy the following requirements: it must be reliable, accurate and robust; it must have the simplest implementation possible; it must be based on basic and easy to understand principles. Moreover, a properly trained ERTC can choose autonomously which features are more appropriate for a particular

Table 2
Summary of the raw features, evaluated from the processed pressure transients dataset.

Moving statistics	Computed at	Causal time window (minutes)											
		5	15	30	45	60	75	90	105	120	180	240	300
Variance	Station A	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
Minimum		x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}
Maximum		x_{25}	x_{26}	x_{27}	x_{28}	x_{29}	x_{30}	x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}
Variance	Station B	x_{37}	x_{38}	x_{39}	x_{40}	x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}
Minimum		x_{49}	x_{50}	x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}	x_{57}	x_{58}	x_{59}	x_{60}
Maximum		x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}	x_{67}	x_{68}	x_{69}	x_{70}	x_{71}	x_{72}
Variance	Station C	x_{73}	x_{74}	x_{75}	x_{76}	x_{77}	x_{78}	x_{79}	x_{80}	x_{81}	x_{82}	x_{83}	x_{84}
Minimum		x_{85}	x_{86}	x_{87}	x_{88}	x_{89}	x_{90}	x_{91}	x_{92}	x_{93}	x_{94}	x_{95}	x_{96}
Maximum		x_{97}	x_{98}	x_{99}	x_{100}	x_{101}	x_{102}	x_{103}	x_{104}	x_{105}	x_{106}	x_{107}	x_{108}

