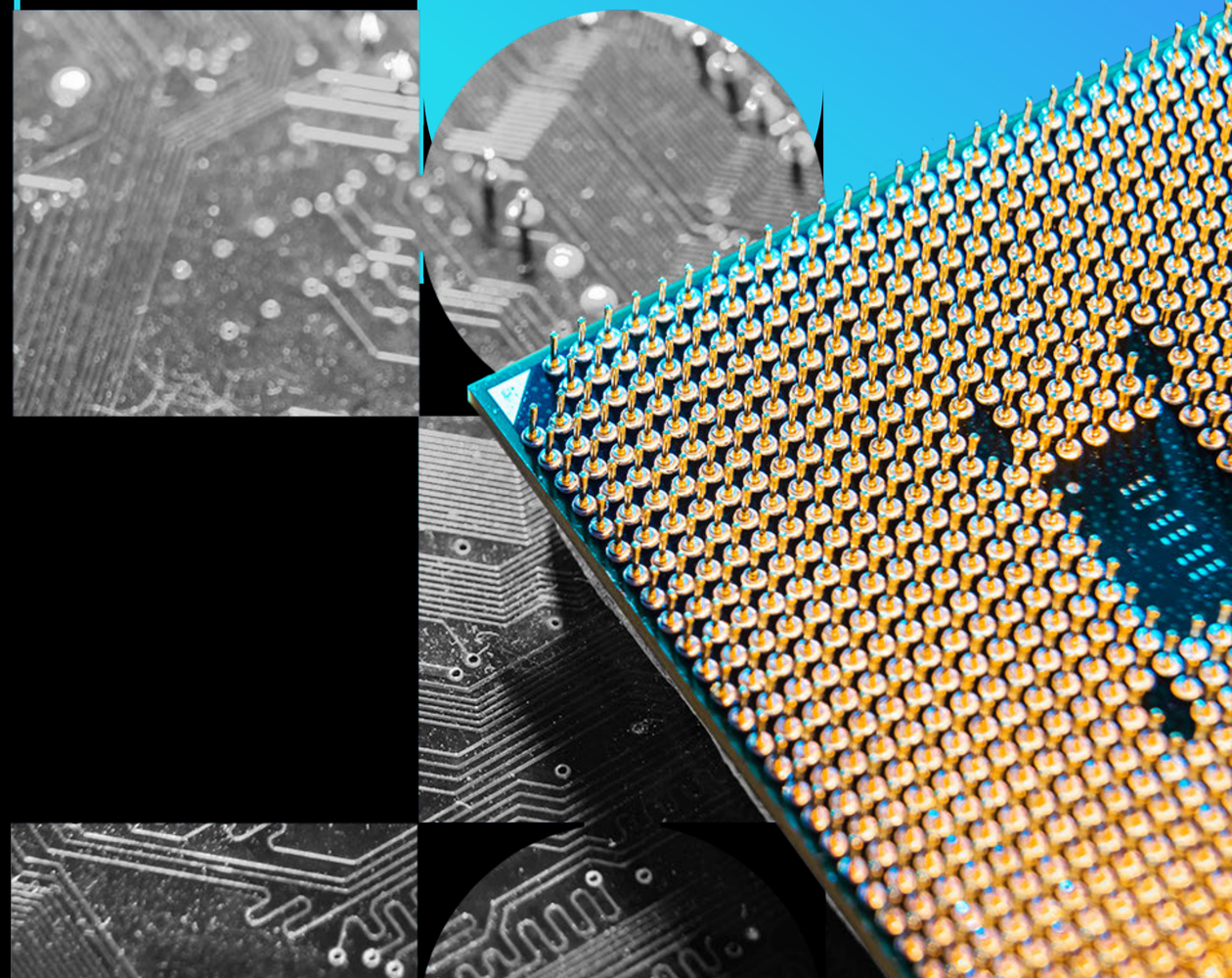
COGNITE

Physics-Guided Machine Learning:

Putting AI to work in industry



Physics-Guided Machine Learning:

Putting AI to work in industry

Table of contents

About Cognite

Cognite is a global industrial SaaS company that supports the full-scale digital transformation of asset-heavy industries around the world. Our core Industrial DataOps platform, **Cognite Data Fusion®**, enables data and domain users to collaborate to quickly and safely develop, operationalize, and scale industrial AI solutions and applications.

Cognite Data Fusion® codifies industrial domain knowledge into software that fits into your existing ecosystem and enables scale from proofs of concepts to truly data-driven operations to deliver both profitability and sustainability.

WHY COGNITE DATA FUSION® → BENEFITS FOR YOUR TEAMS → CONTACT SALES →

Summary

Hydrocarbon production systems generate huge data sets, often with time series stretching back decades. However, much of the data may be obsolete due to changing reservoir conditions and modifications to assets, and there may be scant data close to optimal operating conditions due to the inadequacy of existing optimization tools.

Data science, artificial intelligence (AI), and machine learning can contribute significantly to the optimization of production operations, and there is a trend toward hybrid AI, which combines data science with traditional physics-based simulators to deliver added value.

Physics-guided machine learning can add tremendous value to digitalization initiatives across a wide range of production optimization use cases and speed up decision processes that mitigate production losses in complex industrial phenomena.

This paper explains how to use physical principles in feature engineering to improve machine learning outcomes. Equipped with energy, mass, and force balances; pressure, volume, and temperature (PVT) data for production fluids; and dimensional and order-of-magnitude analyses, oil and gas companies can squeeze additional value from a pure data-based approach while avoiding expensive, time-consuming, and often inaccurate simulations.

WHY COGNITE DATA FUSION® → BENEFITS FOR YOUR TEAMS → CONTACT SALES →

Introduction

Artificial intelligence (AI) has been immensely successful in areas such as image recognition, natural language processing, advertising, and games — let's call them classic applications of AI. However, for industries such as oil and gas and manufacturing, the success stories are fewer, despite the high value potential. **This is because of the fundamental differences between the classic applications of AI and those used for industrial problems.**

There are four main reasons why:

--> One: For many of the classic applications of AI there are few or zero competing methods. One example: There are few mathematical models describing consumer behavior. In the oil and gas industry, in comparison, the majority of the problems are governed by the laws of physics and can be described using mathematical and phenomenological models that form the basis of advanced simulators. The industry has been using these simulators for decades to support critical decisions, and while the simulators have varying degrees of accuracy and uncertainty, users have an understanding of them and take uncertainty into account when making decisions based on their results.

--> Two: For many of the classic applications of AI the consequence of an erroneous prediction is not severe, and the size of the error is often not important. As an example, if an irrelevant advertisement is displayed on a website, it's not the end of the world. Either it results in a click or it doesn't. For the oil and gas industry, the size of the error is usually critical. A small error in the predicted surge volume is usually not a problem, but a large error could lead to a trip or, worse, a flooding incident.

