Doctoral theses at NTNU, 2022:165

Mathilde Hotvedt

On a hybrid approach to model learning applied to virtual flow metering

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical
Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Mathilde Hotvedt

On a hybrid approach to model learning applied to virtual flow metering

Thesis for the Degree of Philosophiae Doctor

Trondheim, June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

ISBN 978-82-326-6719-2 (printed ver.)
ISBN 978-82-326-5668-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

IMT-report 2022-5-W

Doctoral theses at NTNU, 2022:165

Printed by NTNU Grafisk senter

Summary

Process modeling using first-principle equations has existed for centuries as a methodology to represent and analyze real-world processes. In time with increasing computing power and sensor data availability, data-driven modeling for processes has gained attention. Even though data-driven modeling, or machine learning, has shown remarkable results in fields such as image classification and speech recognition, it has yet to be adopted as the preferred approach for process modeling. Arguably, this is due to the long history of first-principles modeling, along with the inherent black-box nature of data-driven models. The latter causes a lack of model explainability, which, in turn, can result in distrusting the predictions originating from data-driven models. Furthermore, disregarding physical laws that have been acknowledged for centuries to model processes can seem irrational.

Hybrid, or gray-box, modeling is a methodology with a vision to utilize all available knowledge, both physics and data, to model processes. It combines first-principle equations with data-driven techniques and is especially intriguing for inherent complex processes where the physical behavior is partly unknown or challenging to model with first principles. One such process is the petroleum production system. The multiphase flow rate through the production system is challenging to model with required precision using first principles due to uncertain subsurface properties and complex dynamic behavior. Furthermore, available sensor data is often limited or of low quality. Therefore, a hybrid modeling approach seems of significant importance to models predicting the multiphase flow rates as it attempts to exploit all available information to its full extent.

The work leading to this thesis has explored hybrid solutions for virtual flow metering. A virtual flow meter (VFM) is a soft-sensor that utilizes process models and already existing sensor measurements, such as pressures and temperatures, to compute the multiphase flow rate at strategic locations in a petroleum asset. The main part of this thesis is a collection of six peer-reviewed papers, three journal publications, and three conference publications. In addition to the paper collection, this thesis introduces the topic of hybrid modeling for virtual flow metering to provide context to the publications. The main contributions of the six publications can be summarized as follows: firstly, a framework for simultaneous estimation of all parameters in a model with varying degrees of hybridity has been proposed. Secondly, six hybrid VFM model types were developed from real and historical production data from a petroleum asset. Thirdly, several hybrid model properties such as explainability, scientific consistency, flexibility, and accuracy have been examined. Lastly, two methods, one to include uncertainty in the modeling, and one to address the inherent nonstationarity of the underlying process to sustain the long-term VFM performance, have been proposed.

The key takeaway of the work leading to this thesis is that hybrid modeling is challenging, yet, also essential for obtaining high accuracy VFMs in certain scenarios. The contributions have shown that the task of balancing learning from physics and learning from data is nontrivial, and if incautious, the hybrid model can exploit the disadvantages of both the mechanistic and data-driven modeling domain instead of the advantages. On the other hand, the results also showed that for processes with unknown or unmodeled physics, a hybrid model can offer improved performance over a mechanistic model, and with little available process data, a hybrid model can obtain a higher performance than a data-driven model. Moreover, in the presence of nonstationarity and little data, frequent updating of a hybrid VFM has shown essential to sustain the prediction accuracy over time.

From the results, it is believed that hybrid modeling can be generalized to other applications and can offer improved performance over a mechanistic and data-driven approach. Furthermore, the solution for hybrid modeling presented in this thesis can be conveniently integrated with existing mechanistic process models in the industry. Naturally, the domain of hybrid modeling for virtual flow metering has not been fully explored. The most promising future research direction is combining hybrid modeling with methods that enable learning from more than one petroleum well at a time.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Philosophiae Doctor (Ph.D.) at the Norwegian University of Science and Technology (NTNU). The work behind this thesis has been completed at the Department of Engineering Cybernetics. The project has been a part of BRU21 - NTNU Research and Innovation Program on Digital and Automation Solutions for the Oil and Gas Industry (`www.ntnu.edu/bru21`) and supported by Lundin Energy Norway. Professor Lars Imsland has been the main supervisor, with Bjarne Grimstad from Solution Seeker and Dag Ljungquist from TechnipFMC as co-supervisors.

Acknowledgments

The journey to the delivery of this thesis actually started during the spring of 2018. I was working hard on my Master's thesis when my supervisor Lars asked if I would be interested in taking a Ph.D. I hadn't given the idea much thought and was in fact under the impression that a career in Academia was brain-twistingly exhausting. He kept on regularly asking me in our meetings and even said at one point that a Ph.D. that was *just perfect* for me would soon be released. In June 2018, I found myself being "tricked" into signing up for that perfect Ph.D. position - who would have thought?

In August 2018, the opening of the ball started with a meeting between my supervisor Lars, Kjartan from Lundin Energy Norway: the company that would sponsor my Ph.D., Vidar and Bjarne from Solution Seeker AS, Dag from TechnipFMC, and myself, in Lundin's offices at Lysaker. I remember Kjartan telling me that they counted on me coming up with a solution to all their problems when their asset Edvard Grieg went off plateau. I remember thinking: what is even *plateau*? I felt that I had been pushed out on slippery ice, trying desperately to remain standing up while keeping a brave face. The turning point arrived just a few moments later when I found myself in conversation with two very excited people: Vidar and Bjarne. My Ph.D. topic was just so cool and extremely interesting to them! Their excitement was, luckily, highly contagious, so my motivation for facing all of Lundin's problems with Edvard Grieg rose. My motivation increased further when I started the work at NTNU's offices in Trondheim and found that all of my Ph.D. colleagues felt the same way about their topics. I was not alone in the labyrinth.

As the weeks passed by, I learned what *plateau* meant, and I began feeling comfortable that a Ph.D. was something I could do. It was a particularly joyful year, much thanks to my awesome office mates: Joakim, Otávio, and Andreas, without whom the extremely steep learning curve would have been impossible to climb. Where would I have been without Andreas to fix all my Python problems, and Joakim and Otávio to discuss challenging petroleum problems? At the beginning of my second year as a Ph.D. candidate, I found myself moving to Oslo and being warmly welcomed at Solution Seeker's offices. I was given a seat next to my co-supervisor Bjarne, which I guess he regretted as soon as I started asking tons of questions. It didn't take long before I was given a part-time position with Solution Seeker, for which I am very grateful. With it, I got a whole new perspective on my work and felt that I could contribute to something meaningful that would not just end up on the shelf.

With this, I would like to give an enormous thanks to all people who have been a part of my journey to completing this thesis. First and foremost to my supervisor Lars and co-supervisors Bjarne and Dag, for supporting my ideas and work, and for improving the results with suggestions and critics. Without you, I would still be stuck on that very first paper. I would like to thank Joakim, Otávio, and Andreas for making that first year in Trondheim a cakewalk. I thank all my colleagues at Solution Seeker, for warmly welcoming me into your midst, for back-up, laughs, and joyful moments even in the middle of a pandemic that removed all my plans to travel the world. I would like to thank the BRU21 project for setting up this exciting Ph.D. position with cooperation between many companies. Last, but not least, I would like to thank my soon-to-be husband Eilif, for a shoulder to cry on in moments of exasperation and praise in moments of success, and my parents Tonje and Terje for unconditional support in my education, from age six to the completeness of this thesis.

Mathilde
March 2022

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

BNN	Bayesian neural network
DM	Data-driven model
GD	Gradient descent
MAP	Maximum a posteriori
MCMC	Markov Chain Monte Carlo
ML	Machine learning
MLE	Maximum likelihood estimation
MM	Mechanistic model
MPFM	Multiphase flow meter
NLS	Nonlinear least squares
NN	Neural network
ReLU	Rectificed linear unit
SC	Standard conditions
SGD	Stochastic gradient descent
VFM	Virtual flow meter
VI	Variational inference

Greek letters

γ	Step size / learning rate	
ϵ	Measurement noise	
η	Mass fraction	
$\hat{\theta}$	Parameter estimate	
θ	Parameter	
κ	Gas expansion coefficient	
μ	Prior mean of parameter	
λ	Regularization factor	
ρ	Density	$\mathrm{kg\,m^{-3}}$
σ	Prior standard deviation of parameter	
σ_ϵ	Standard deviation of measurement noise	

Symbols

A	Area	$\mathrm{m^2}$
a	Activation function	
$\mathcal{B}$	Minibatch	
$\boldsymbol{b}$	Bias vector	
b	Bias	
C_D	Discharge coefficient	
C_v	Valve flow coefficient	$\mathrm{USG\ min^{-1}\ PSI^{-0.5}}$
$\mathcal{D}$	Dataset	
D	Diameter	m
E	Number of iterations	
f_θ	Parametric model	
g	Gravitational constant	$9.81\,\mathrm{m\,s^{-2}}$
J	Objective function	

$\dot{m}$	Mass flow rate	$\mathrm{kg\,s^{-1}}$
$\mathcal{M}$	Step direction	
M	Molar mass	$\mathrm{kg\,mol^{-1}}$
∇_θ	First-order gradient with respect to parameters	
∇_θ^2	Second-order gradient with respect to parameters	
N_c	Unit conversion coefficient 27.3 $\mathrm{kg^{0.5}\ h^{-1}\ USG^{-1}min\ PSI^{0.5}\ bar^{0.5}\ m^{1.5}}$	
$N_\mathcal{D}$	Number of observations in the training dataset	
P_θ	Number of parameters in the model	
$\mathcal{P}$	Process	
p	Pressure	bar
p_r	Pressure ratio	
$p_t(*)$	Probability distribution of given variable $*$	
$\boldsymbol{q}$	Flow rate vector	
Q	Total multiphase flow rate	$\mathrm{m^3\,s^{-1}}$
q	Phasic flow rate	$\mathrm{m^3\,s^{-1}}$
R	Universal gas constant	$\mathrm{J\,K^{-1}\,mol^{-1}}$
T	Temperature	K
u	Choke opening	%
$\boldsymbol{W}$	Weight matrix	
$\boldsymbol{w}$	Weight vector	
$\boldsymbol{X}$	Design matrix	
$\boldsymbol{x}$	Input vector	
$\boldsymbol{y}$	Output vector	
$\hat{y}$	Model estimate	
y	Output	
Z	Gas compressibility factor	

1 Introduction

This chapter will firstly present the motivation and research objective behind the work leading to the completion of this thesis. Thereafter, a summary of the contributions leading to this thesis is given together with the list of original publications. Lastly, the remainder of this thesis will be outlined.

1.1 Motivation

For centuries, humans have tried to describe real-world objects with abstract representations (Schichl, 2004). In the modern world, such representations are otherwise known as models. Some of the older, well-known, prestigious models are, for instance, the solar system movement model by Ptolemy around 150 AD, Newton's laws of motion derived in the 17th century, and Einstein's theories on relativity published at the beginning of the 20th century. With the introduction of computers in the late 1940s, numerical simulation of mathematical models evolved to play a vital part in, for example, industrial applications (Guillaume, 2018). Furthermore, with increasing computing power, research on automatic discovery of models by inference on data escalated (Ghahramani, 2015). Presently, with numerical simulation tools, increasingly complex processes can be represented, their response to changes in process conditions analyzed, and the model used in control and optimization of the process in question (Gravdahl and Egeland, 2002).