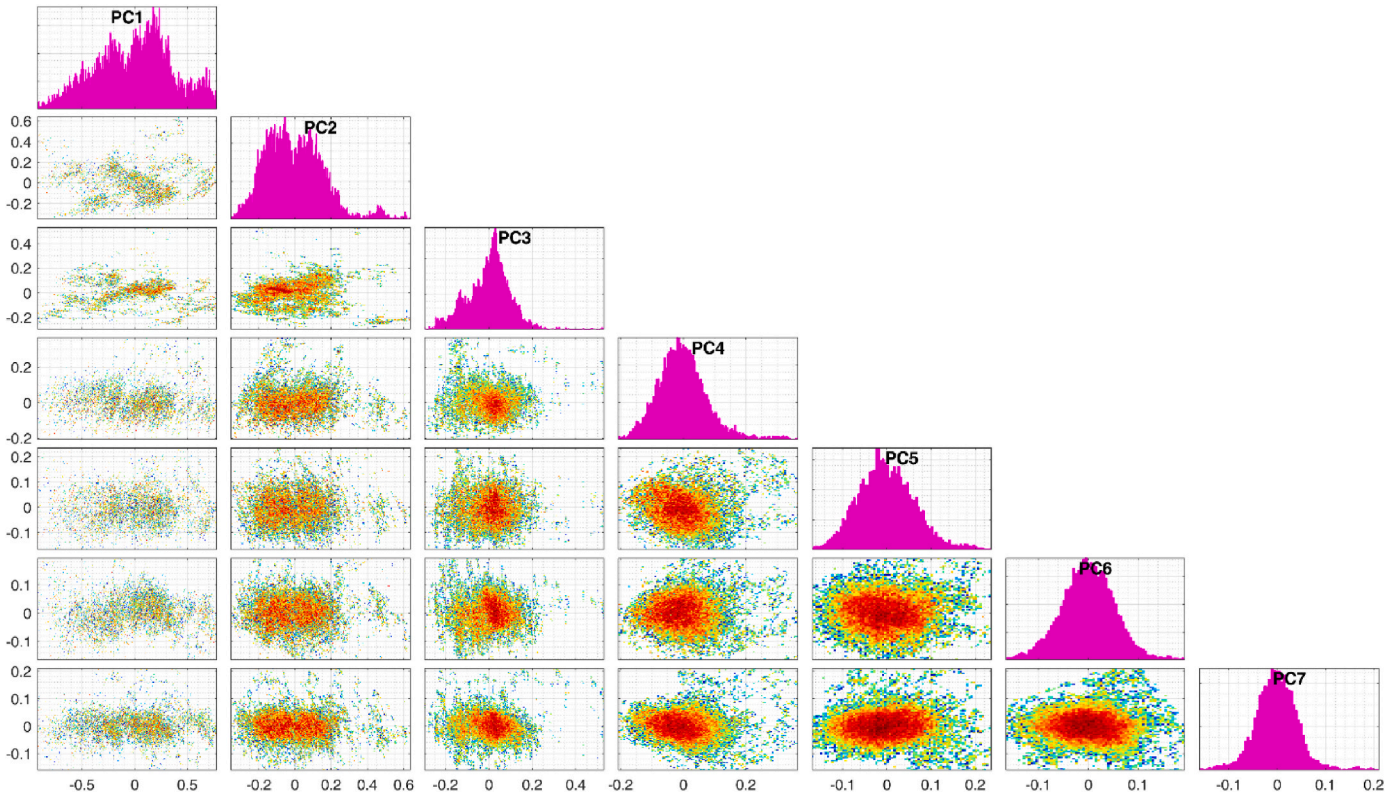


Fig. 10. Principal component analysis, performed on the experimental data.

slug/state combination: this feature importance evaluation process is a peculiar property of the algorithm itself, and makes it very reliable from a performance point of view.

Besides a $m \times 3$ matrix of ground truth labels y (i.e., the m manually tagged examples times 3 measurement stations), the classifier requires as additional input a $m \times N$ matrix of features X, where the N entries of each row are obtained by evaluating several statistical indicators from the dynamic pressure signals: more specifically, we have computed the variance, minimum and maximum over 12 causal rolling windows, ranging from 5 min up to 5 h; therefore, each training example x_k is described by a $1 \times N$ row vector, where $N = 108$. Table 2 summarizes the 108 raw features that have been evaluated from the processed pressure data. The procedure is described as follows: at each time step k, one has to evaluate x_i, where $i \in [1, \ldots, 108]$. For example, x_1 corresponds to the variance of the pressure transients recorded at station A over the previous 5 min (e.g., an interval between $k - 5$ min and k), x_2 is the same statistics but computed over the past 15 min, and so on.

The aforementioned feature engineering process follows this logic: if we had to manually look at the pressure measurements and decide between the two classes "state 1" and "state 2", we would look at the current shape of the signal (e.g., visually assessing its variance, minimum and maximum values) and compare it with its past history: the definition of "past history" varies widely, since a certain type of slug (or, equivalently, GOR) can last from several minutes up to a few hours (based on our experimental data). We have therefore carefully designed the ERTC with this logic in mind: we wanted the algorithm to react like we typically do and to automatically perform a task that would otherwise be done by hand, which ultimately takes a lot of time and experience to be executed correctly.

Working in high-dimensional spaces (e.g., having a lot of features) can potentially be undesirable for many reasons: in particular, training a machine learning algorithm using too many features is usually computationally cumbersome and does not guarantee a better performance of the algorithm itself. Given the high cardinality of our feature space ($N > 100$), we have therefore performed dimensionality reduction using principal component analysis (PCA) (Jolliffe, 2005) to find a smaller feature set X' having size $m \times N'$ (where $N' < N$), while still preserving most of the variance from the original data. In practice, the raw feature matrix X undergoes an orthogonal linear transformation which maps the data into a new coordinate system having N' components: we have therefore applied PCA on X with the goal of preserving at least 90% of the original variance of the data. By accepting a tradeoff with a small loss of information (e.g., 10% variance reduction), a substantial dimensionality reduction can be obtained: in fact, the minimum value of N' that satisfies this condition is $N' = 7$, thus achieving a compression factor greater than 15. We also remind that PCA does not eliminate any of the initial features per se, it simply combines them to create a new set of features, smaller than the former one. Fig. 10 represents the outcome of PCA on our experimental data: as with many real world problems, the boundaries between the two target classes (e.g., state 1 and state 2) are not always clear: this is one of the reasons why basic, deterministic solutions are not good enough to correctly tag and track multiphase mixtures. Still, some dense and distinct regions in the PCA domain can be detected (e.g., the density plots below the magenta histograms entitled "PC1" and "PC2"): this confirms that our subset of 7 transformed features are useful indicators for distinguishing the two multiphase regimes of interest.