--> Three: For classic applications of AI the amount of training data may be enormous. For example, the ImageNet data set contains more than 14 million pictures. The amount of text available for natural language processing is almost unfathomable. For some applications it is even possible to automatically generate training data. It is a common misconception that the oil and gas industry has large amounts of data. Although a typical installation will be instrumented with thousands of sensors that may have been collecting data for decades, the actual amount of relevant data is small.

--> Four: Finally, an important difference between some of the industries where AI has been successful and the oil and gas industry is the quality of the data. A typical data set used by classic applications of AI will have no or negligible noise levels. Much of the data used in oil and gas engineering comes from physical sensors located in harsh environments, which means they are subject to varying degrees of noise and bias and different raw data compression levels.

Machine learning has a clear value potential in the oil and gas industry, but it must go hand-in-hand with physics.

What is physics-guided machine learning?

Early in the Fourth Industrial Revolution it was widely believed that through digitalization all problems would be solved using AI, and that machine learning would replace mathematical models. That understanding is changing. AI and machine learning are increasingly seen as complementary tools to be used with existing industry-specific tools (for example physics simulators). One example of this is the use of mathematical modeling and feature engineering to reinforce a machine learning model.

Constructing physics-guided machine learning

Assume we want to predict a set of parameters *Y* from a set of observed variables *X*. Typically *X* represents our sensor data. We denote this relationship as

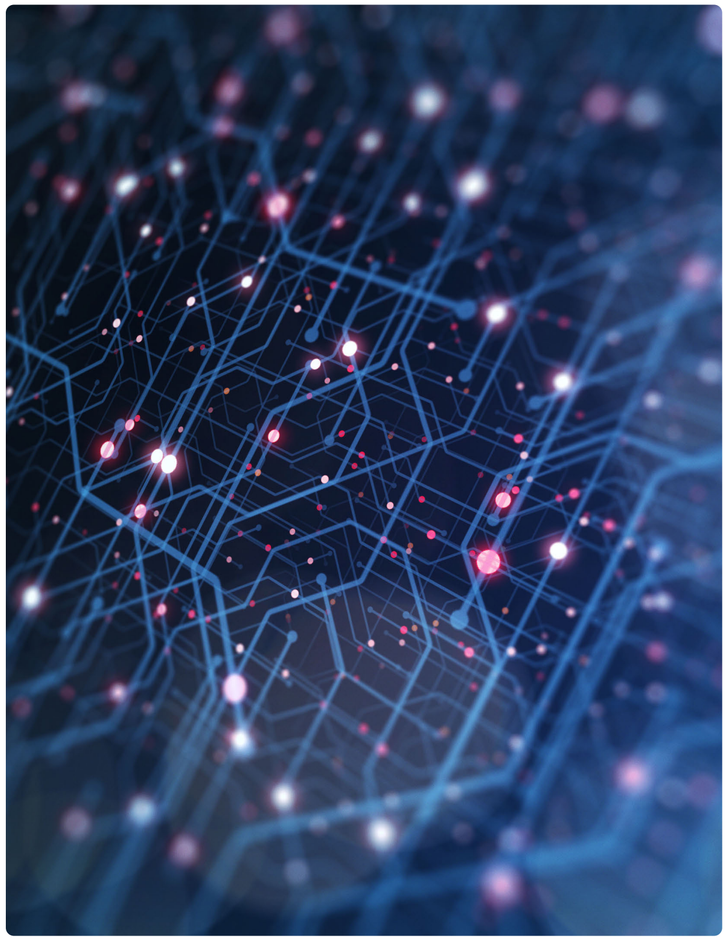
$$Y=f(X),$$

where *f* is our predictive model. One example of an application is where *Y* is a property that is not continuously measured, such as the quality of a product. Instead of only relying on infrequent spot samples, it is possible to create a model *f* that approximates the product quality *Y* based on the state *X* of the system. This is often called a **virtual or soft sensor**, and it is particularly useful when the results of spot samples are not available in time to carry out mitigating actions if the quality is not satisfactory.

There are several ways to create the model *f*. Historically this was done using physics insight and mathematical modeling. There are various techniques for deriving such models, where the most rigorous approach is based on **first principles** like conservation and balance principles. Relevant examples are conservation of mass, momentum, volume, and energy, which are the foundation for many of the successful physics simulators used in the oil and gas industry.

Not all problems are easily described using first principles, or the resulting mathematical model may be too complex to be solved within a reasonable time frame. An effective approach is to average effects in time or time and space, reducing the complexity of the phenomena that are modeled and sometimes also the number of spatial dimensions. **Turbulence modeling** is an example of averaging small-scale effects while **hydraulic modeling** is an example of averaging effects in entire spatial dimensions.

On the other side of the spectrum we find **empirical or phenomenological modeling**, where only measurements are used to derive the model. Pure machine learning belongs to this category. In between, there is a continuous range from rigorous first-principle modeling to pure empirical modeling, and this is the area we want to explore to understand how we can construct physics-guided machine learning.



Mathematical modeling vs machine learning

Let's compare the strengths and weaknesses of the more rigorous physics simulators and machine learning methods: ▶

Simulators and machine learning models clearly complement each other. Combining the two methods keeps the strengths and reduces the weaknesses.

Adding physics to machine learning models: a deep dive

One of the most fundamental problems in the oil and gas industry is pressure drop in a pipeline. This determines the maximum throughput for a given pipe length and diameter; alternatively, it shows the need for a pressure boost to meet a required flow rate.

For the purposes of this section, we will consider single-phase pressure drop measurements from two different laboratories[1] for six different fluids (He, O_2, N_2, Air, CO_2, and SF_6), two different pipes (both diameter D and roughness ϵ), different temperatures T, different pressures P, and different velocities U, giving us six different input variables, where one is a categorical variable. Assuming we need 10 data points per continuous variable to resolve the behavior, we need 105 experiments in total per fluid. In addition, reservoir fluids consist of thousands of different components, leading to a literally infinite number of possible compositions. This exponentially increasing data requirement as a function of the number of input parameters is known as the **curse of dimensionality.**

Physics simulators	Machine learning
Can predict without access to historical data (from first oil)	Requires a large set of training data for relevant conditions
Tested, tried, and proven across industries, even for critical applications	Unproven; considered hard to interpret ("black box")
Require a mathematical model derived from physics principles (not always possible)	Possible to set up without any knowledge of the underlying physics
Require a complete set of data such as boundary conditions, geometry, and fluid and material properties	Can work even on a small set of sensors (but may not be very accurate)
Can predict outside the range of data used to create and validate the model (with varying uncertainty)	High uncertainty outside the range of the training data
Can predict future events; transient models	Fewer success stories for predicting time-dependent problems
Provide all values from the equations at all positions in the numerical grid	Provides only the output variables it was trained on

[1] Swanson, C. J., Julian, B., Ihas, G. G., and Donnelly, R. J. 2002. Pipe flow measurements over a wide range of Reynolds numbers using liquid helium and various gases. J. Fluid Mech. 461, 51–60.