The oil and gas industry is one of many industries where process models and digital computing technologies have been one of the keys to success (Guo et al., 2007). The petroleum production system consists of the reservoir, wells, flowlines, separators, pumps, and transportation pipelines, and examples of models exist for all components (Jansen, 2015). Reservoir models are used to simulate the response of the reservoir to petroleum depletion and play an important role in long-term production strategies, including drilling schedules and decisions regarding injection rates and target production flow rates (Foss et al., 2018). Other models can aid in the short-term, daily control and optimization of petroleum production or provide situational awareness and flow assurance. A type of model called a virtual flow meter (VFM) can monitor the multiphase flow rate at strategic locations in a petroleum asset (Toskey, 2012) and is useful for many of the above-mentioned engineering problems. With today's global dependency on petroleum production, in addition to an increasing focus on phasing out non-renewable energy sources, the importance of extracting as much petroleum as possible from already existing assets becomes evident. To this end, high accuracy VFMs yielding information about the multiphase flow rate over time can become useful also for long-term optimization strategies.

Nevertheless, a particular troublesome challenge of modeling the multiphase flow rate in petroleum production systems is the often high uncertainty in the available information (Jansen, 2015; Monteiro, Chaves, et al., 2017). For instance, describing the flow rates in detail using first-principle equations is challenging due to uncertain subsurface characteristics and complex dynamic behavior. Furthermore, available sensor data are often limited and of low quality and may even fall out in periods (Corneliussen et al., 2005). Therefore, to design high accuracy VFMs, all available knowledge should be combined and utilized to its full potential. A modeling methodology where first-principle equations are combined with machine learning (ML) techniques to exploit available data is referred to as hybrid, or gray-box, modeling. Although hybrid process models have existed since the 1990s, with application to a fed-batch bioreactor as one of the first examples (Psichogios and Ungar, 1992), hybrid models for VFMs have emerged only recently (Bikmukhametov and Jäschke, 2020a). The work behind this thesis has contributed to the development of hybrid VFMs. The top-level research objective can be stated as follows:

Research Objective

Develop and examine hybrid model solutions suitable for virtual flow metering applications.

1.2 Contributions

The top-level research objective presented in Section 1.1 has a wide scope with numerous possibilities. Therefore, the work leading up to this thesis has not covered the whole scope but has focused on some specific research directions. The main part of this thesis is a collection of peer-reviewed papers. The contributions of the papers are elaborated on in Chapter 3, while a summary is given here:

- A framework for simultaneous estimation of all parameters in hybrid models has been proposed.
- In total, six different hybrid VFM model variants have been developed using historical production data from the asset Edvard Grieg. Their performances have been compared with the performances of mechanistic and data-driven VFM models.
- Several model properties such as explainability, scientific consistency, and flexibility have been investigated, where the hybrid models have been compared to mechanistic and data-driven models.
- A method to include epistemic and aleatoric uncertainty in VFM modeling has been proposed.
- A method to address the inherent nonstationarity of the process and sustain the long-term prediction performance of VFMs has been proposed.

1.3 Publications

Six publications have been written as a result of the work on this thesis. Below is a list of the original publications ordered **A**-**F** chronologically by the date of publication. Of the six, three are conference papers, and three are journal papers. In Paper **C**, the candidate contributed to a major part of the visualization and analysis of the results, and the write-up of the paper. One additional paper was written in the duration of the Ph.D. but is not directly relevant to the top-level research objective in the thesis. Paper **E** is accepted but has not yet been published, while Paper **F** is submitted for possible publication.

Conference publications

Paper **A** M. Hotvedt, B. Grimstad, and L. Imsland (2020). “Developing a Hybrid Data-Driven, Mechanistic Virtual Flow Meter - a Case Study”. In: *IFAC-PapersOnLine* 53 (2), pp. 11692–11697

Paper **B** M. Hotvedt, B. Grimstad, and L. Imsland (2021). “Identifiability and physical interpretability of hybrid, gray-box models - a case study”. In: *IFAC-PapersOnLine* 54 (3), pp. 389–394

Paper **E** M. Hotvedt, B. Grimstad, D. Ljungquist, and L. Imsland (2022b). “When is gray-box modeling advantageous for virtual flow metering?” In: *Accepted for publication in IFAC-PapersOnLine*

Journal publications

Paper **C** B. Grimstad, M. Hotvedt, A.T. Sandnes, O. Kolbjørnsen, and L. Imsland (2021). “Bayesian Neural Networks for Virtual Flow Metering: An Empirical Study”. In: *Applied Soft Computing* 112 (1)

Paper **D** M. Hotvedt, B. Grimstad, D. Ljungquist, and L. Imsland (2022a). “On gray-box modeling for virtual flow metering”. In: *Control Engineering Practice* 118 (1)

Paper **F** M. Hotvedt, B. Grimstad, and L. Imsland (2022). “Passive learning to address nonstationarity in virtual flow metering applications”. In: *Submitted to Expert Systems with Application for possible publication*

Additional publications written in the duration of the PhD

M. Hotvedt, S. O. Hauger, F. Gjertsen, and L. Imsland (2019). “Dynamic Real-Time Optimisation of a CO2 Capture Facility”. In: *IFAC-PapersOnLine* 52, pp. 856–861

1.4 Outline

The remainder of this thesis is organized as follows. Chapter 2 will introduce background material that has been relevant for the work on the top-level research objective. Section 2.1 will introduce process modeling in general and describe the differences of mechanistic, hybrid, and data-driven models. Section 2.2 will describe how process models are usually trained using available process data. Section 2.3 will present the concept of virtual flow metering and the challenges related to modeling. Thereafter, Chapter 3 will elaborate on the contributions that have led to the completion of this thesis. Chapter 4 will discuss the choices made in the research, reflect on the generalizability of the results to different applications, and outline future promising research directions for hybrid VFMs. Finally, Chapter 5 gives the original publications, reformatted to fit the format of this thesis.

2 | Background

This chapter will introduce relevant background material of the work leading to this thesis. Section 2.1 will introduce process modeling, with a classification of model types into mechanistic, hybrid, and data-driven models. Section 2.2 will continue by describing how process models can be learned, or trained, using available process data. Thereafter, Section 2.3 will introduce virtual flow metering in petroleum production systems and discuss challenges related to these models.

2.1 Process modeling

According to the English dictionary, a physical process is a phenomenon that is either sustained or characterized by gradual changes through a series of states. It can be naturally occurring, like the decay of food or photosynthesis, or constructed, such as the boiling of water or extraction of hydrocarbons from a reservoir. Typically, a process is linked to an apparatus such as a distillation column, heat exchanger, or compressor and can be characterized by its mass, energy, and momentum bounds (Mikleš and Fikar, 2007).

Process modeling is the act of expressing a process with a mathematical model (Mikleš and Fikar, 2007). The model will have an output, or target, variable, which is the process quantity of interest, and inputs, which are remaining process conditions. The output will be related to input variables through mathematical equations. Take the boiling of water as an example. The output can be the temperature of the water, while the pressure, the effect of the heater, and the surrounding temperature are examples of inputs. An exhaustive classification of process models is challenging due to the many characteristics (Zendehboudi et al., 2018). For instance, the models can be differentiated by the degree of spatial variation, linearity, time-dependency, or determinacy. One of the more common classifications is time-dependency, separating models into dynamic and steady-state. In steady-state models, the rate of change of the process variables with respect to time is zero, and the mass, energy, and momentum of the output variable at two points in time are conserved (Gravdahl and Egeland, 2002). In other words, there is no accumulation in the system, for example, of mass. Conversely, dynamic models allow the process variables to change with time, accounting for accumulation. In this work, classification with regard to the degree of dependency on prior knowledge or process data has been the main focus. Models with a structure developed from prior knowledge such as first-principles are defined as mechanistic models, whereas models with a structure determined by inference on data are referred to as data-driven, or ML, models (Solle et al., 2016). The intersection of the two model types is called hybrid, or gray-box, models. In literature, hybrid models are also a term used for models combining discrete and continuous process dynamics. Such models are not considered in the thesis. An illustration of the classification is shown in Figure 2.1.

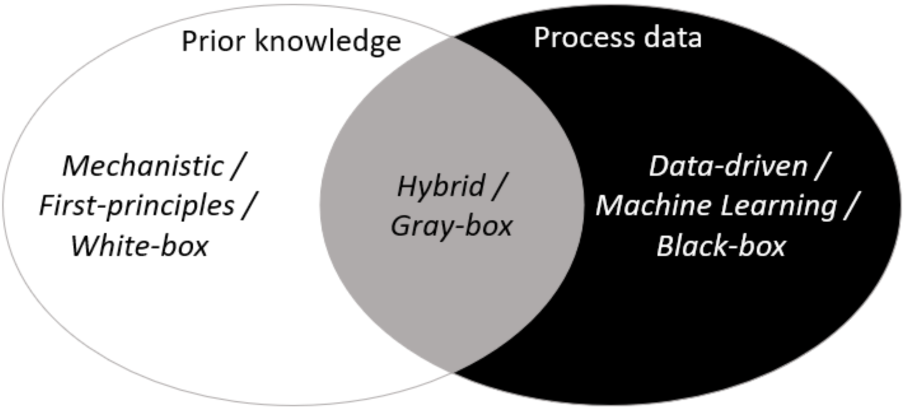

Prior knowledge
Process data
Mechanistic / First-principles / White-box
Hybrid / Gray-box
Data-driven / Machine Learning / Black-box

Figure 2.1: Overview of process model types classified according to the dependency on prior knowledge or process data. In mechanistic models, the structure is built from prior knowledge, such as first-principles, whereas in data-driven models, the structure is determined from inference on data.

The model types have different properties that generally make one advantageous over another in different scenarios. Some of these are summarized in Table 2.1, where the definitions of the concepts are given below. Naturally, there will always exist exceptions where one model type does not cohere with the expectations given in Table 2.1. For instance, mechanistic models are known for their high scientific consistency as a natural result of being based on physical laws. However, the scientific consistency will likely be limited to the range at which any model assumptions or simplifications are valid.

Table 2.1: Selected properties of the mechanistic, hybrid, and data-driven models.

Property	**Mechanistic**	**Hybrid**	**Data-driven**
Prior knowledge	High	Medium	Low
Data demand	Low	Medium	High
Explainability	High	Medium	Low
Flexibility	Low	Medium	High
Extrapolation power	High	Medium	Low
Scientific consistency	High	Medium	Low
Computational burden	High	Medium	Low
Maintenance cost	High	Medium	Low

Definition 2.1.1 (Explainability)**.** Explainability is the ability to describe, or reason about, the cause of the model response to a given input (Linardatos et al., 2020).

Definition 2.1.2 (Flexibility)**.** Flexibility is the ability to adapt to arbitrarily complex patterns in data. This concept is also known as model capacity (Goodfellow et al., 2016).

Definition 2.1.3 (Extrapolation power)**.** The extrapolation power is the ability of the model to accurately predict the process response to new and previously unobserved conditions outside the bounds of the already observed conditions (Hahn, 1977).

Definition 2.1.4 (Scientific consistency)**.** A model is scientifically consistent if the response to a given input is in line with the physical principles of the process it represents (Roscher et al., 2020).

Definition 2.1.5 (Computational burden)**.** The computational burden is the time spent on the evaluation, or computation, of the model response to a given input (excluding model training).

Definition 2.1.6 (Maintenance cost)**.** The maintenance cost is the time and effort spent on model updating to ensure an acceptable accuracy.

2.1.1 Mechanistic modeling

In mechanistic models, the equations are derived from first principles such as the mass, energy, and momentum balance equations, with possible empirical closure relations (Solle et al., 2016). Therefore, deriving a mechanistic model requires a thorough understanding of the process' physical behavior. The terminology white-box, see Figure 2.1, is used as a reference to the prior knowledge of the internal structure of the model. This is contrary to a black-box model where the structure is hidden from the developer or difficult to analyze. Mechanistic models can be developed before the start-up of the process and require no process data before deployment (Zendehboudi et al., 2018). Typically, the parameters have a physical interpretation that simplifies the specification of prior values. For example, modeling the inflow-outflow relationship of a water tank, one model parameter is the water density. If the water is fresh, the density should have a value of approximately $1000\mathrm{kg\,m^{-3}}$. On the other hand, in practice, some process data are used to tune or calibrate the model parameters to better fit the measurements.