The multi-output ERTC has been trained for a full calendar month (February 2020) using examples collected from all three stations. Training a data-driven model using this type of approach becomes advantageous, since we can embed the notions of spatial and temporal memory into the ERTC, which are respectively provided by the multi-output design and by computing the feature matrix using only moving, causal statistics. Testing has instead been performed on the remaining time frame (March 2020): given the abundance of available examples (>5 billion), an equal subdivision between the two sets proves to be sufficient, as it guarantees enough samples for training the algorithm, allows for a thorough testing phase and makes it easier to detect

Table 3
Performance of the ERTC on the three test sets.

Metrics	Station A, 0 km		Station B, 3.6 km		Station C, 7.8 km	
	State 1	State 2	State 1	State 2	State 1	State 2
P	97.69%	98.66%	99.18%	98.74%	99.39%	98.70%
R	99.69%	90.54%	99.82%	94.56%	99.76%	96.76%
F_1 (per class)	98.68%	94.43%	99.49%	96.60%	99.57%	97.72%
F_1 (overall)	97.87%		99.12%		99.28%	

any potential issues with bias or overfitting the data to the training set. Moreover, performance can be either assessed on the overall test set or it can be further refined by treating the three measurement stations as independent of each other: the latter approach permits to detect in which locations the classifier provides the highest or the lowest accuracy. Lastly, by having multi-point outputs at disposal, one can also infer the mean composition of the multiphase batch along the entire pipeline and/or individual segments: the overall GOR at a specific time instant is obtained by averaging the output tags, weighted by the length of each line section.

To assess the performance of the proposed mixture tracking model, we have considered the typical metrics employed in statistical analysis of binary classification, namely: precision P, recall R and F_1 score. Those quantities are a function of the number of true positives T_p, false positives F_p and false negatives F_n, and are respectively defined as follows:

$$P = \frac{T_p}{T_p + F_p}, \tag{1}$$

$$R = \frac{T_p}{T_p + F_n}, \tag{2}$$

$$F_1 = 2\frac{PR}{P + R}. \tag{3}$$

Table 3 reports the values of P, R and F_1, respectively attained for each binary class and for the three test sets considered (data collected from stations A, B and C). The results are quite satisfactory, as the overall accuracy level (Table 3, lowermost row) is greater than 97% for every test set considered. Such a high degree of robustness can also be observed by comparing the time series of the predicted tags with the ground truth, reference values. An example is displayed in Fig. 11: the outputs of the ERTC (Fig. 11, bottom row) closely match the original labels (Fig. 11, top row).

6. Conclusion

This paper presented a novel machine learning strategy which has been applied to historical pressure data for tracking sequences of oil-gas multiphase mixtures, conveyed along a pipeline in an upstream scenario. The proposed work demonstrates that the position of such mixtures can be successfully monitored, since each slug presents a characteristic propagation signature which is strongly dependent on the composition of the mixture itself: this allows to precisely describe the high variability in the operational statuses of the pipeline. Having a mixture of fluids in a pipeline brings many challenges due to the individual characteristics of each phase: for example, reactions between different components can cause corrosion of the pipeline, which leads to leaks and ruptures; moreover, these could potentially go unnoticed if occurring in a subsea pipeline, where it is not possible to install physical meters. Another critical issue consists in appropriately preparing the receiving terminal to optimally process the incoming multiphase mixture, which is a key step in ensuring the safety of operations: given such a context, knowing its real-time composition is an important requirement; our tracking system can provide such an information, and additionally allows for virtual flow rate metering along the entire conduit. In fact, the model has been designed and validated on experimental data, collected for two months from a real pipeline asset. Results obtained so far confirm the capability of tracking different oil-gas mixtures at an intermediate point and at the two ends of a conduit, achieving an overall accuracy level greater than 97% for all the test sets considered. Future work will be focused on testing the data-driven model presented here in other oil & gas pipeline networks.

Credit author statement

Riccardo Angelo Giro: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Giancarlo Bernasconi**: Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **Giuseppe Giunta**: Conceptualization, Methodology, Investigation, Resources, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **Simone Cesari**: Investigation, Resources, Writing – original draft, Writing – review & editing, Visualization.