Zagarola, M. V., and Smits, A. J. 1998. Mean-flow scaling of turbulent pipe flow. J. Fluid Mech., vol. 373, pp. 33–79.

Obviously it is not realistic to generate such a vast amount of data, so the goal in this example is to transform the input parameters into new features, which simplifies the problem the machine learning model needs to approximate. From fluid mechanics we know that the pressure gradient depends on the pipe diameter *D* and the wall shear stress τ. The wall shear stress is not a measured quantity, but we know that the wall shear stress depends on the fluid density *ρ* and the velocity *U*. The fluid density is a property of the chosen fluid and can be computed using the laws of thermodynamics and the pressure and temperature for the individual experiments. The pressure gradient is expressed by a force balance (momentum conservation)

$$-\frac{dP}{dP} = \frac{4}{D}\tau = \frac{\lambda}{2}\frac{\rho U}{D}$$

There is a remaining unknown in the equation, namely λ, which is known as the friction factor. An important step in any data science work is to investigate the data by visualization. We will plot the friction factor instead of the pressure gradient, but we still have the challenge of selecting the parameters that the friction factors should be plotted against. Again, from fluid mechanics we know the importance of the Reynolds number *Re*, and we select that as our x-value. Applying this transformation on all the experiments results in **Figure 1** ▶. Note that we also did a log transformation of both axes.

There are three important observations:

1. The transformation collapses all the input features into one (Reynolds number). Hence, our model *Y=f(X)* will be λ=*f(Re)*

2. There seems to be a change in the trend of the friction factor around Reynolds number 2300. This is well-known from fluid mechanics and is the transition from laminar to turbulent flow.

3. By using the log transformation, the friction factor looks very close to linear for Reynolds numbers less than 2300 and something similar to a slowly exponentially decaying function for Reynolds numbers larger than 4000. It seems like a good idea to change our model to *(λ) =f((Re))*.

Figure 2 ▶ shows the result of a linear regression for *Re*<2300 and a Gaussian Process regression for *Re*>4000.

The transformation reduced our five-parameter input space to one parameter (the Reynolds number), greatly reducing the data needed. It also transformed the problem into a mostly smooth problem with a linear part and a slowly decaying part. Importantly, it also allowed us to isolate and model the discontinuous behavior around *Re*=2300.

The log transformation from a strongly nonlinear behavior to a more linear behavior reduced the need for data. In addition, it also makes it easier to

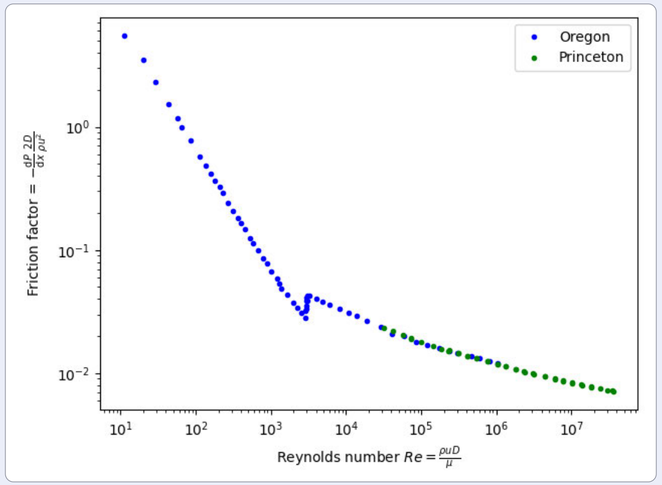


Fig. 1: Single-phase pipe flow experiments from Oregon (blue) and Princeton (green).

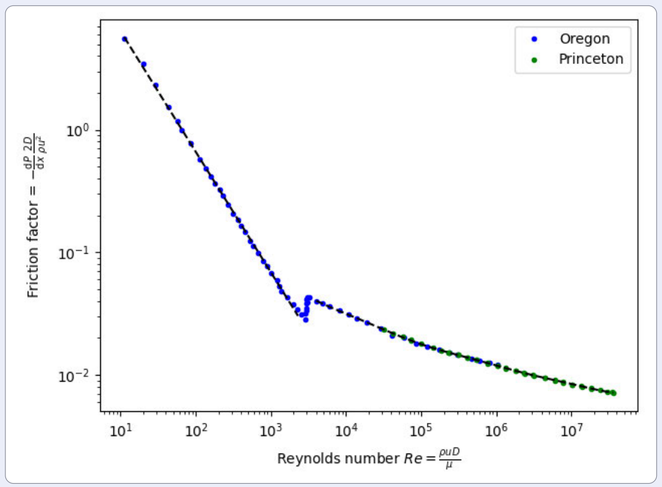


Fig. 2: Linear regression model for *Re*<2300 and Gaussian process regression for *Re*>4000.

impose regularization in the model, making it less sensitive to noise in the data.

Note that it is possible to fit a model to the entire set of data. As an example, a three-layer feedforward neural network gave good predictions. However, the extrapolation property for lower Reynolds numbers was poor. The linear model has excellent extrapolation properties, since it captures the correct physics in the transformed space.

One final but very important lesson from this example: We know from fluid mechanics that there exist well-established models for the friction factor, like the Colebrook model. From this we know that the friction factor is also a function of the relative pipe roughness

$$\frac{\epsilon_{\text{rel}}=\epsilon}{D}$$

something that is supported by other experiments. The pipes from the Oregon and Princeton experiments have different relative pipe roughnesses; however, the Oregon data does not contain data for Reynolds numbers in the region where the wall roughness becomes important (the hydraulic rough region). If we had a more extensive data set, our model should have been $(\lambda) = f((Re), \epsilon_{rel})$. This is a reminder that for nonlinear problems the importance of a feature may be strongly dependent on the operational conditions, and it may not be revealed by the available data.

Feature engineering

The single-phase pipe flow example above shows the power of feature engineering, and it illustrates two important techniques.

The first is transformation of features using **dimensional analysis**. A commonly used starting point for dimensional analysis is Buckingham's Pi theorem, which states that the number of dimensionless parameters is equal to the number of relevant variables minus the number of independent dimensions. Imagine an example with five variables ($U, \rho, \mu, D, \epsilon$) and three independent dimensions (time, length, and mass). According to Buckingham's Pi theorem, that gives us 5-3=2 dimensionless parameters, which is the same as the number of input parameters to the Colebrook friction factor model. A challenge is that there are endless possibilities for creating dimensionless parameters. It takes experience and sometimes a lot of trial and error to find the best set of dimensionless parameters.