Due to the transparency of white-box models, the explainability is high. In other words, given a change in input or parameter value, it is often possible to deduce the model response from inspection. As the model is based on physical laws, the scientific consistency is high. The aforementioned is also the reason for the commonly observed high extrapolation power to previously unseen process conditions. Nevertheless, not all industrial problems are easily formulated with first principles, for instance, due to unknown physical phenomena (Cherkassky and Mulier, 2007). The exact solution can be challenging to obtain even for simple process models and can often lead to a high computational burden. Indeed, simplifications and assumptions of the process physics are commonly necessary for a mechanistic model to be computationally feasible in real-time applications (Willard et al., 2020). Therefore, mechanistic models typically lack flexibility, and process-model mismatch commonly exists, causing low model prediction accuracy.

Ironically, many of today's well-established first principles originally emerged from experimental studies with the inference of data. This is called empirical modeling and is a data-driven modeling methodology (Zendehboudi et al., 2018). One example is the ideal gas law, which describes the relationship between the amount of substance of a gas and its pressure, temperature, and volume (Moran et al., 2014).

2.1.2 Data-driven modeling

Data-driven models are derived solely from available process data and require no prior knowledge about the process (Solle et al., 2016). Typically, the equations are generic without any attempt at capturing the underlying physics of the process, and the parameters do not have a physical representation. The motivation behind is to make data-driven models applicable to many types of problems, from image recognition to prediction of the growth rate in a fed-batch bioreactor. Typically, algorithms for training data-driven models can be differentiated into four: supervised, unsupervised, semi-supervised, and reinforcement (Hastie et al., 2009). In the first category, measurements of the output variable are available, and the model is learned by iteratively adjusting its parameters such that the error between the model output and measurements is small. Examples of such models are linear regression models or decision trees. In the second category, measurements of the output are not available, and the model learns from patterns in the input data only. Examples of algorithms are K-means clustering or Principle Component Analysis. In the third category, the models are trained using a combination of data with and without measurements of the output. In the last category, the model attempts to learn a set of sequential decisions to take to reach a specified goal. The optimal decisions are learned by receiving rewards (Sutton and Barto, 2018). Reinforcement learning is typically utilized for teaching a computer to play a game.

In this work, only supervised learning algorithms are considered for developing process models. To give examples of typical models, the linear model and the neural network model for regression will be presented mathematically below. For simplification, the steady-state versions will be considered. The linear regression model with output $\hat{y} \in \mathbb{R}$, inputs $\boldsymbol{x} \in \mathbb{R}^d$, is given by:

$$\hat{y} = \boldsymbol{w}^T \boldsymbol{x} + b \tag{2.1}$$

where the model parameters consist of a weight vector $\boldsymbol{w} \in \mathbb{R}^d$ and a bias $b \in \mathbb{R}$. The fully connected, feed-forward neural network is represented by the following set of equations using $\hat{y} \in \mathbb{R}$ and $\boldsymbol{x} \in \mathbb{R}^d$:

$$\begin{aligned} \text{Input layer:} \quad & \boldsymbol{z}^{(1)} = \boldsymbol{x} \\ \text{Hidden layer(s):} \quad & \boldsymbol{z}^{(l+1)} = a(W^{(l)}\boldsymbol{z}^{(l)} + \boldsymbol{b}^{(l)}), \quad l = 1, \ldots, L \\ \text{Output layer:} \quad & \hat{y} = W^{(L+1)}\boldsymbol{z}^{(L+1)} + \boldsymbol{b}^{(L+1)}, \end{aligned} \tag{2.2}$$

where L is the number of layers in the network and the parameters consists of the weights $\boldsymbol{W}$ and biases $\boldsymbol{b}$ on each layer. The $a : \mathbb{R}^d \to \mathbb{R}^d$ is the activation function, for instance, the rectified linear unit (ReLU) $a(\boldsymbol{z}) := \max(0, \boldsymbol{z})$, where the max operator is applied element-wise for each element in $\boldsymbol{z}$. Observe, with no hidden layers in the network, the model is equal to the linear regression model.

In time with increased computational power and process sensor availability, data-driven modeling has gained attention and is becoming a popular methodology in many process industries (Cherkassky and Mulier, 2007). Due to the models' generally high flexibility, arbitrarily complex, or unknown, physical phenomena can be captured as long as these are reflected in the available data. This is highly advantageous for complex processes where the physical behavior is not completely understood. Furthermore, the development and maintenance cost of data-driven models are often lower than for mechanistic models (Solle et al., 2016). Additionally, after development, the computational burden is typically low and fixed, which makes the models suitable for utilization in real-time applications. Hence, data-driven modeling is a powerful tool that has provided high accuracy models in many cases and has even surpassed human abilities in some application areas (Liu et al., 2019).

Nevertheless, some of the disadvantages of this model type hamper the method of becoming an industry standard. For instance, the inherent black-box nature causes a lack of explainability, which in turn can result in model distrust (Willard et al., 2020). Furthermore, the model is generally only valid in the domain of the data it has been exposed to. They are data-hungry and sensitive to the quality of the data. If not careful, the models can adapt to undesired patterns, such as noise, and can struggle with extrapolation to unseen process conditions (Prada et al., 2018). The aforementioned is also a cause of the common low scientific consistency. Naturally, the advantages and disadvantages of data-driven models vary between model types. Simple regression models have low flexibility, yet, typically higher explainability, and vice versa (Zendehboudi et al., 2018).

2.1.3 Hybrid modeling

Hybrid models attempt to combine the advantages of the mechanistic and data-driven models while diminishing their disadvantages, in a sense, combining the best of both worlds. In particular, there is a desire to create models with high explainability and scientific consistency while retaining the flexibility to adapt to complex or unknown physical phenomena. Although some of the first examples of hybrid modeling appeared in the 1990s (Psichogios and Ungar, 1992; Kramer et al., 1992; Su et al., 1992; Johansen and Foss, 1992), the field of research did not escalate until recent years (Karpatne et al., 2017), likely due to the evolution of data-driven modeling methodologies. Today, some claim that hybrid modeling is the key to success in the process industry (Qin and Chiang, 2019; Willard et al., 2020).

The space of possible combinations of mechanistic and data-driven models is infinite and can be placed on a gray-scale from mechanistic to data-driven models, see Figure 2.2. Approaching hybrid modeling from either side of the scale, the left side can be thought of as enhancing mechanistic models with ML and the right side as guiding data-driven models with physics (Willard et al., 2020).

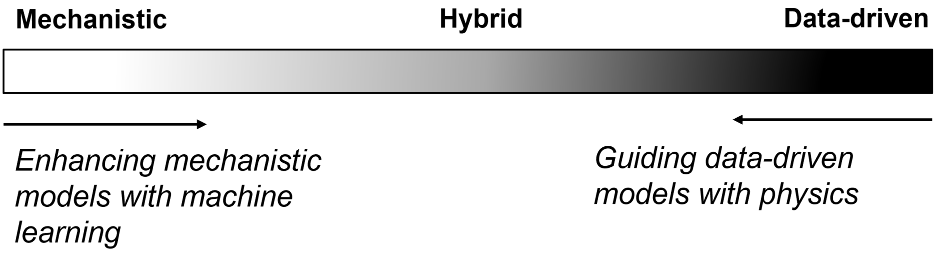
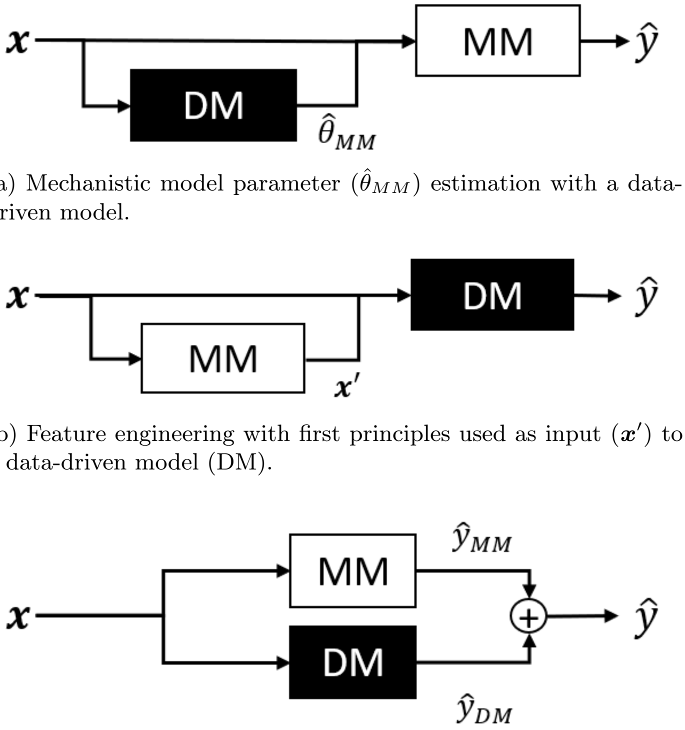

Mechanistic
Hybrid
Data-driven
Enhancing mechanistic models with machine learning
Guiding data-driven models with physics

Figure 2.2: Gray-scale of hybrid models ranging from mechanistic to data-driven models.

On the left side of the gray-scale, the models are based on first-principles equations but utilize data-driven techniques for different purposes. For instance, certain parameters in the mechanistic model can be estimated using a data-driven model, see Figure 2.3a. An example of this type of model can be found in Psichogios and Ungar (1992), where the growth rate of a fed-batch bioreactor is modeled with a neural network. Another example is a model where the process-model mismatch of a mechanistic model is attempted captured with a data-driven model. For instance, the model in Bikmukhametov and Jäschke (2020a) where the error of a mechanistic model for multiphase flow rates is captured with a neural network.

On the right side of the gray-scale in Figure 2.2, the models are based on data-driven solutions but use physics in some form to improve the predictions. Popular naming conventions for these types of models are physics-informed ML, physics-guided ML, physics-aware AI, and theory-guided data science (Willard et al., 2020; Karpatne et al., 2017). One example is a data-driven model with feature engineering. Here, additional inputs to the data-driven model are designed using first principles. Figure 2.3b illustrates this model. An example of utilization is found in Yin et al. (2021) where a convolutional neural network with feature engineering is used to classify eye-tracking data. Another example is a data-driven model with a physics-guided loss function, where the deviation between known physical process behavior and the model output is penalized in the optimization problem. In Pukrittayakamee et al. (2009), a physics-guided loss function is used to train a neural network for a potential-energy surface and corresponding force fields.



(a) Mechanistic model parameter ($\hat{\theta}_{MM}$) estimation with a data-driven model.

(b) Feature engineering with first principles used as input ($\boldsymbol{x}'$) to a data-driven model (DM).

(c) Ensemble model combining a mechanistic ($\hat{y}_{MM}$) and data-driven ($\hat{y}_{DM}$) model output.

Figure 2.3: Hybrid model types. Abbreviations used are mechanistic model (MM) and data-driven model (DM). a) Mechanistic model parameter estimation with a data-driven model, b) feature engineering with a first-principles as input to a data-driven model c) Ensemble model combining a mechanistic and data-driven model. Model inputs are $\boldsymbol{x}$, and the output is $\hat{y}$.

Examples of models that can be placed in the middle of the scale are ensemble models. Here, at least two types of models are developed for the same process. The output of the ensemble model is a more or less intelligent combination of its sub-models, for instance, a weighted average. Figure 2.3c illustrates this. In Baraldi et al. (2014) an ensemble model combining mechanistic and data-driven models are used to predict the degradation of choke valves in offshore oil platforms.

Although hybrid modeling seems promising, several pitfalls are imaginable. Instead of exploiting the best of both worlds, the model can easily become the worst of both worlds. Therefore, a pertinent integration of the mechanistic and data-driven component is essential and is not a trivial exercise. In Sansana et al. (2021), advice on how to choose an appropriate hybrid model structure is given. These are strongly influenced by the available mechanistic model for the process. For instance, if the mechanistic model yields a high process-model mismatch, a structure on the right side of the gray-scale should be chosen, as it increases model flexibility. On the other hand, if the mechanistic model offers a behavior close to the process with few simplifications needed, a hybrid model structure on the left side of the gray-scale should be chosen, mainly because of the generally higher extrapolation power and scientific consistency of such models.