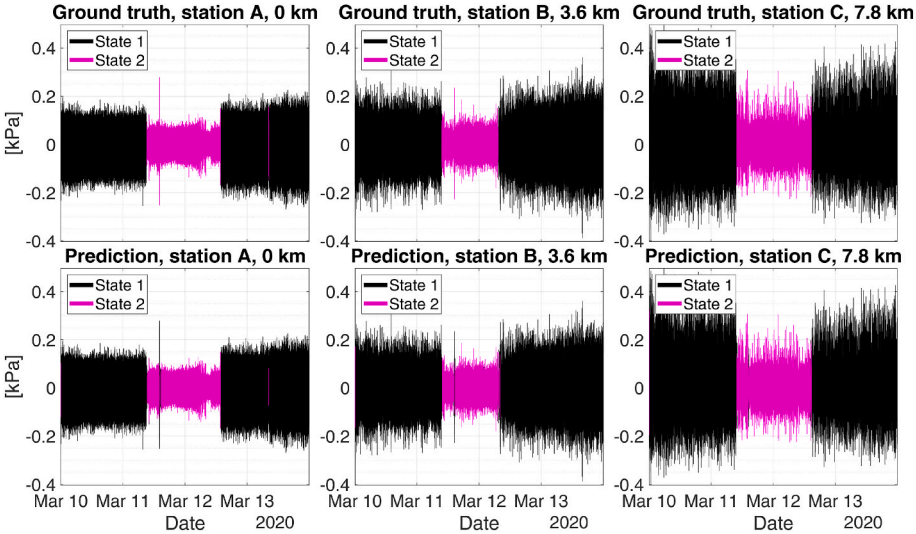


Fig. 11. Top row, from left to right: ground truth labels on dynamic pressure signals at stations A, B and C, respectively. Bottom row, from left to right: predicted tags at stations A, B and C, respectively.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was mainly carried out in the framework of the R&D – SIMON project funded by Eni S.p.A. The authors are grateful to Eni SpA Distretto Meridionale (DIME) and SolAres JV teams for technical support during the field tests.

Appendix: Extremely Randomized Trees Classifier

An Extremely Randomized Trees Classifier (ERTC) is an ensemble machine learning algorithm for classification, based on decision trees (DTs). To better understand how an ERTC operates, it is fundamental to outline how a standard DT works.

Decision Trees

Decision Trees (Quinlan, 1986) are a non-parametric and supervised learning method that can be either employed for classification and/or regression: in practice, a DT is a very basic and flexible model that (if properly designed) maintains a low degree of complexity. When building a DT, the goal is to devise a model which predicts the value of a target variable by learning simple decision rules (such as a sequence of IF/ELSE statements), which are inferred from an input set of features.

Fig. 1 displays the general architecture of a DT. The latter is composed of two core elements: nodes and branches. The root node is the starting point of every DT; decision nodes represent all the intermediate levels, while the leaf nodes correspond to the terminations of the DT. A branch is instead a subsection of the entire tree. When training a DT, all the input features are evaluated at each node (leaves excluded) to redirect every training example into one of the many terminations. The tree is gradually built by recursively evaluating different features at the various nodes and by finding the one that provides the best split of the data at each node. The criteria for deciding the optimal way of splitting the input examples can vary: for classification purposes, the most common ones are the Gini Impurity or the Shannon entropy. To provide ease of understanding, suffice it to say that these optimization metrics can be interpreted as a cost function that has to be minimized, and choosing the best features at each node will result in a reduction of such a cost.

Fig. 2 shows an example application of a DT on the renowned Iris dataset (Fisher, 1936) to solve a simple classification problem. The dataset consists of 50 samples from three different species of Iris flowers (e.g., Iris setosa, Iris virginica, and Iris versicolor), and each example is characterized by 5 attributes (e.g., 4 features and 1 target class label): sepal length, sepal width, petal length, petal width and species. The root node is initially fed with the entire sample size (samples = 150), and in this case the feature that provides the best split for that node is the petal width: 50 flowers having petal width less or equal than 0.8 cm are declared to belong to a specific class (e.g., Iris setosa); otherwise, one descends to an intermediate decision node and searches once again for a feature that provides the best split for the remaining 100 examples. In this particular example, the petal width is still the most informative attribute among all: this time, the new decision threshold is set at 1.75 cm. Lastly, the remaining 100 data points are split into two leaf nodes (54 on the leftmost one and 46 on the rightmost). It is worth noting that the number of branches in a tree does not necessarily equal the number of input features, and that certain attributes can be the most informative at multiple nodes, whereas others can remain unused for the entire decision process. Still, one should remember that all the features are being evaluated at each node, however only the one returning the lowest Gini Impurity index is chosen as the decisive one. This brings an important consequence to be highlighted: a DT develops the capability of automatically and autonomously choosing which features are the most appropriate for taking each decision, and which are not.