A very attractive method in machine learning is **transfer learning**, where knowledge from solving one problem can be transferred to a different but related problem. One example is to train a facial recognition model by training on images from ImageNet to learn how to recognize a face and then train on a specific person to be able to recognize that individual. This is highly attractive for problems with scarce data but numerous similar problems. An example would be to train a model on data from a set of wells instead of each well individually. For this to be possible a given combination of input features from two different wells has to have the same output value. For the single-phase pressure drop example, the data from the two different labs was comparable when looking at the Reynolds number and the friction factor instead of the pressure gradient and the original input variables.

The second technique that was applied in the example in the previous section was the inclusion of physics models. In the experiments the fluid composition was known and the pressure and temperature were measured; however, the model needed the fluid properties density and viscosity. In some situations the fluid properties can be measured separately, but they can also be computed using equations of state, the composition of the fluid, and the pressure and temperature. By converting the fluid composition, the pressure, and the temperature to fluid properties, we relieved the machine learning method of the burden of learning this complex behavior from a scarce data set. Another way of interpreting this is that we used our physics knowledge and some sensor values to create virtual sensors of the fluid properties that are used as input features instead of the originally measured pressure and temperature.

Most sensors are already feature-engineered. A temperature sensor can be of the thermocouple type, where the electrical voltage between two dissimilar metals is measured. The temperature is calculated based on the voltage measurement

and knowledge of the proportionality constant. Another slightly more sophisticated example is a Venturi single-phase flow meter, which measures the pressure drop across a throat and computes the volumetric flow rate based on Bernoulli's equation, the continuity equation, and the fluid properties. The fluid properties can be computed based on the fluid composition, the measured pressure, and the temperature. Consequently, the majority of sensors already incorporate important physics knowledge. Feature engineering is just a continuation of this approach.

Feature-engineered variables do not have to be perfect in order to be useful. They only need to capture the main features of the behavior. The discrepancy will be compensated for by the machine learning model. Most mathematical modeling techniques have less flexibility in this sense. When a functional form is chosen, the unknowns in the model are determined by matching data. If the functional form is correct, it results in a robust model that can extrapolate with low uncertainty. However, if the functional form does not capture the true physics, it has no way of compensating for it. A simple example would be to fit a linear model to a quadratic function. The mathematical model will never be correct, while a machine learning model that takes the linear model as an input feature will be able to correct for the missing physics, at least in the range of the data.

Proxy models

When optimizing a process we do not only need to know the current state but also how changes in operational conditions will influence the outcome we are trying to optimize. This usually requires numerous calls to our model *f*. If *f* is a computationally expensive model, for example a physics simulator, it may be impossible to compute the optimal operational conditions fast enough for the operator to act on the advice. An added challenge is that a simulator may produce no results for certain input conditions (crash), or it may have a nonsmooth behavior as a function of some of the control parameters. This creates additional challenges for the optimizer.

A well-known technique from optimization is the use of proxy models. Instead of optimizing on the full model *f*, we optimize on an approximation model *f*. One technique is to create *f* by fitting a machine learning method to presimulated results from a physics simulator. For a sufficiently large data set the proxy model will inherit the accuracy and predictive capabilities of the simulator while having the evaluation speed and robustness of a machine learning model, provided the model is not used outside the range of the training data. A remedy for evaluations outside the available data set of *f* is to automatically run the simulator for those evaluations, extending the training set and retraining the machine learning model *f*.

Customizing the loss function using physics knowledge

Most out-of-the-box machine learning models use unweighted least squares as the default loss function. However, in reality, the consequence of errors is dependent on the operational conditions. When predicting surge volumes arriving at the receiving facility, large relative errors for small surges have little or no consequence, but medium errors for larger surges may lead to trips, emergency flaring, or in the worst case accidents. This is particularly important if the data set is biased to the less problematic area, which is very common due to operational practicalities; most historical operations have occurred safely, so there is little if any data for irregular conditions.

There are numerous ways to include this knowledge into the loss function, and it is an important technique to ensure that during training we prioritize the accuracy of the model in the region where accuracy is important. **A simple approach is to increase the weight in the loss function for data in the region where errors are critical.**

This needs to be combined with rebalancing the data set. For a classification problem the groups are explicitly given, making it easy to detect imbalance. For our regression problem the groups are not explicitly given, but our understanding of physics will help us determine how to classify the data into groups so that we can evaluate if we have an unbalanced data set and hence compensate for it.

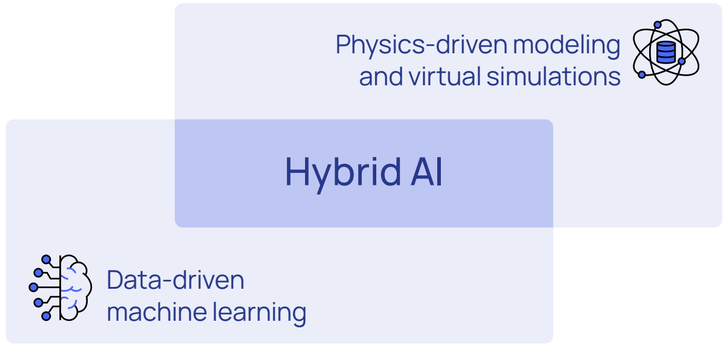
The same weighted loss function technique can be used for weighting data based on other importance factors such as the age of the data, assuming that the field is changing and older data is less relevant than newer.

It is equally important to report the error from the test data not only as a single number but as a function of the parameters that characterize the criticality of an error. This information is crucial to be able to understand the uncertainty and correctly determine safety margins for the model.

Physics-guided machine learning solutions

Cognite takes a hybrid approach to artificial intelligence, combining the best of data-driven machine learning and physics-based modeling.

Cognite differentiates from pure AI companies with a hybrid data science model unique to industrial reality.



Oil-water separation

Solution: A smart monitoring system that visualizes all data relevant for troubleshooting water contamination and a recommender system with an underlying machine learning model to identify worst actors related to high oil-in-water concentrations.

Impact: In one example, the solution saved an oil and gas operator an estimated $6 million a year.

READ MORE→

Cognite's approach

Produced water disposal is one of many challenges at oil and gas facilities with high water-cut wells. Keeping the oil contamination level in the produced water below environmental limits requires an efficient separation process. Obtaining produced water that meets environmental regulations requires an efficient separation process, which is governed by a series of complex physical interactions.