2.2 Model learning

Model learning is the task of utilizing data to aid in mathematical modeling and the estimation of parameters in these models (Beck and Arnold, 1977). Model learning is also referred to as system identification. The act of estimating model parameters using data is also referred to as parameter estimation or model training (Hastie et al., 2009).

Consider a stochastic process $\mathcal{P}$ to generate a stream of observations $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_t, y_t), \ldots\}$, where $\boldsymbol{x}_t \in \mathbb{R}^d$ are the process conditions and $y_t \in \mathbb{R}$ is the desired process quantity at time t. In general, the set S can be thought of as a realization of $\mathcal{P}$ governed by a generative model (Oliveira et al., 2021):

$$p_t(\boldsymbol{x}, y) = p_t(y \mid \boldsymbol{x}) p_t(\boldsymbol{x}). \tag{2.3}$$

In (2.3), $p_t(\boldsymbol{x})$ is the marginal distribution of the process conditions, and $p_t(y \mid \boldsymbol{x})$ is the conditional distribution of the desired process quantity. The index t indicates that the distributions can be time-variant.

A common approach to process modeling is to use an inductive method to learn a steady-state approximation of the true and generally unknown conditional distribution $p_t(y \mid \boldsymbol{x})$ using a fixed dataset of $N_\mathcal{D}$ historical observations:

$\mathcal{D} = \{(\boldsymbol{x}_t, y_t)\}_{t=1}^{N_\mathcal{D}}$. Although dynamic models are often better-suited to represent the behavior of nonstationary processes, they are generally of higher computational complexity and have higher development and maintenance cost (Granero-Belinchón et al., 2019). Therefore, dynamic models can be challenging to utilize in real-time applications. Furthermore, many real processes are slowly varying, making the steady-state assumption reasonable under normal operation. In the remainder of this section, steady-state model learning will be the focus. Under steady-state assumptions, the generative model in (2.3) is time-invariant: $p(\boldsymbol{x}, y) = p(y \mid \boldsymbol{x})p(\boldsymbol{x})$. Naturally, theory regarding parameter estimation of dynamic models exists in literature (Schittkowski, 2002). Some popular approaches are the Kalman Filter (Kalman, 1960) and the moving horizon estimator.

A typical form of the steady-state approximation of the conditional distribution $p(y \mid \boldsymbol{x})$ is

$$\hat{y}_t = f_{\boldsymbol{\theta}}(\boldsymbol{x}_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2), \tag{2.4}$$

where $f_{\boldsymbol{\theta}}$ is the parametric model of the mean, $\boldsymbol{\theta}$ are the model parameters, and ϵ_t is measurement noise with zero mean and variance σ_ϵ^2. The (2.4) is steady-state because $\hat{y}_t$ is conditioned on $\boldsymbol{x}_t$ for a given t and not at previous time-steps, and $\boldsymbol{\theta}$ and σ_ϵ^2 are time-invariant. For (2.4), the assumption of independent and identically distributed (i.i.d) variables is used (Hastie et al., 2009). Furthermore, (2.4) assumes homoscedastic measurement noise. This means that the noise is independent of the signal being measured (Woodward et al., 1998). The opposite is heteroscedasticity. For instance, define $\bar{y}_t$ as the true value of the desired quantity being measured. A heteroscedastic noise term would be a function of the true value: $\epsilon_t(\bar{y}_t)$, whereas homoscedastic noise is not, as defined in (2.4).

Typically, a model with a high generalization power is desired (Goodfellow et al., 2016). With high generalization power, the model yields a high prediction accuracy on previously unobserved process conditions that are not represented in the training data $\mathcal{D}$. For models with high flexibility, this can be nontrivial, especially if the data is noisy or exhibits unusual undesired patterns. If the model adapts to undesired data patterns, it is said to be overfitted. Take Figure 2.4 as an example. Let $x_t = 0, ..., 3\pi$. The true underlying process is a sine-wave. However, the device measuring the sine-wave is corrupted with normally distributed measurement noise. Mathematically, the true conditional distribution of the process generating the observations in Figure 2.4 is given by

$$y_t = \sin(x_t) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 0.5^2). \tag{2.5}$$

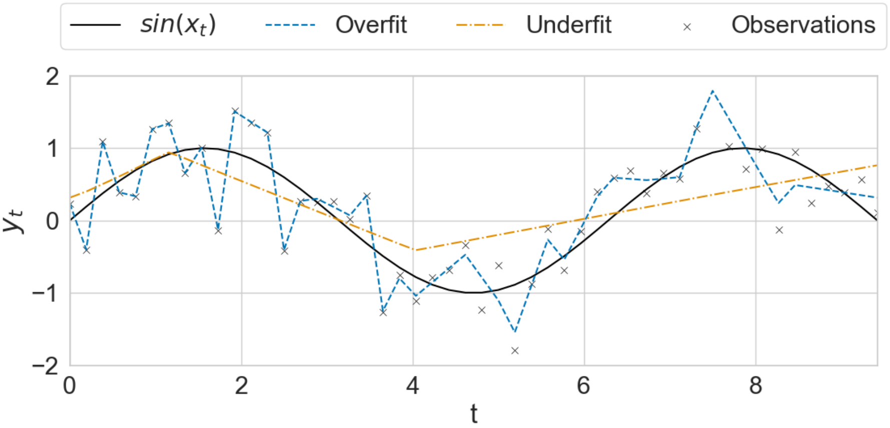


Figure 2.4: Example of modeling a sine-wave. Two identical data-driven models with high flexibility have been used to approximate the true process. The blue curve is the output from a model that is intentionally overfitted to the available data, while the orange curve is the output from a model that is intentionally underfitted.

Two identical steady-state data-driven models $f_{\boldsymbol{\theta}}$ are trained to capture (2.5), see Figure 2.4. The models are neural networks with five hidden layers and 500 nodes in each layer, yielding models with high flexibility. The two models give different outputs because the learning algorithm used to train the two models is different. The model yielding the blue curve is intentionally overfitted to the historical data. This can be observed as the model has adapted not only to the sine-wave but also to the measurement noise. The model yielding the orange curve is intentionally underfitted. Therefore, the model is barely capturing the wavy behavior of the sine. The details around the learning algorithm will be described in the following sections.

2.2.1 The parameter estimation problem

The parameter estimation problem can be phrased as an optimization problem

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} J, \tag{2.6}$$

where $\hat{\boldsymbol{\theta}} \in \mathbb{R}^{P_\theta}$ are the parameter estimates, and J is the objective function that generally describes a metric of model fit to the available process data.

In the data-driven modeling domain, J is often called a loss function and is typically minimized instead of maximized (Hastie et al., 2009). Maximization can easily be converted to minimization by minimizing $-J$. An alternative to phrasing the problem as an optimization problem is the utilization of Monte Carlo sampling (Luengo et al., 2020). However, such techniques will not be described further.

Approaching model learning from a Bayesian perspective, the prior probability distribution of the model parameters $p(\boldsymbol{\theta})$ can be updated to a posterior parameter distribution after observing data. The posterior parameter distribution is given by

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D})}. \tag{2.7}$$

The $p(\mathcal{D} \mid \boldsymbol{\theta})$ is called the likelihood function and is given by (2.4) and $p(\mathcal{D})$ is called the evidence. Conditional models like (2.4) are commonly trained with maximum likelihood estimation (MLE) or maximum a posteriori (MAP) estimation. In MLE, the parameters are found by maximizing the mode of the likelihood function, whereas in MAP estimation the mode of the posterior parameter distribution is maximized. Bayes law, together with the MAP and MLE parameter estimates, are visualized in Figure 2.5. The MLE problem can be derived as

$$\hat{\boldsymbol{\theta}}_{MLE} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{D} \mid \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \log p(\mathcal{D} \mid \boldsymbol{\theta}). \tag{2.8}$$

By using the i.i.d. assumption, which states that the observations in $\mathcal{D}$ are independent of each other and identically distributed, the log-likelihood function of the model in (2.4) can be written as

$$\begin{aligned} \log p(\mathcal{D} \mid \boldsymbol{\theta}) &= \log \prod_{t=1}^{N_{\mathcal{D}}} p(y_t \mid \boldsymbol{x}_t, \boldsymbol{\theta}) = \sum_{t=1}^{N_{\mathcal{D}}} \log p(y_t \mid \boldsymbol{x}_t, \boldsymbol{\theta}), \\ &= \sum_{t=1}^{N_{\mathcal{D}}} \log \left[\frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} e^{\frac{(y_t - f_{\boldsymbol{\theta}}(\boldsymbol{x}_t))^2}{2\sigma_\epsilon^2}} \right] \\ &= -\sum_{t=1}^{N_{\mathcal{D}}} \frac{1}{2\sigma_\epsilon^2} \left(y_t - f_{\boldsymbol{\theta}}(\boldsymbol{x}_t)\right)^2 - N_{\mathcal{D}} \log \sqrt{2\pi\sigma_\epsilon^2}, \end{aligned} \tag{2.9}$$

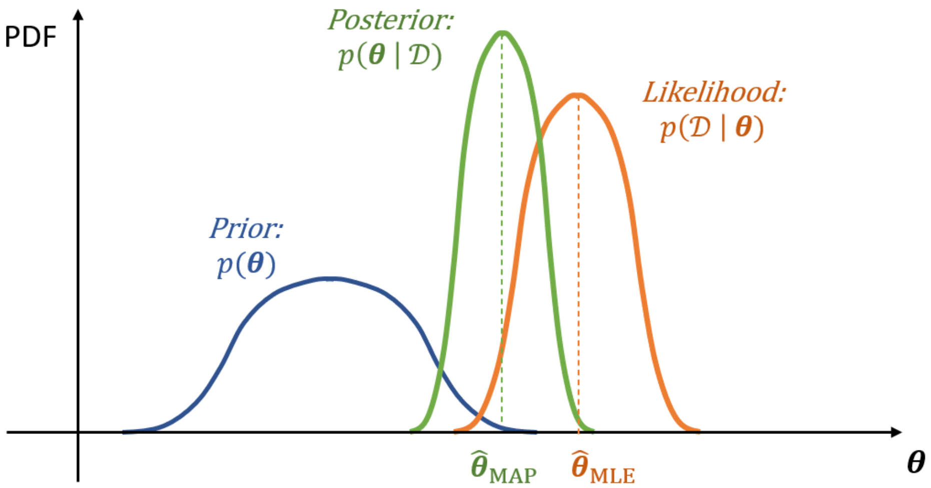

PDF
Posterior:
$p(\boldsymbol{\theta} \mid \mathcal{D})$
Likelihood:
$p(\mathcal{D} \mid \boldsymbol{\theta})$
Prior:
$p(\boldsymbol{\theta})$
$\hat{\boldsymbol{\theta}}_{\text{MAP}}$
$\hat{\boldsymbol{\theta}}_{\text{MLE}}$
$\boldsymbol{\theta}$

Figure 2.5: Visualization of Bayes law. The prior parameter distribution is updated to a posterior distribution with the likelihood of the model. In maximum likelihood estimation, the estimated parameters are the mode of the likelihood function ($\hat{\boldsymbol{\theta}}_{\text{MLE}}$). Using maximum a posteriori estimation, the estimated parameters are the mode of the posterior distribution ($\hat{\boldsymbol{\theta}}_{\text{MAP}}$).