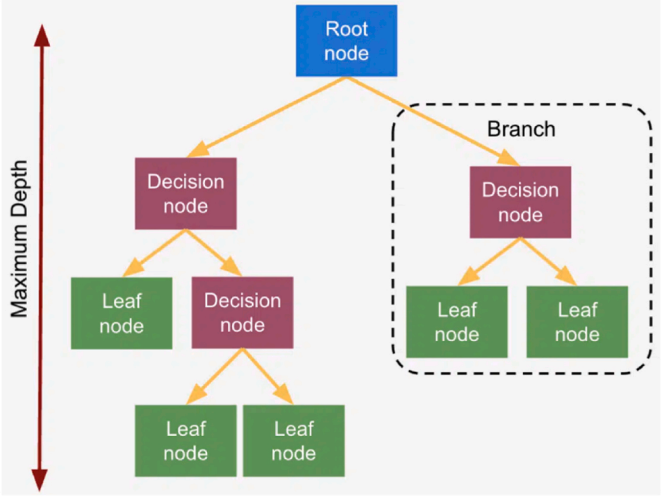


Fig. 1. General architecture of a DT (source).

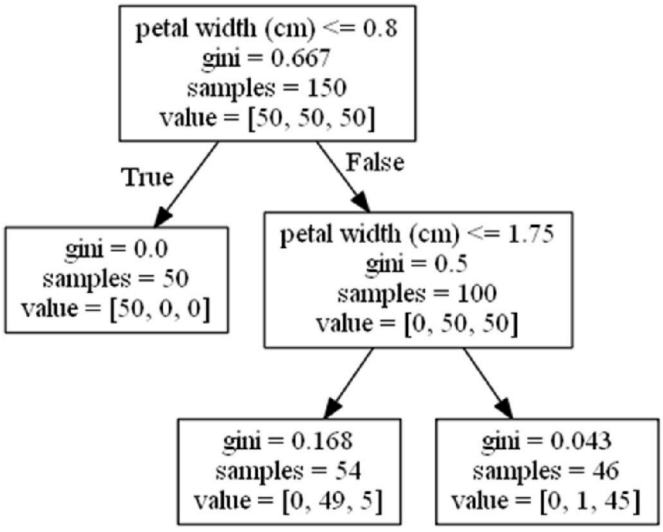


Fig. 2. Application of a DT on the Iris dataset (source).

Extremely Randomized Trees

Standard decision trees suffer from several drawbacks that make them often unviable:

- At times, one ends up with overcomplicated trees (e.g., having too many branches and leaf nodes) which do not generalize the data well: this phenomenon is also known as overfitting;
- The risk of overfitting is increased further if a DT is fed with hundreds of features;
- DTs can easily become unstable because very small variations in the input data might result in the generation of a completely different tree;
- DTs are uncapable of solving tasks that surpass certain complexity levels, such as modeling a sequence of XOR operations or learning the behavior of a multiplexer.

Extremely Randomized Trees (ERTs) overcome all these issues by creating a large forest of many different DTs, each built randomly from the training data. Predictions are obtained by averaging each DT outcome (in case of regression) or by using a majority voting system (in case of classification): for an ERTC, the predictions of many decision trees are therefore considered together, and the predicted class collecting the highest number of votes is declared as the winner for a given input instance. For example, let us assume that an ERTC has been built using a forest of 200 DTs to classify each flower of the Iris dataset. For a given input data point, the entire forest might output the following:

- 146 trees predict Iris setosa;
- 15 trees predict Iris virginica;
- 39 trees predict Iris versicolor.

In this specific case, the input example is classified as Iris setosa, since the majority of the DTs within the random forest voted for that target class.

References

Agwu, O.E., Okoro, E.E., Sanni, S.E., 2022. Modelling oil and gas flow rate through chokes: a critical review of extant models. J. Petrol. Sci. Eng. 208, 109775 https://doi.org/10.1016/j.petrol.2021.109775.