Significant production losses are associated with situations with high oil-in-water levels, because safely discharging water to the sea requires slowing down production while troubleshooting for worst actors on the facility.

To identify what could be causing high oil-in-water concentration, operators often take spot sample measurements at different parts of the production facility and then perform mitigating actions once the bad actor is located. Operators rarely have much information to determine where to start the search, however, which can make finding the bad actor a time-consuming process. Each spot sampling campaign can take up to two hours and occur multiple times a week.

Separation of oil-water dispersions is a complex, multistage process that involves gravity settling (separators), centrifuging (hydrocyclones), floatation (degasser), and the use of chemicals. In situations where the disposed water is highly contaminated, it is extremely difficult to determine which part of the plant is responsible for the problem. Possible causes range from excessive emulsification due to wellhead choke setting to inefficient floatation in the degasser due to unfavorable pressure.

To make matters even more complicated, oil and gas plants undergo continuous adjustments imposed by control room engineers in order to maximize production and minimize the risk of hazards. Furthermore, occasional modifications to the plant composition, such as the startup of a new well or replacing equipment or injection chemicals, may have a significant impact on the produced water treatment.

It is practically impossible to accurately model oil-in-water concentrations based on live operational conditions with a traditional (deterministic) approach. Even the most sophisticated process simulators would require tremendous computational resources and skilled engineers and still yield undeterminable accuracy.

The only realistic approach to modeling oil in water is by means of regression, using computing power to find hidden patterns and relationships between

operational conditions (X) and the oil-in-water concentration (Y). Furthermore, since the problem is both multivariate and nonlinear in nature, we have to solve a nonlinear multivariate regression problem.

The technical toolkit required to solve this sort of problem exists within the machine learning domain. It includes ensemble algorithms such as gradient boosted trees (GBT) and recurrent neural networks (RNN). One important difference between these algorithms is the way temporal coherency is embedded in their respective architectures. GBTs consider each row of the data set as a system snapshot, while RNNs take into account the sequential nature of time series data.

Addressing obstructive events in historical data, for example the replacement of equipment or a chemical compound, requires a dynamic machine learning approach. One such approach is to automatically retrain and reconfigure models until validation criteria are met.

The output from machine learning models needs to undergo comprehensive processing in order to render it interpretable. Machine learning interpretability libraries such as SHAP and LIME let users extract local importance of features with respect to any given target prediction. This is an essential aspect of the process, as the importance measure will in turn be associated with a potential root cause of local oil-in-water observation.

Physics and domain knowledge are included in the model through an extensive data engineering pipeline. First-principle physics modeling of key physical processes, such as choke dispersion and separator efficiency, provides a way to compress the variable space and reduce the number of dependent variables in the data set. Multiphase flow and process simulators Digital Oil Field and Unisim, respectively, enrich the data set with key data such as fluid properties and well-specific flow rates. The flow rates can be used to calculate the time delay between wellhead and point of discharge. This time shift must be taken into account, as some wells are located more than 30 km from the processing facility.

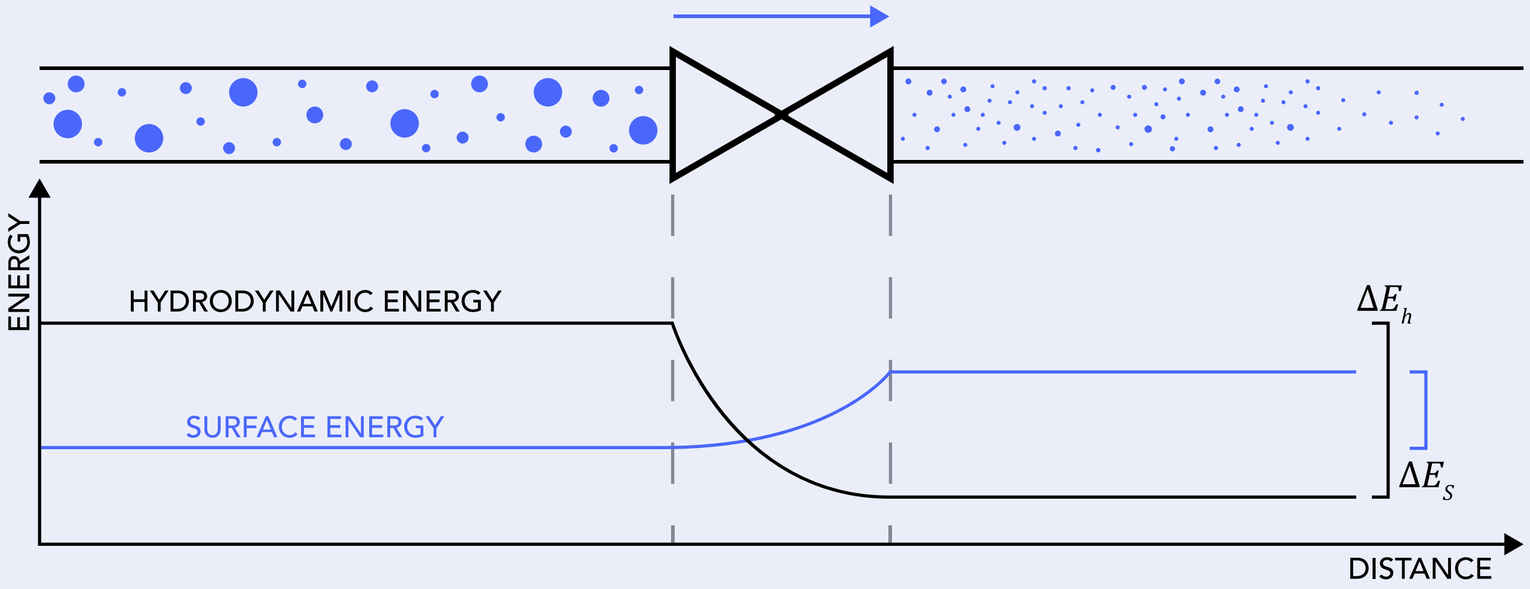
The choke dispersion model considers an energy balance between hydrodynamic kinetic (E_h) energy and potential surface energy (E_S). The former is a function of the pressure-drop ΔP_{choke}, which in turn depends on the flow rate Q_m across the choke, the kinematic mixture viscosity v_m, and k-factor. Whereas the latter depends on the droplet diameter d_{drop} and surface tension σ, where the droplet diameter is modeled using Hinze's model and the surface tension is interpolated from a lookup table generated using offline thermodynamics simulations. The ratio E_h/E_S provides an indication of the dispersion level that arises from shear forces induced by the choke settings. The ratio is expressed as

$$\epsilon_S = \frac{\Delta E_h}{\Delta E_S} = Q_m k \frac{\Delta P_{choke}}{v_m \sigma} \frac{d^2_{pipe}}{d^2_{drop}}$$

where d_{pipe} is the pipe's inner diameter.