resulting in the estimation problem

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{MLE} &= \arg\max - \sum_{t=1}^{N_{\mathcal{D}}} \frac{1}{2\sigma_\epsilon^2} \left(y_t - f_{\boldsymbol{\theta}}(\boldsymbol{x}_t)\right)^2 - \underbrace{N_{\mathcal{D}} \log \sqrt{2\pi\sigma_\epsilon^2}}_{\text{constant}}, \\ &= \arg\min \sum_{t=1}^{N_{\mathcal{D}}} \frac{1}{2\sigma_\epsilon^2} \left(y_t - f_{\boldsymbol{\theta}}(\boldsymbol{x}_t)\right)^2 . \end{aligned} \tag{2.10}$$

If the estimation problem is divided by $N_{\mathcal{D}}$ and by assuming a constant noise level σ_ϵ = const., the MLE problem is equal to the nonlinear least squares (NLS) estimation problem. The NLS is a common approach in both the mechanistic and data-driven modeling domains. From (2.10), it is seen that the only concern of the MLE is to minimize the squared deviation between measurements and predictions. Hence, the model can be easily overfitted to undesired attributes of the data, as for one of the models in Figure 2.4. To counteract this effect, the models can be trained with maximum a posteriori (MAP) estimation

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{MAP} &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathcal{D}) = \arg\max_{\boldsymbol{\theta}} \frac{p(\mathcal{D} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})} \\ &= \arg\max_{\boldsymbol{\theta}} \log p(\mathcal{D} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \underbrace{\log p(\mathcal{D})}_{\text{constant}}.\end{aligned} \tag{2.11}$$

If the priors on the parameters are assumed to follow a Gaussian distribution $\theta_i \sim \mathcal{N}(\mu_i, \sigma_i^2), i = 1...P_\theta$, the log-prior parameter distribution is given as

$$\log p(\boldsymbol{\theta}) = \log \prod_{i=1}^{P_\theta} p(\boldsymbol{\theta}_i) = -\sum_{i=1}^{P_\theta} \frac{1}{2\sigma_i^2} (\theta_i - \mu_i)^2 - P_\theta \log \sqrt{2\pi\sigma_i^2}. \tag{2.12}$$

Inserting (2.9) and (2.12) into (2.11), the MAP estimation problem becomes

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{MAP} &= \arg\max_{\boldsymbol{\theta}} \log p(\mathcal{D} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}), \\ &= \arg\max_{\boldsymbol{\theta}} -\sum_{t=1}^{N_\mathcal{D}} \frac{1}{2\sigma_\epsilon^2} (y_t - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_t))^2 - \underbrace{N_\mathcal{D} \log \sqrt{2\pi\sigma_\epsilon^2}}_{\text{constant}} \\ &\quad -\sum_{i=1}^{P_\theta} \frac{1}{2\sigma_i^2} (\theta_i - \mu_i)^2 - \underbrace{P_\theta \log \sqrt{2\pi\sigma_i^2}}_{\text{constant}}, \\ &= \arg\min_{\boldsymbol{\theta}} \sum_{t=1}^{N_\mathcal{D}} \frac{1}{\sigma_\epsilon^2} (y_t - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_t))^2 + \sum_{i=1}^{P_\theta} \frac{1}{\sigma_i^2} (\theta_i - \mu_i)^2 .\end{aligned} \tag{2.13}$$

Observe, MAP estimation is a trade-off between minimizing the squared errors and parameter deviation away from its respective prior mean value. If (2.13) is multiplied by the factor $\sigma_\epsilon^2/N_\mathcal{D}$ and the prior parameter distribution is equal for all parameters with mean value $\mu_i = 0$ and $\sigma_i = \sigma$, (2.13) is equal to a the NLS with ℓ_2-regularization, a common approach in the data-driven modeling domain. With a mean parameter value equal to zero, the ℓ_2-regularization penalizes large parameter values, preventing a data-driven model from adapting to arbitrarily patterns in the data, in other words, preventing overfitting. The regularization factor, which from the MAP estimation problem is given by $\lambda = \sigma_\epsilon^2/N_\mathcal{D}\sigma^2$, is typically used as a tuning constant to determine the degree of regularization. The neural networks in Figure 2.4 are trained with NLS estimation. For the overfitted model, parameter regularization is excluded, whereas for the underfitted model, $\lambda = 0.1$. The effect of ℓ_2-regularization on the optimal parameter value is illustrated for the two-dimensional space in Figure 2.6. The solid ellipses illustrate the contours of the NLS where the optimum is given at $\hat{\boldsymbol{\theta}}_{NLS}$. The dotted ellipses illustrate the contours of the ℓ_2-regularization of the parameters, where the optimum value for the NLS with regularization $\hat{\boldsymbol{\theta}}_{NLS,\ell_2}$ will depend on the size of the regularization factor λ. For large λ, the optimum will be drawn towards the center of the ellipses.

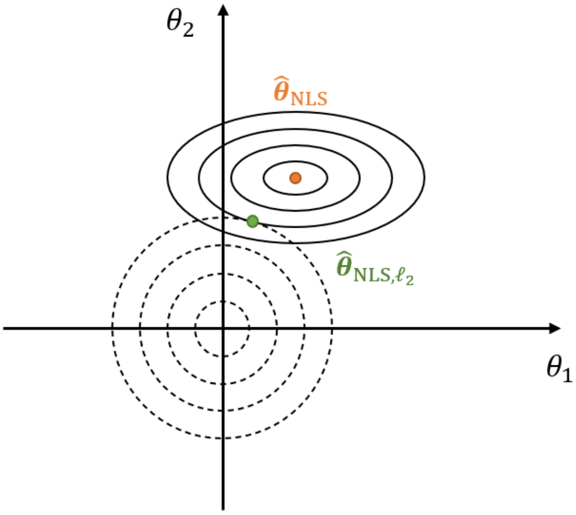

θ_2
$\hat{\boldsymbol{\theta}}_{\mathrm{NLS}}$
$\hat{\boldsymbol{\theta}}_{\mathrm{NLS},\ell_2}$
θ_1

Figure 2.6: Visualization of the effect of ℓ_2-regularization in 2D-space on the nonlinear least squares estimation problem. The solid ellipses represent the contours of the NLS problem with the optimum in the center. The dotted ellipses represent the contours of ℓ_2-regularization, where the optimal value for NLS with regularization will change depending on the regularization factor λ.

A disadvantage with the MLE and MAP as described above is that the result of the estimation problem is point estimates of the parameters only, disregarding uncertainty. Naturally, there exist methods that estimate the full posterior parameter distribution and not just the mode. Examples are Markov Chain Monte Carlo methods (MCMC) or Variational Inference (VI) (Blei et al., 2017). These methods are of higher computational complexity than MAP estimation, and a detailed description will not be given here.

2.2.2 Solving the estimation problem

Regardless of the chosen type of estimation problem, it must be solved. There exist many methods to solve (2.6). For linear, deterministic models, the exact solution can often be calculated. Consider the linear model

$$\hat{y}_t = \boldsymbol{x}_t^T \boldsymbol{\theta}, \tag{2.14}$$

for which maximum likelihood will be used to estimate the parameters. Collect all measured inputs and outputs from $\mathcal{D}$ in the design matrix $\boldsymbol{X} \in \mathbb{R}^{N_\mathcal{D} \times P_\theta}$ and the vector $\boldsymbol{y} \in \mathbb{R}^{N_\mathcal{D}}$, respectively. The MLE for the linear model becomes

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\right)^T \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\right). \tag{2.15}$$

The solution to (2.15) is given by

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\right)^T \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\right) = 0, \\ \implies &2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\theta} - 2\boldsymbol{X}\boldsymbol{y} = 0, \\ \implies &\hat{\boldsymbol{\theta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}, \end{aligned} \tag{2.16}$$

and a unique solution exists if $\boldsymbol{X}^T\boldsymbol{X}$ has full column rank (Nocedal and Wright, 2006).

On the other hand, for highly complex, highly nonlinear models, exact solutions to (2.6) are challenging to obtain. Therefore, iterative, gradient-based, numerical optimization algorithms are commonly used (Bishop, 2006). The general formulation for such algorithms is given by (Bengio, 2012)

$$\hat{\boldsymbol{\theta}}^{(k+1)} = \hat{\boldsymbol{\theta}}^{(k)} - \gamma^{(k)}\mathcal{M}(\mathcal{B}^{(k)}, \hat{\boldsymbol{\theta}}^{(k)}),\ k = 1, .., E \tag{2.17}$$

where E is the number of iterations or steps taken towards the optimal value, γ is the learning rate or step-size, $\mathcal{M}$ is the set of equations calculating the step direction, and $\mathcal{B}$ is a minibatch consisting of a set of observations extracted from $\mathcal{D}$. Any parameter of the learning algorithm that is not included in $\boldsymbol{\theta}$ is called a hyperparameter, for instance, γ, E, and $|\mathcal{B}|$. Several of the hyperparameters are essential to obtain a good model fit to data. For instance, if the step-size is too large, the learning algorithm can diverge, and an increase in the average error is typically observed (Bengio, 2012). If the step-size is too small, the learning problem can fail to converge in the given iterations, resulting in underfitting. A possible solution to determine an appropriate learning rate is to introduce a learning rate scheduler that adapts the learning rate at each iteration according to a criteria (Bengio, 2012). The number of iterations E can also significantly influence the result of model learning. A too large or small E can result in the model being overfitted or underfitted, respectively. A typical approach to select E is the regularization method early stopping (Goodfellow et al., 2016). Early stopping monitors the error on a subset of the historical data called the validation data and terminates the training if the model error on the validation data increases significantly compared to the error on the remaining historical data. Furthermore, the choice of method to calculate the step-direction $\mathcal{M}$ can affect the convergence of the learning algorithm.

In the ML domain, first-order optimization algorithms are commonly applied (Goodfellow et al., 2016). In these methods, the step direction is calculated by using first-order derivative information of the objective function. For instance, in minibatch gradient descent (GD)

$$\mathcal{M} = \nabla_{\boldsymbol{\theta}} J(\mathcal{B}^{(k)}, \hat{\boldsymbol{\theta}}^{(k)}), \tag{2.18}$$

Stochastic gradient descent (SGD) is applied if the minibatch consists of only one observation. Minibatch GD and SGD are among the most popular methods in ML for solving (2.6) because of their high computational efficiency in large-scale problems, such as text classification (Bottou et al., 2018). A disadvantage is that they can struggle to converge for highly nonlinear domains with several local minima. An example of this is illustrated in Figure 2.7. The objective function J has several local minima in parameter space. If the parameters are initialized at $\boldsymbol{\theta}_0$, GD and SGD can end up in the wrong minimum, dependent on the hyperparameters of the learning algorithm. Therefore, other gradient descent methods have been developed to counteract this effect, for instance, Adam, which utilizes momentum (Kingma and Ba, 2015). As an analogy, consider a ball rolling down the hill from the starting point $\hat{\boldsymbol{\theta}}^{(0)}$ in Figure 2.7. Using the momentum the ball has at the bottom of the first local minimum ($\hat{\boldsymbol{\theta}}_{\text{(S)GD}}$), it can continue past the hilltop and continue towards the second local minimum ($\hat{\boldsymbol{\theta}}_{\text{Adam}}$). Other popular gradient descent methods in the ML domain are AdaGrad and Adadelta, where both utilize adaptive learning rates (Tan and Lim, 2019).