Alhashem, M., 2019. Supervised machine learning in predicting multiphase flow regimes in horizontal pipes. In: Abu Dhabi International Petroleum Exhibition & Conference. Abu Dhabi. https://doi.org/10.2118/197545-MS.

Alhashem, M., 2020. Machine learning classification model for multiphase flow regimes in horizontal pipes. In: IPTC International Petroleum Technology Conference. Dhahran. https://doi.org/10.2523/IPTC-20058-Abstract.

Al-Naser, M., Elshafei, M., Al-Sarkhi, A., 2016. Artificial neural network application for multiphase flow patterns detection: a new approach. J. Petrol. Sci. Eng. 145, 548–564. https://doi.org/10.1016/j.petrol.2016.06.029.

Andrade, G.M., de Menezes, D.Q., Soares, R.M., Lemos, T.S., Teixeira, A.F., Ribeiro, L.D., Pinto, J.C., 2022. Virtual flow metering of production flow rates of individual wells in oil and gas platforms through data reconciliation. J. Petrol. Sci. Eng. 208, 109772 https://doi.org/10.1016/j.petrol.2021.109772.

Andrianov, N., 2018. A machine learning approach for virtual flow metering and forecasting. In: 3rd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production OOGP 2018. https://doi.org/10.1016/j.ifacol.2018.06.376.

Aziz AL-Qutami, T., Ibrahim, R., Ismail, I., Azmin Ishak, M., 2018. Virtual multiphase flow metering using diverse neural network ensemble and adaptive simulated annealing. Expert Syst. Appl. 72–85. https://doi.org/10.1016/j.eswa.2017.10.014.

Babakhani Dehkordi, P., Colombo, L.P., Guilizzoni, M., Sotgia, G., 2017. CFD simulation with experimental validation of oil-water core-annular flows through Venturi and Nozzle flow meters. J. Petrol. Sci. Eng. 149, 540–552. https://doi.org/10.1016/j.petrol.2016.10.058.

Babanezhad, M., Taghvaie Nakhjiri, A., Rezakazemi, M., Marjani, A., Shirazian, S., 2020. Functional input and membership characteristics in the accuracy of machine learning approach for estimation of multiphase flow. Sci. Rep. 10 (1) https://doi.org/10.1038/s41598-020-74858-.

Bernasconi, G., Giunta, G., 2020. Acoustic detection and tracking of a pipeline inspection gauge. J. Petrol. Sci. Eng. 194, 107549 https://doi.org/10.1016/j.petrol.2020.107549.

Bikmukhametov, T., Jäschke, J., 2020. First principles and machine learning virtual flow metering: a literature review. J. Petrol. Sci. Eng. 184, 106487 https://doi.org/10.1016/j.petrol.2019.106487.

Brennen, C.E., 2005. Fundamentals of Multiphase Flow. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511807169.

Chaves, G.S., Karami, H., Ferreira Filho, V.J., Vieira, B.F., 2022. A comparative study on the performance of multiphase flow models against offshore field production data. J. Petrol. Sci. Eng. 208, 109762 https://doi.org/10.1016/j.petrol.2021.109762.

Danushka, B., 2017. Dynamic feature scaling for online learning of binary classifiers. Knowl. Base Syst. 97–105. https://doi.org/10.1016/j.knosys.2017.05.010.

Falcone, G., Alimonti, C., 2007. The challenges of multiphase flow metering: today and beyond. In: ASME 2007 26th International Conference on Offshore Mechanics and Arctic Engineering. San Diego. https://doi.org/10.1115/OMAE2007-29527.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugenics 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x.

Gene, M., Xingru, W., Kegang, L., 2019. An improved model for gas-liquid flow pattern prediction based on machine learning. J. Petrol. Sci. Eng. 183, 106370 https://doi.org/10.1016/j.petrol.2019.106370.

Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Spring 63 (1), 3–42. https://doi.org/10.1007/s10994-006-6226-1.

Giro, R.A., Bernasconi, G., Giunta, G., Cesari, S., 2021a. A data-driven pipeline pressure procedure for remote monitoring of centrifugal pumps. J. Petrol. Sci. Eng. https://doi.org/10.1016/j.petrol.2021.108845, 108845.