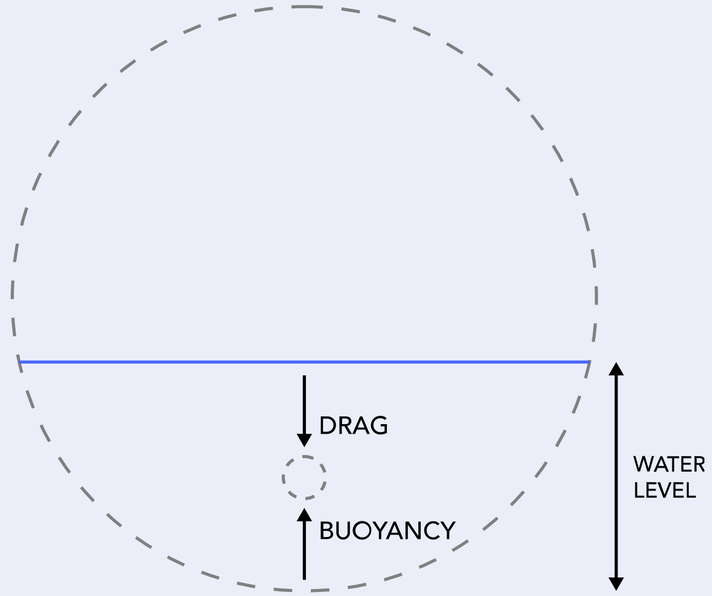
The ratio also enables us to compress the input variable space significantly and reduce the number of dependent features to train on. Although the absolute value of the dispersion levels can be inaccurate as a result of the leading order approximation and lack of tuning, when used as an input parameter to machine learning, it showed significant improvement of the model predictions as well as the importance allocation.

The separator efficiency model was formulated by means of a time-scale balance approach. Here the buoyancy time scale that arises from the Stokes drag and buoyancy force is compared to the separator residence time scale of the water body. This model results in a dimensionless parameter that comprises multiple independent variables, including flow rates, temperature, separator geometry, water level, and fluid properties. Instead of introducing each parameter as an input to the machine learning model, this single nondimensional parameter represents the entire separator. When importance is allocated to this parameter, the user of the tool will understand that this particular equipment is behaving abnormally.



HYDRODYNAMIC ENERGY LOSS → GENERATION OF DISPERTION

Fig. 3: An illustration of the choke dispersion model.



SEPARATOR EFFICIENCY: $\epsilon_S \propto \frac{\tau_{\text{residence}}}{\tau_{\text{buoyancy}}}$

Fig. 4: Separator efficiency model.

Figure 5 ▶ shows the information available to the operator in the control room. The upper part of the dashboard shows the feature importance for the different components. The schematics to the left show the influence of the different well templates and the main separation components, while the table to the right shows the number for each component in the separation train. The lower graph shows the predicted oil-in-water concentration in blue and the measured concentration in green. The predictions match the measured values well except for a few short time-scale incidents.

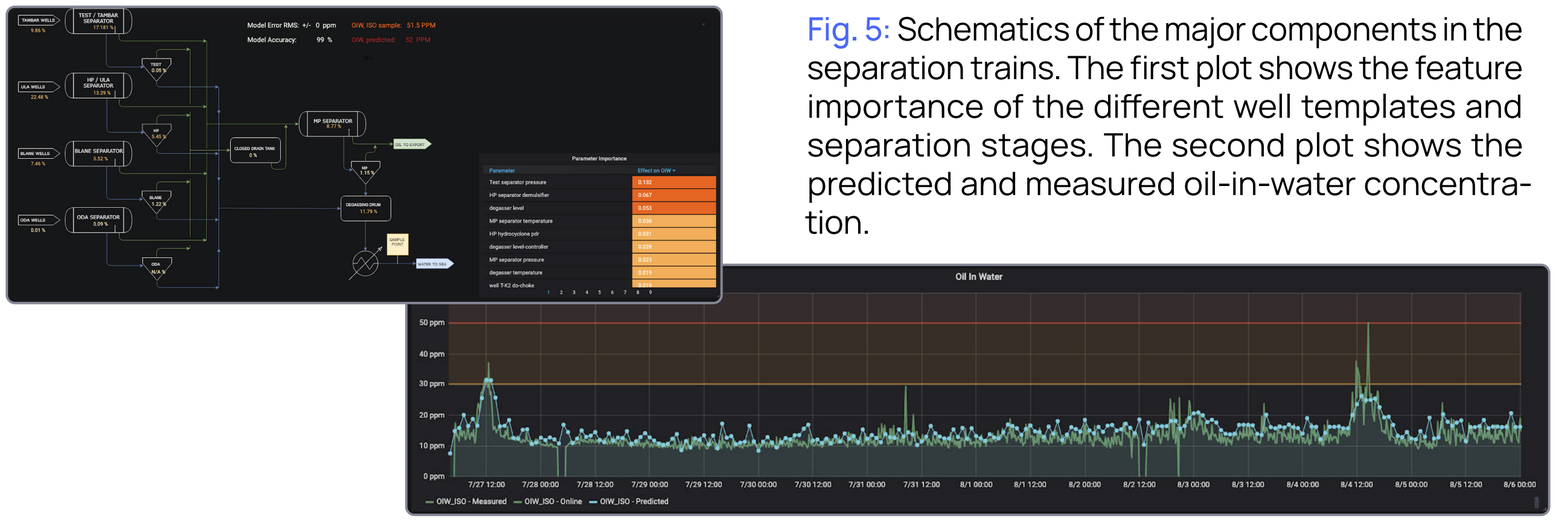


Fig. 5: Schematics of the major components in the separation trains. The first plot shows the feature importance of the different well templates and separation stages. The second plot shows the predicted and measured oil-in-water concentration.

Virtual flow meters

Solution: A combination of physical modeling of fluid flow with data analytics on sensor data, creating a virtual window into the production system that continuously supplies gas, oil, and water flow rates.

Impact: In one example, the solution saved an oil and gas operator an estimated $5-10 million a year by giving petroleum engineers and field operators 24-hour access to granular insights for better, faster decision-making.

READ MORE→

Cognite's approach

Flow rates of gas, hydrocarbon liquid, and water are key inputs to most optimization solutions. Upstream of separation, the flow is a mixture of gas, oil, and water, making measurement a difficult task. Multiphase flow meters (MPFM) can measure two- or three-phase flow using different techniques, but these meters are expensive and need frequent calibration in order to produce reliable measurements.