Another approach in addressing nonlinear domains is by introducing second-order derivative information through second-order optimization algorithms (Bottou et al., 2018). One example is the Newton method (Nocedal and Wright, 2006)

$$\mathcal{M} = \left[\nabla^2_{\boldsymbol{\theta}} J(\mathcal{B}^{(k)}, \hat{\boldsymbol{\theta}})\right]^{-1} \nabla_{\boldsymbol{\theta}} J(\mathcal{B}^{(k)}, \hat{\boldsymbol{\theta}}). \tag{2.19}$$

where $\nabla^2_{\boldsymbol{\theta}} J$ is the Hessian matrix. However, the Newton method is rarely used for complex models due to the computational burden of calculating the Hessian, and particularly the inverse of the Hessian (Goodfellow et al., 2016). On the other hand, there exist algorithms that approximate the Hessian matrix

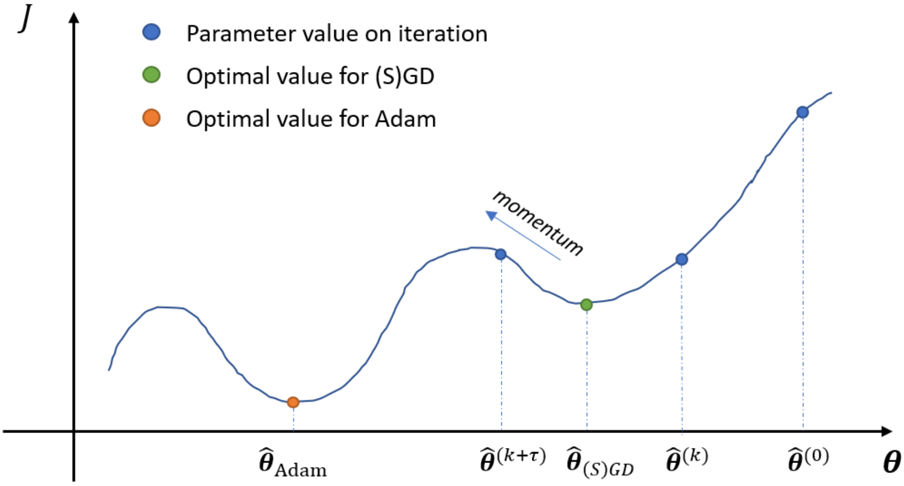

J
● Parameter value on iteration
● Optimal value for (S)GD
● Optimal value for Adam
momentum
$\hat{\boldsymbol{\theta}}_{\text{Adam}}$
$\hat{\boldsymbol{\theta}}^{(k+\tau)}$
$\hat{\boldsymbol{\theta}}_{(S)GD}$
$\hat{\boldsymbol{\theta}}^{(k)}$
$\hat{\boldsymbol{\theta}}^{(0)}$
$\boldsymbol{\theta}$

Figure 2.7: An illustration of the possible convergence of (S)GD and Adam. The objective function is nonlinear with several local minima. Starting the learning algorithm at $\hat{\boldsymbol{\theta}}^0$, (S)GD is likely to converge to the first local minimum, whereas Adam, which utilizes momentum, likely continues to the next local minimum.

with first-order derivative information to retain curvature information. Examples of such methods are the conjugate gradient method, the Gauss-Newton method, the Levenberg-Marquardt method, and the BFGS method (Nocedal and Wright, 2006).

For all the model types described in Section 2.1, the model parameters are uncertain and can be estimated using historical data. As mentioned in 2.1.1, mechanistic model parameters typically have good prior values, and parameter estimation is not a requirement. On the other hand, obtaining a good model fit to data, parameter estimation is often a necessity. For data-driven models, model learning is essential to obtain acceptable models as the parameters are typically initialized at random values. Generally, the latter also applies to hybrid models as they can have several data-driven model parameters. Naturally, the number of data-driven and mechanistic model parameters will depend on the hybrid model structure. Nevertheless, the estimation problem still needs to handle that the model has two types of parameters with different interpretations and with a varying degree of uncertainty on their prior values.

2.3 Virtual flow metering

2.3.1 The petroleum production system

A petroleum production system is the assembly of wells, pipes, valves, chokes, pumps, separators, and transportation pipelines, which task is to transport petroleum fluids from the reservoir to the processing facilities (Guo et al., 2007). A well-equipped, offshore petroleum production system is illustrated with a simplified drawing in Figure 2.8. Three wells produce a mixture of oil, gas, and water from the reservoir, where the amount of each phase will depend on the type of reservoir. The multiphase flow rate through individual wells is controlled by the production chokes located in the wellheads of the wells. Upstream and downstream the production choke, the pressure (p) and temperature (T) are measured. The pressure and temperature are also measured down-hole, the closest point to the reservoir. Upstream the production choke, a multiphase flow meter (MPFM) measures the multiphase flow rate through the choke. The production from the wells is commingled at the manifold and sent to the production separator for separation into the three phases. Alternatively, the flow can be routed through the test separator using the routing valves.

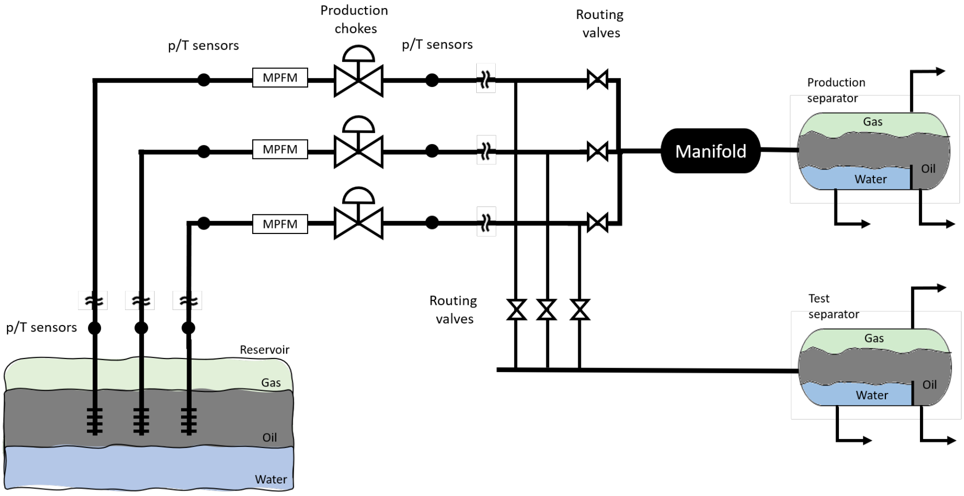


Figure 2.8: A simplified, offshore petroleum production system. Three wells produce a mixture of oil, gas, and water.

The production system is nonstationary with time-dependent process conditions and properties (Guo et al., 2007). For instance, in a crude oil reservoir, the fraction of oil is high at the beginning of the asset's life. In time with reservoir depletion, the fraction of gas, water, or both, typically increases. The production from large fields goes through two phases: the plateau and decline phase. In the plateau phase, the production is approximately constant, maintained by gradually opening the production choke valves in the wells. This will counteract the pressure declination in the reservoir that occurs as a consequence of depletion (Foss et al., 2018). In the decline phase, the production chokes are generally fully open, except for wells that produce to the same pipeline, where some wells must be choked back slightly to maintain a pressure balance. Therefore, the production will decline in time with the pressure declination in the reservoir. The nonstationarity of the production system causes the multiphase flow rate through the system to have a dynamic nature with both fast and slow transients. The fast transients are caused by control changes such as the opening of the choke valve and occur in the time range of minutes to hours (Foss et al., 2018). The slow transients are caused by the reservoir depletion (Jansen, 2015) and occur in the time range months to years, dependent on the size of the reservoir.

2.3.2 Flow rate metering

Several efforts in petroleum engineering require knowledge about the multiphase flow rate, for example, flow assurance, situational awareness, production optimization, and reservoir management (Foss et al., 2018). There are several ways to obtain information about the multiphase flow rate. The traditional way of measuring the flow rates is through well-testing, either by using a test separator as illustrated in Figure 2.8 or through deduction testing, where the production from one well is shut down, and the difference in total production at the manifold is compared before and after the shutdown. Well-testing typically provides high accuracy measurements with an error as low as 0.25% and 1% for oil and gas rates, respectively. However, due to the required production stabilization time, the well-tests are intermittent and infrequent, often with less than one new well-test per month (Monteiro, Duque, et al., 2020). Furthermore, the test separators are large and require a sufficient amount of space on the production platform, and deduction testing will result in economical loss as part of the production is shut down.

Another option is the MPFMs as illustrated in Figure 2.8. These are physical devices that offer continuous predictions of flow rates. Continuous measurements are useful for many of the above-mentioned engineering problems. Nevertheless, MPFMs are prone to drift over time and call upon intervention with failure (Falcone et al., 2013). Intervention requires shutdown of wells and results in an economical loss. Moreover, compared to well-testing, the accuracy of MPFMs is lower with an error of approximately 5% (Thorn et al., 2013).

A third alternative is a VFM, a soft-sensor enabling continuous multiphase flow rate predictions by exploiting mathematical models and already existing measurements (Varyan et al., 2015). Virtual flow metering is a non-intervening technology as it avoids the need to install additional physical devices and is, therefore, easier maintained. Today, many petroleum assets use some form of VFM technology in addition to physical flow metering, either as a standalone system or as a backup to an MPFM. There are two main applications of a VFM: 1) real-time predictions of the flow rate and 2) prediction of missing historical measurements, either due to failure or lack of physical devices. The VFM can be developed for any desired component in the production system. For instance, for the production choke in wells to measure the individual flow rates or for the manifold to measure the commingled flow rate.

2.3.3 Challenges of virtual flow metering

Developing a mathematical model to predict the multiphase flow rate at strategic locations in the production system is not trivial as there are several sources for uncertainty, both in the process physics and in the available data. The first is a particular challenge for mechanistic modeling, while the latter is disadvantageous for data-driven modeling.

Firstly, a tremendous effort has been invested in describing the physics of the production system in the last decades. Nevertheless, the multiphase flow rates are challenging to model accurately with first-principles due to its transient and complex nature and uncertain subsurface properties (Guo et al., 2007). Furthermore, both the governing equations and fluid properties can change with process conditions and flow regime (Thorn et al., 2013). Therefore, simplifications and assumptions are necessary to solve the process equations in real-time applications. Consequently, the model will never be a perfect realization of the true process, and there will be uncertainties related to both the model structure and parameters. This type of uncertainty is called epistemic uncertainty (Hüllermeier and Waegeman, 2021). Due to epistemic uncertainty, it is not uncommon for mechanistic VFMs to have process-model mismatch that tends to increase with time. Hence, mechanistic VFMs require a high degree of expertise for developing and demand regular calibration to maintain high prediction accuracy (Bikmukhametov and Jäschke, 2020b).

Secondly, the available process data are typically subject to several disadvantageous attributes. A particular troublesome artifact is the often poor measurement quality with high noise levels, causing imprecision in measurements and systematic errors (Antonelo et al., 2017). Some sensors tend to drift in time, and some may even fall out for longer periods (Antonelo et al., 2009). Imprecision in sensor measurements is referred to as aleatoric uncertainty (Hüllermeier and Waegeman, 2021). Furthermore, available process data from production systems commonly resides in the small data regime, which, according to Mishra and Datta-Gupta (2018), is characterized by low volume, variety, and velocity. A low volume and low velocity of datasets originating from petroleum production systems are due to the often infrequently and intermittently occurring well-tests. Yet, even for assets with continuous measurements, the issue of low variety can remain because of how the production systems are typically operated. For instance, as described above, the production choke valve is gradually opened to counteract the pressure declination in the reservoir. Consequently, a characteristic path through the data space as visualized in Figure 2.9 for two dimensions is typically observed for historical data. The data is taken from a petroleum production well and scaled to the interval $[0, 1]$. The coloring indicates time, where the darkest color is the oldest observation. The path spans a relatively small part of the two-dimensional space and will be even sparser in higher dimensions. Both the volume and variety of the data could be improved with designed experiments. However, such experiments typically have a high expense and are not applied (Rackauckas et al., 2021). These known data challenges have demonstrated a significant impact on the performance of data-driven VFMs.