Giro, R.A., Bernasconi, G., Giunta, G., Cesari, S., 2021b. Predicting Pigging Operations in Oil Pipelines. *Pipeline Technology Conference 2021*. EITEP Institute, Berlin. https://doi.org/10.6084/m9.figshare.16553286.

Giunta, G., Cesari, S., Giro, R.A., Bernasconi, G., 2020. Digital transformation of historical data for advanced predictive maintenance. In: Abu Dhabi International Petroleum Exhibition & Conference (ADIPEC). Abu Dhabi. https://doi.org/10.2118/202906-MS.

Góes, M.R., Guedes, T.A., d'Avila, T.C., Vieira, B.F., Ribeiro, L.D., Campos, M.C., de Secchi, A.R., 2021. Virtual flow metering of oil wells for a pre-salt field. J. Petrol. Sci. Eng. 203, 108586 https://doi.org/10.1016/j.petrol.2021.108586.

Gudmundsson, J., Celius, H., 1999. Gas-liquid metering using pressure-pulse technology. In: SPE Annual Technical Conference and Exhibition. https://doi.org/10.2118/56584-MS.

Henry, R.E., Grolmes, M.A., Fauske, H.K., 1971. Pressure-pulse Propagation in Two-phase One- and Two-Component Mixtures. https://doi.org/10.2172/4043485. Argonne.

Hsu, Y.-Y., 1972. Review of Critical Flow Rate, Propagation of Pressure Pulse, and Sonic Velocity in Two-phase Media. NASA, Cleveland. Tratto da. https://ntrs.nasa.gov/citations/19720019314.

Jolliffe, I., 2005. Principal Component Analysis. Wiley Online Library. https://doi.org/10.1002/0470013192.bsa501.

Kanin, E., Osiptsov, A., Vainshtein, A., Burnaev, E., 2019. A predictive model for steady-state multiphase pipe flow: machine learning on lab data. J. Petrol. Sci. Eng. 180, 727–746. https://doi.org/10.1016/j.petrol.2019.05.055.

Kanin, E., Vainshtein, A.O., Burnaev, E., 2018. The method of calculation the pressure gradient in multiphase flow in the pipe segment based on the machine learning algorithms. In: IOP Conference Series: Earth and Environmental Science. Novosibirsk. https://doi.org/10.1088/1755-1315/193/1/012028.

Kumar, A., Olmi, C., Ogundare, O., Jha, P., Bennett, D., 2020. Detecting pipeline anomalies and variations in acoustic velocity in multiphase flow regimes using computational fluid dynamics. Open J. Fluid Dynam. 10, 184–197. https://doi.org/10.4236/ojfd.2020.103012.

Qiang, X., Chenying, L., Xinyu, W., Yeqi, C., Haiyang, Y., Wensheng, L., Liejin, G., 2021. Machine learning classification of flow regimes in a long pipeline-riser system with differential pressure signal. Chem. Eng. Sci. 233, 116402 https://doi.org/10.1016/j.ces.2020.116402.

Quinlan, J.R., 1986. Induction of Decision Trees. *Machine Learning*. https://doi.org/10.1007/BF00116251.

Taylor, G.I., 1935. Turbulence in a contracting stream. ZAMM - J. Appl. Math. Mech. 91–96. https://doi.org/10.1002/zamm.19350150119.

Unalmis, O.H., 2015. The use of sound speed in downhole flow monitoring applications. Proc. Meet. Acoust., 045003 https://doi.org/10.1121/2.0000069.

Welch, P., 1967. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. IEEE Trans. Audio Electroacoust. 70–73. https://doi.org/10.1109/TAU.1967.1161901.

Yan, Y., Wang, L., Wang, T., Wang, X., Hu, Y., Duan, Q., 2018. Application of Soft Computing Techniques to Multiphase Flow Measurement: A Review. Flow Measurement and Instrumentation, pp. 30–43. https://doi.org/10.1016/j.flowmeasinst.2018.02.017.

Ye, J., Guo, L., 2013. Multiphase flow pattern recognition in pipeline–riser system by statistical feature clustering of pressure fluctuations. Chem. Eng. Sci. 102, 486–501. https://doi.org/10.1016/j.ces.2013.08.048.