To understand a multiphase virtual flow meter (VFM), it is useful to understand how a typical MPFM works. Designs may differ, but the operational principle of most meters is as follows: The fluids are mixed and pass through a throat where the pressure drop is measured, similar to most single phase flow meters. The average density is measured using a gamma densitometer or an x-ray sensor, and the water fraction is measured using a capacitance or conductance sensor. The fluid properties are computed based on the fluid composition and the measured pressure and temperature, and the rates of the different phases are then computed based on a mathematical model for the pressure drop across the throat.

A virtual flow meter is a virtual sensor that uses the existing sensors (as shown in **Figure 6** ▶) combined with a mathematical model of the multiphase flow. Many commercial vendors offer VFMs. What these VFMs have in common is that they are based on rigorous models for conservation of mass, momentum, energy, and volume. These are sophisticated solutions that require little to no data, can predict outside the available data, and can be used for look-ahead and planning applications. A virtual flow meter based on physics-guided machine learning, in comparison, uses simpler and more approximative physics models.

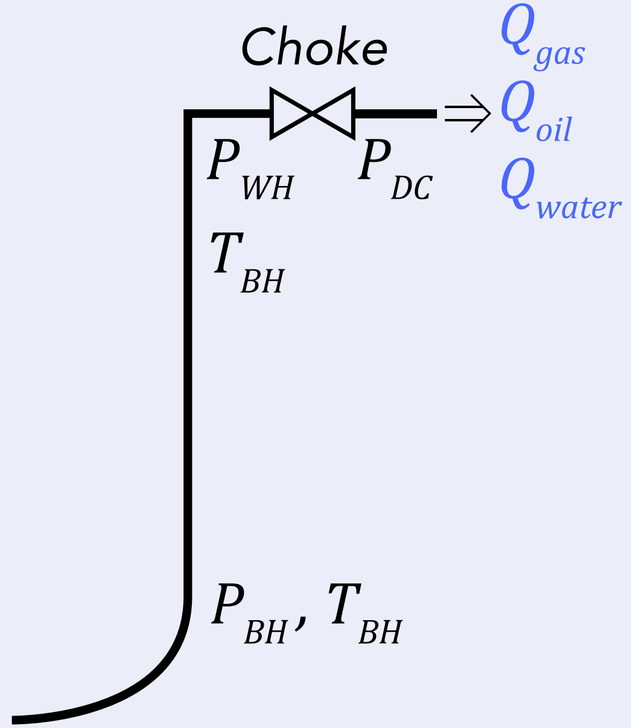


Fig. 6: An illustration of a well and the commonly available sensors used in a virtual flow meter.

We have derived a list of different engineered features where some of the features require additional information, such as a CV curve for the choke, fluid properties, and information about the wellbore. If any part of this information does not exist, it can either be approximated or a different engineered feature can be selected. Remember that the engineered feature does not have to be exact, but it should approximate the correct trend. The purpose of the machine learning model is to compensate for the imperfections in the physics.

An initial observation is that the well tests are commonly reported at standard conditions, but the physics is governed by the in-situ conditions. The conversion from standard conditions to in-situ conditions requires information about the fluid densities as a function of pressure and temperature, as well as flashing between liquid and gas. If the correct information is not available, it can be approximated using ideal gas law for gas and assuming incompressibility for the liquids.

The production choke can be viewed as a single-phase flow meter with an adjustable throttling. From Bernoulli's equation we can derive a simple valve equation which relates the measured pressure drop across the valve $dP_{choke}=P_{WH}-P_{DC}$ to the mixture's volumetric flow rate, the flow area in the choke, and fluid properties. The valve opening is usually reported as the fraction of the stem travel, and we convert it to the flow area using the choke CV curve.

The pressure drop in the well is the sum of the hydrostatic pressure drop in the well and the frictional pressure drop

$$dP_{well}=P_{BH}-P_{WH}=\rho_M gH+\frac{L}{2D}\rho_M\lambda(Re_M,\epsilon_{\text{rel}})U^2$$

where ρ_M is the mixture density, g is the acceleration due to gravity, H is the difference in height between the bottomhole and wellhead pressure sensors, L is the length of the wellbore between the bottomhole and wellhead pressure sensors, D is the wellbore inner diameter, Re_M is the mixture's Reynolds number, ϵ_{rel} is the relative wellbore roughness, and U is the average velocity of the fluid mixture.

From our single-phase flow experiment example, we already know how to estimate the frictional pressure drop, assuming homogeneous mixing of the phases. The densities for the different phases and the height difference between the pressure sensors gives the hydrostatic pressure drop. Note that both the frictional and hydrostatic pressure drop estimations require an estimate of the cross-sectional fraction of each phase.

Assuming low velocity, the pressure drop will be dominated by the hydrostatic pressure drop, and hence we have a good estimate of the gas-liquid ratio by making an assumption of the water cut. Since the density difference between oil and water is low, it is a relatively robust estimation. This is an indication that the well pressure drop dP_{well} has a strong influence on determining the gas-liquid fraction. From this we understand how error and drift in these sensors affect the model.

The heat balance in the well is another important engineered feature. Again, this is related to the mass flow F, the specific heat capacity of the phases C_p, the heat transfer coefficient Ω, and an estimation of the surrounding temperature T_r.

$$FC_p(T_{BH}-T_{WH})=\Omega\pi DL(T_f-T_r)$$

The difference in heat capacity between oil and water is usually significantly larger than the difference in density, indicating that the heat balance engineered feature strongly influences the estimation of the water cut. From our experience, the wellhead temperature sensor is often the most unreliable sensor. If it is poorly insulated, it is strongly affected by weather conditions, causing an increased uncertainty in the water-cut predic tions. However, if weather information exists, a machine learning-based model will to a certain extent be able to compensate for this. Unknown or uncertain parameters, such as the heat transfer coefficient Ω and the rock temperature T_r are effectively estimated by the machine learning algorithm.

Note that several of the parameters may change along the wellbore. In this simple approach, this

is handled by using a single representative value. More sophisticated approaches would integrate along the wellbore.

Figure 7 ▶ shows comparisons of the volumetric oil (Qo) and water (Qw) rate between well test data (black line) and virtual flow meter predictions (red line), using the approach detailed above. Note that the well test data in the comparison was not part of the training data set for the virtual flow meter model.