Due to the complex physical characteristics of multiphase flow and the data challenges, hybrid modeling seems like a promising methodology for the virtual flow metering application. In theory, a hybrid VFM should be able to exploit data-driven models to increase the flexibility of a mechanistic model, thereby reducing process-model mismatch resulting from simplifications. Moreover, the hybrid model should be able to take advantage of physics to avoid adaptation to undesired artifacts in the available process data. In other words, a hybrid model will attempt to extract as much information as possible from physics and data without relying solely on only one of the sources of information. On the other hand, for the same reasons, hybrid models need to manage the uncertainty in both information sources. Therefore, the balance between learning from physics and learning from data becomes an important task for hybrid VFMs. If not careful, the model can easily enhance the worst of both domains

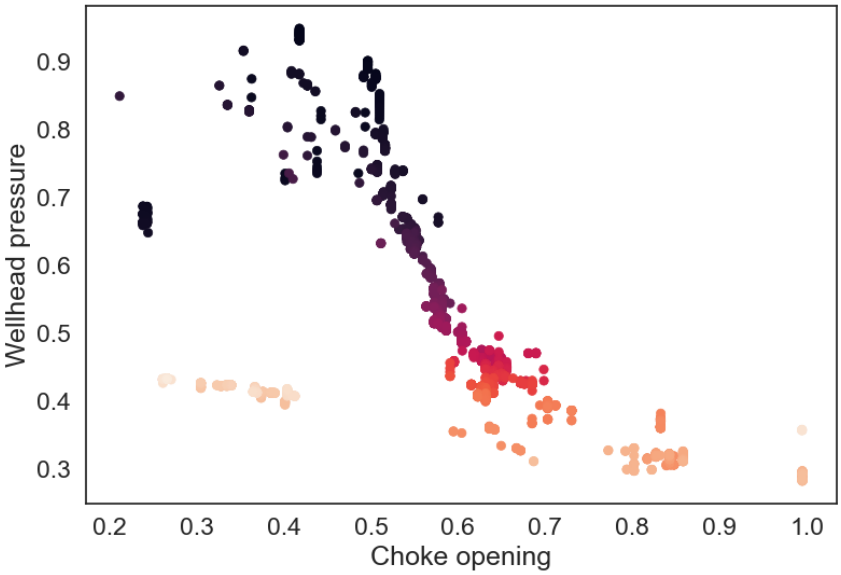


Figure 2.9: A typically observed path through the data space of a real petroleum well, here illustrated with historical data of the choke opening and the pressure in the wellhead. The coloring indicate time where the darkest points are the oldest observations. The data is scaled to reside in the interval $[0, 1]$.

instead of the best.

2.3.4 Virtual flow meters in industry and literature

There exist many examples of VFMs in industry and literature. Among the commercially available VFMs, the leading approach is mechanistic modeling. Some examples are Prosper, ValiPerformance, LedaFlow, FlowManager, and Olga (Amin, 2015). Recently, hybrid (Ruden, 2020) and data-driven (Solution Seeker, 2022) VFMs have emerged on the market.

In the literature, numerous research efforts exist on both mechanistic and data-driven modeling of a VFM. The most common approach in the literature is steady-state models, although several of the commercial VFMs mentioned above offer dynamic as well as steady-state models (Shippen, 2012). One reason for this is likely the inherent complex dynamics of multiphase flow, resulting in a high computational burden, making it challenging to develop dynamical models suitable for utilization in real-time applications. Another reason can be due to the process being slowly varying such that the steady-state assumption can be acceptable for shorter periods in time. An in-depth literature review on mechanistic and data-driven VFMs can be found in Bikmukhametov and Jäschke (2020b). In comparison, the research efforts on hybrid VFMs in the open literature are sparse, although existent. Table 2.2 gives a brief overview of recent work on hybrid VFMs. Acronyms utilized in the table are NN: neural network, DM: data-driven model, MM: mechanistic model. The papers listed are sorted by date of publication. Keep in mind that there can be papers published on hybrid VFMs not covered by the table, for instance, if the authors have used different naming conventions than used in this thesis or have not realized the hybridity of the model.

Table 2.2: Hybrid virtual flow meters in the literature. Acronyms used are NN: neural network, DM: data-driven model, MM: mechanistic model.

Paper	Description
Xu et al. (2011)	Feature engineering in an NN for wet gas metering
Al-Rawahi et al. (2012)	NN to estimate the mixture density of the multiphase flow
Mohammadmoradi et al. (2018)	MM to describe prior knowledge and constraints of a DM
Bikmukhametov and Jäschke (2020a)	Three approaches. 1) Feature engineering in DM. 2) Capture mismatch between MM and process with DM. 3) Ensemble model combining outputs from several models.
Staff et al. (2020)	Feature engineering in DM
Andrade et al. (2022)	Data reconciliation constrained by MM

2.3.5 Examples of mechanistic production choke models

As mentioned in Section 2.3.2, a VFM can be developed for most components in a petroleum asset, for instance, the wellbore and production choke of a single petroleum well, see Figure 2.10. In the work leading up to this thesis, the production choke has been the focus.

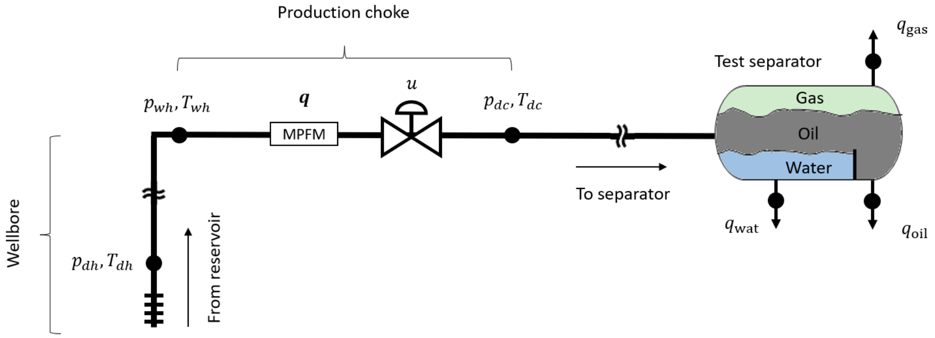


Figure 2.10: A simplified, offshore production system of a single petroleum well along with available sensor measurements for well-equipped wells. The wellbore is located between the down-hole (dh), the closest sensor location to the reservoir, and the wellhead (wh). The production choke is located between the wellhead and the separator, with the downstream choke (dc) as the closest sensor location to the separator.

The production choke is located between the wellhead and the separator in the production system see Figure 2.10. An MPFM measures the phasic flows through the well $\boldsymbol{q} = [q_{\text{oil}}, q_{\text{gas}}, q_{\text{wat}}]$. The total multiphase flow is given by $Q = q_{\text{oil}} + q_{\text{gas}} + q_{\text{wat}}$. Typically, the sensor pairs in the wellhead and downstream the choke are close to the production choke. The choke is a restriction in the pipe, illustrated in Figure 2.11, where the area of the choke at the outlet (A_{dc}) is adjustable with the choke opening u. There are two types of flow

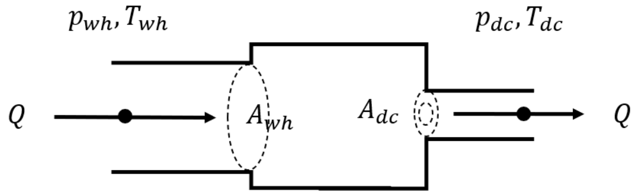


Figure 2.11: A close-up illustration of the production choke. The multiphase fluid flows through a restriction, where the outlet of the choke (A_{dc}) can be adjusted with the choke opening.

behavior in a choke, critical and sub-critical (Guo et al., 2007). When the flow rate is in the sub-critical zone, the rate is determined by the pressure drop across the choke. If it is in the critical zone, the flow has reached a maximum rate through the choke, and a further decrease in the downstream pressure for a constant upstream pressure will not influence the rate, see Figure 2.12. Several mechanistic models exist for the production choke, in a varying scale of complexity in space and time. They are usually developed assuming steady-state, one-dimensional (lumped) flow since increasing the dimensionality of the problem requires a numerical solution of the complex Navier-Stokes equations. These equations are computationally demanding and may not be suitable for use in real-time optimization (Shippen, 2012). There are several well-known choke models in literature and industry (Selmer-Olsen, 1995; Sachdeva et al., 1986; Perkins, 1993; Al-Safran and Kelkar, 2009). Here, two examples of steady-state mechanistic models for the production choke are given. These have been the models used in the work leading up to this thesis.

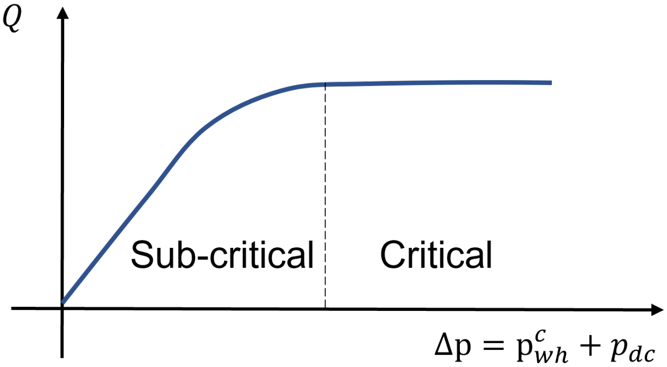


Figure 2.12: Illustration of the sub-critical and critical boundary of the flow through a choke for constant upstream pressure (p_{wh}^{c}) and decreasing downstream pressure, resulting in an increased pressure drop. In the sub-critical zone, the flow will increase for increasing pressure drop, while in the critical zone, the flow has reached a maximum rate and is not influenced by a further reduction in the downstream pressure.

The Kittilsen choke model

The Kittilsen model for multiphase flow through a choke (Kittilsen et al., 2014) is a one-dimensional (lumped) and steady-state model. The following assumptions are applied for the multiphase flow rate:

- no-slip: the gas and liquid travels through the choke with equal velocity,
- incompressible liquid: liquid densities are constant,
- frozen flow: no mass transfers from one phase to another across the choke,
- no gas expansion across the choke,
- thoroughly and homogeneously mixed fluid,
- neglection of momentum effects in the upstream part of the choke.

The equation for the mass flow rate through the choke is given by

$$\dot{m} = N_c C_v(u)\sqrt{Y^2 \rho (p_{wh} - p_{dc})}, \tag{2.20}$$

where $\dot{m}$ is the mass flow rate, N_c is a unit conversion coefficient, C_v is valve flow coefficient, typically a function of the choke opening (u), ρ is the mixture density, and Y^2 is given by

$$Y^2 = \left(1 - \frac{1}{3}\frac{x_{lim}}{x_{TP}}\right)^2 \frac{x_{lim} p_{wh}}{p_{wh} - p_{dc}}, \tag{2.21}$$

where

$$x_{lim} = \min\left(x_P, x_{TP}\right), \tag{2.22}$$

and

$$x_P = \frac{p_{wh} - p_{dc}}{p_{wh}}, \qquad x_{TP} = \frac{p_{wh} - p_{dc}}{p_{wh}}|_{\text{critical}}, \tag{2.23}$$

where x_{TP} is a constant relating the pressure drop to the upstream pressure in the critical flow regime. A rule of thumb is that the downstream to upstream pressure ratio at critical conditions is $p_{dc}/p_{wh} \approx 0.6$ such that $x_{TP} \approx 0.4$ (Jansen, 2015). With the assumption of thoroughly and homogeneously mixed fluid, the mixture density can be described with

$$\frac{1}{\rho} = \frac{\eta_{\text{gas}}}{\rho_{\text{gas}}} + \frac{\eta_{\text{oil}}}{\rho_{\text{oil}}} + \frac{1 - \eta_{\text{gas}} - \eta_{\text{oil}}}{\rho_{\text{wat}}}, \tag{2.24}$$

where $\rho_i, \eta_i, i \in \{\text{gas}, \text{oil}, \text{wat}\}$ are the phasic densities and mass fractions of the fluid. It is assumed that no other fluids exist such that

$$\eta_{\text{gas}} + \eta_{\text{oil}} + \eta_{\text{wat}} = 1. \tag{2.25}$$

The density of gas is described with the real gas law

$$\rho_{\text{gas}} = \frac{M p_{wh}}{Z R T_{wh}}, \tag{2.26}$$

where M is the molar mass of gas, R is the universal gas constant, and Z is the compressibility factor to account for a non-ideal gas. To convert the mass flow rate to a volumetric flow rate

$$Q = \frac{\dot{m}}{\rho_{SC}}, \tag{2.27}$$

where SC stands for standard conditions, typically 1atm and 15°C (International Organization for Standardization, 1996).