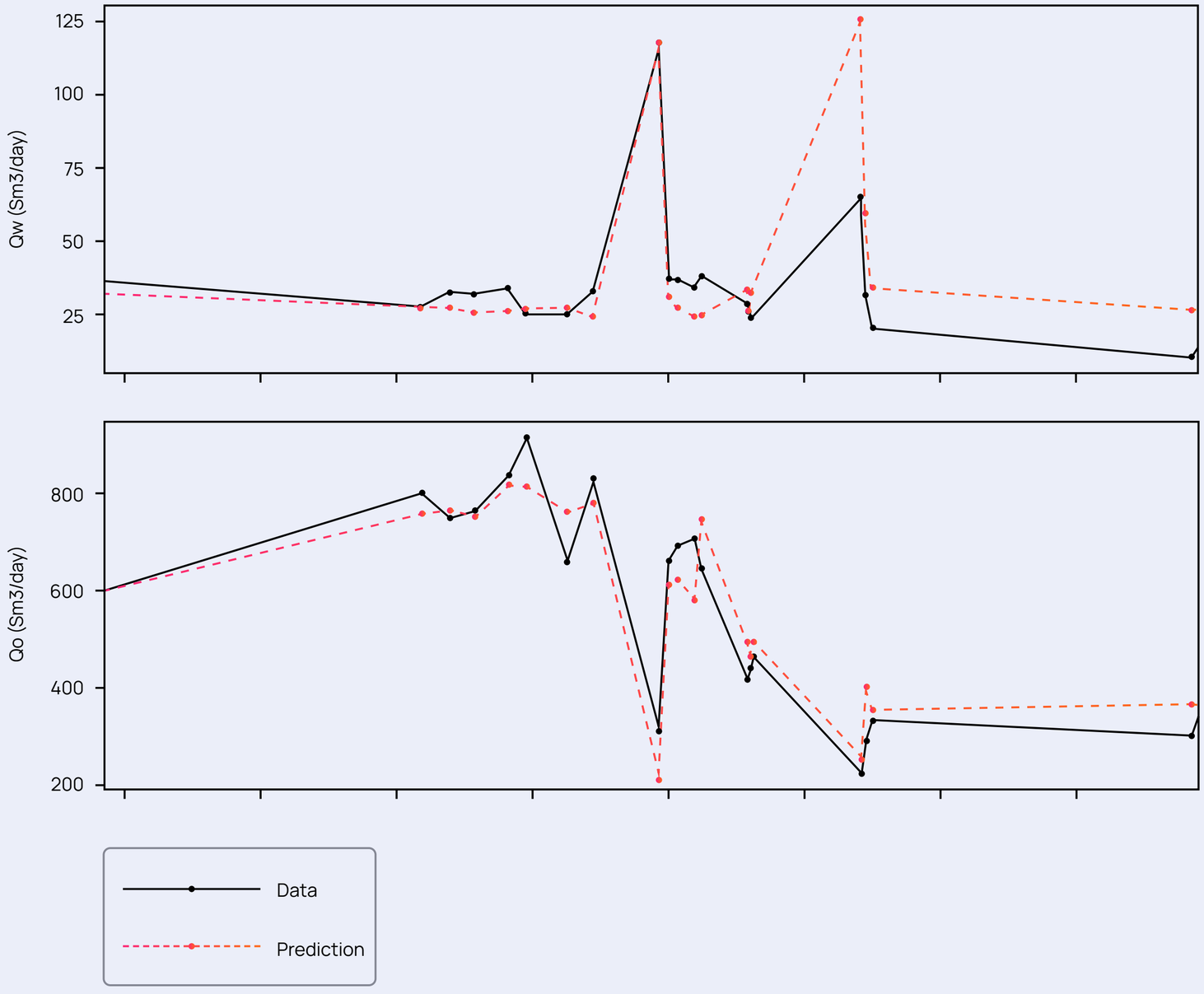


Fig. 7: Comparison of well test data and virtual flow meter results (the x-axis corresponds to the data points time stamp). The first plot shows the volumetric oil rate Qo, while the second plot shows the volumetric water flow rate Qw.

Conclusion

AI and machine learning should not be considered a goal in digitalization, but rather seen as another tool in the toolbox available to heavy-asset industries. Using machine learning to replace an existing solution is not disruption; however, when machine learning is used to solve previously unsolvable problems, or when it significantly outperforms existing solutions, then it becomes a disruptive tool. **Combining our understanding of physics with data is the key to unlocking the potential of machine learning in industrial settings.**

Some consider the addition of physics a sign of defeat, since the appeal of machine learning is that models are supposed to find relations themselves based on nothing but data. This couldn't be further from the truth. **The combination of physics and data science represents an opportunity to gain a competitive advantage.** Machine learning finds patterns from information, and by adding physics, we provide more information — and more importantly, more accurate information.

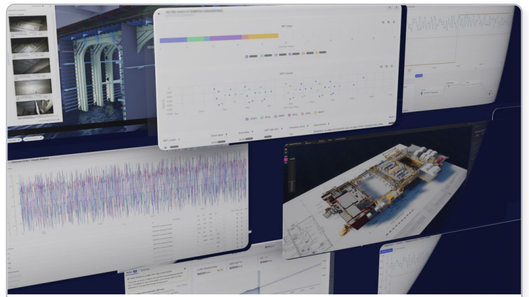
The oil and gas industry already has the subject-matter experts needed to take advantage of physics-guided machine learning. Now the challenge is to set up cross-disciplinary teams with both subject-matter experts and data scientists, and to create a common working language that both camps can speak as they collaborate.

The building blocks are there: the data, the tools, and the domain knowledge. It is up to the industry to put them all together to unleash the value potential that lies ahead.

WHY COGNITE DATA FUSION® → BENEFITS FOR YOUR TEAMS → CONTACT SALES →

Want to know more about our product?

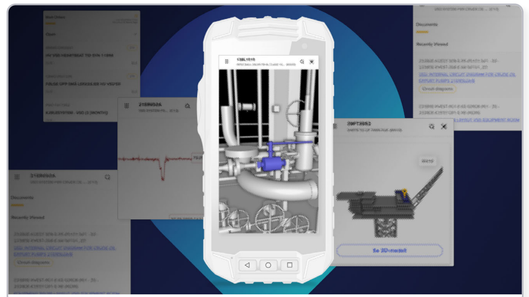
Explore more insights from Cognite



PRODUCT TOUR

Learn from Cognite customers and product managers how Cognite Data Fusion® simplifies and streamlines the data experience of a subject matter expert.

WATCH NOW →



CUSTOMER STORIES

Discover how Cognite Data Fusion® makes data more accessible and meaningful, driving insights that unlock opportunities in real-time, reduce costs, and improve the integrity and sustainability of your operations.

GO TO STORIES →

ANALYST REPORT

Customer interviews and financial analysis reveal an ROI of 400% and total benefits of $21.56M over three years for the Cognite Data Fusion® platform.

READ THE REPORT →

BLOG

Discover our rich catalog of industry insights and technology deep dives.

READ OUR NEWEST BLOGS →

WHY COGNITE DATA FUSION® → BENEFITS FOR YOUR TEAMS → CONTACT SALES →

COGNITE

COGNITE.COM →