The valve flow coefficient is associated with the hydraulic performance of a control valve and is given in [USG min^{-1} $\text{PSI}^{-0.5}$] (Emerson Automation Solutions, 2017, p. 99). The C_v will vary depending on the opening of the choke. It is typical to differentiate between three characteristics of the C_v-curve: quick opening, linear, and equal percentage, see Figure 2.13. In practice, the shape of the curve can deviate from these three.

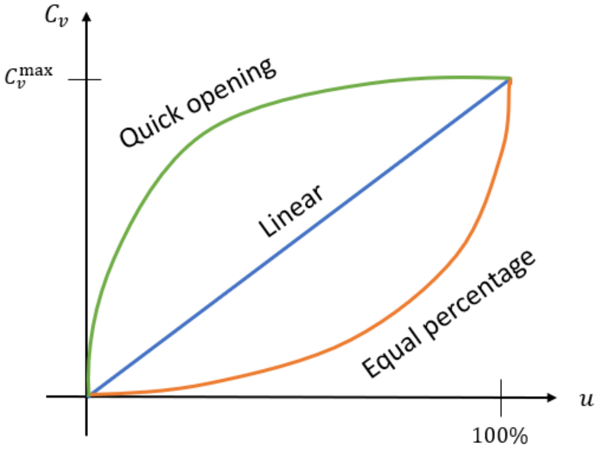


Figure 2.13: Three typical characteristics of the C_v-curve for a production choke.

The Sachdeva choke model

The Sachdeva model is derived from the simplified, one-dimensional, steady-state mass and momentum balance equations (Sachdeva et al., 1986). By further assuming

- no-slip,
- incompressible liquid,
- frozen flow,
- adiabatic gas expansion across the choke: no mass or heat transfers between the fluid and the surroundings,
- thoroughly and homogeneously mixed fluid,
- neglect of momentum effects in the upstream part of the choke,

the equation for the mass flow rate through the choke is derived

$$\dot{m} = C_D A_{dc}(u) \times \sqrt{2\rho_{dc}^2 p_{wh} \left(\frac{\kappa}{\kappa - 1} \eta_{\text{gas}} \left(\frac{1}{\rho_{\text{gas},wh}} - \frac{p_r}{\rho_{\text{gas},dc}} \right) + \left(\frac{\eta_{\text{oil}}}{\rho_{\text{oil}}} + \frac{\eta_{\text{wat}}}{\rho_{\text{wat}}} \right) (1 - p_r) \right)}, \tag{2.28}$$

where p_r is the downstream to upstream pressure ratio, κ is the gas expansion coefficient, and C_D is a discharge coefficient commonly introduced to account for modeling errors. Comparing to the Kittilsen model, the valve flow coefficient is proportional to the area-function multiplied by the discharge coefficient

$$C_v(u) \propto C_D A_{dc}(u). \tag{2.29}$$

Similar to the Kittilsen model, the gas density upstream the choke is given by the gas law in (2.26) and the mixture density with the homogeneous mixture density equation in (2.24). The gas density downstream the choke is given by a polytropic gas expansion

$$\rho_{\text{gas},dc} = \rho_{\text{gas},wh} p_r^{\frac{1}{\kappa}}. \tag{2.30}$$

Again it is assumed that no other phases than oil, gas, and water exist in the fluid such that the mass fractions sum to one, as in (2.25).

The model differentiates between critical and sub-critical flow using

$$p_r = \begin{cases} \frac{p_{dc}}{p_{wh}} & \frac{p_{dc}}{p_{wh}} \geq p_{\text{critical}} \\ p_{r,c} & otherwise, \end{cases} \tag{2.31}$$

where $p_{r,c} \approx 0.6$ (Jansen, 2015). As for the Kittilsen model, the mass flow rate can be converted to the volumetric flow rate with (2.27).

2.4 Production dataset from Edvard Grieg

In the work behind this thesis, real, historical production data has been utilized for development and analysis of the VFMs. In most of the work, data from the asset Edvard Grieg, located in the North Sea at the Norwegian Continental Shelf (Lundin Energy Norway, 2020), is utilized. The field is an undersaturated oil field without a gas cap, and started its production in late 2015.

The dataset contains measurements from 10 oil production wells at the asset, spanning more than five years of production, see Figure 2.14. The wells have been anonymized and are referred to as W1-W10. These have a variable number of observations. The production from the wells mostly are mostly oil, with little water and some gas (Lundin Energy Norway, 2020). In the historical data, the wellsdo not have gas lift injection.

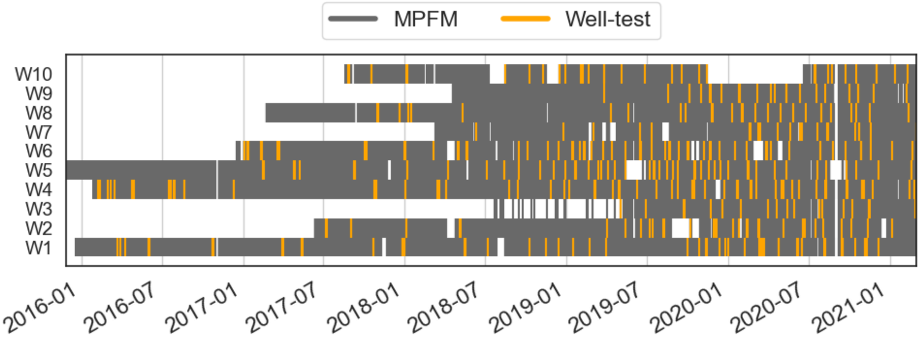


Figure 2.14: Visualization of the observations for the 10 wells at Edvard Grieg against time. Some wells have older historical observations than others. Both multiphase flow meter and well-test measurements are available.

Measurements of the flow rate through the production system is available from both a MPFM located upstream the choke valve, and well-tests conducted with a test separator. The input measurements are pressures and temperatures upstream and downstream the choke valve and the choke valve opening as visualized in Figure 2.10. Further, the mass fractions of the phasic flows are calculated using the available MPFM measurements. Ideally, these should be determined by another model, for instance, a wellbore model as the one in Kittilsen et al. (2014).

Before the dataset is utilized in model development, several preprocessing steps are applied. First, the dataset is compressed using the technology described in Grimstad, Gunnerud, et al. (2016) to obtain steady-state operating points suitable for steady-state modeling. Thereafter, the data is passed through a set of filters that remove faulty measurements. Examples are negative pressures, choke openings, or flow rates.

3 | Contributions

The contributions of this thesis, in the form of six papers, have investigated the top-level research objective presented in Section 1.1. The objective has a wide scope with many opportunities, and the research leading up to this thesis has not covered the whole scope. Mostly, the research has focused on developing steady-state models on the left side of the gray-scale illustrated in Figure 2.2, with a few exceptions. The data-driven model type utilized in the hybrid models has been the neural network. The chosen component for the VFM has been the production choke as illustrated in Figure 2.10. The remaining production system has not been modeled. First-order gradient-based optimization using the MAP objective function has been the typical method to train the VFMs. In five of six papers real and historical production data from the petroleum asset Edvard Grieg (Lundin Energy Norway, 2020) was utilized in the development and investigation of the VFM models.

The papers are listed below and the original papers can be found in Section 5.

- Paper **A** “Developing a Hybrid Data-Driven, Mechanistic Virtual Flow Meter - a Case Study”
- Paper **B** “Identifiability and physical interpretability of hybrid, gray-box models - a case study”
- Paper **C** “Bayesian Neural Networks for Virtual Flow Metering: An Empirical Study”
- Paper **D** “On gray-box modeling for virtual flow metering”
- Paper **E** “When is gray-box modeling advantageous for virtual flow metering?”
- Paper **F** “Passive learning to address nonstationarity in virtual flow metering applications”

The main contributions of the papers are listed in Section 1.2, and repeated here with reference to the relevant papers. An elaboration of the contributions is given in Section 3.1 (not ordered by the date of publication).

- In Paper **A**, a framework for simultaneous estimation of all parameters in hybrid models was proposed.
- In total, in Papers **A**, **B**, and **D**, six different hybrid VFM model variants were developed using historical production data from the asset Edvard Grieg. The hybrid model performances were compared with the performances of mechanistic and data-driven models.
- In **B**, **D**, and **E**, several model properties such as explainability, scientific consistency, and flexibility were investigated, where the hybrid models were compared to mechanistic and data-driven models.
- In Paper **C**, a method to include epistemic and aleatoric uncertainty in VFM modeling was proposed.
- In Paper **F**, a method to address the inherent nonstationarity of the process and sustain the long-term prediction performance of VFMs was proposed.

Author statement

In five of six of the papers, the candidate is the first author and have contributed to the main idea of the paper and development of method, has been responsible for the review of relevant literature, the major part of the coding and analysis of the results, and has been responsible for the write-up of the papers. In Paper **C**, the candidate is the second author and contributed to a major part of the write-up of the paper, and visualization and analysis of the result from the simulation study.

3.1 Elaboration on contributions

Paper A

In Section 2.2, it is described that the estimation problem for hybrid models needs to handle two types of parameters, originating from the mechanistic and the data-driven component, respectively. These parameters have a different interpretation and varying uncertainty on their prior values. Paper **A** suggested a solution for this problem. A practical and convenient framework for simultaneous estimation of all parameters using historical input-output data was developed. With the framework, there was no need to iteratively train the mechanistic and data-driven model components individually. To accomplish simultaneous estimation of the parameters, the Python package PyTorch (Paszke et al., 2019) was utilized to build and train the models. PyTorch utilizes automatic differentiation, which handles the calculation of gradients required in the learning algorithms. Furthermore, the MAP estimation problem in (2.13) was modified to handle different types of parameters. An advantage with the developed framework is that models on the complete gray-scale in Figure 2.2, from purely mechanistic to purely data-driven models can be developed and trained with the same learning algorithm. The framework was tested by developing the first hybrid VFM of its kind on real production data from the asset Edvard Grieg. The VFM was based on the mechanistic Kittilsen choke model introduced in Section 2.3.5 but represented the C_v-curve with a neural network to handle a curve different from the three standard characteristics illustrated in Figure 2.13.

Paper B

Moving a model on the gray-scale in Figure 2.2, from mechanistic towards data-driven, it generally becomes more flexible. With higher flexibility, the model can adapt to unknown physical phenomena not included in the mechanistic model due to simplifications, and thus, reduce process-model mismatch. On the other hand, the model also generally becomes non-identifiable given the available data. Non-identifiability means that it does not exist a unique solution to the parameter estimation problem. Consequently, the model can lose physical interpretability, which is another word for scientific consistency. Paper **B** described mathematically and demonstrated with an industrial scenario that MAP estimation as in Section 2.2 with sufficient regularization of the parameters, can be used to retain the scientific consistency of non-identifiable models. Furthermore, even for identifiable models, the true parameter values are challenging to obtain in practice due to both deterministic and stochastic uncertainties. Therefore, Paper **B** argued that due to increased flexibility and the ability to retain scientific consistency with sufficient parameter regularization, a hybrid VFM can be a better solution than mechanistic VFMs in industrial scenarios as a higher prediction accuracy is often achieved.

Paper D

In Paper **D**, a more thorough investigation into hybrid models on the left side of the gray-scale in Figure 2.2 was conducted. Five hybrid VFMs were compared to a mechanistic and a data-driven VFM. The mechanistic VFM was the Sachdeva choke model from Section 2.3.5 and the data-driven model a neural network. The hybrid models in Paper **D** were developed by examining and targeting the introduced assumptions and simplifications of the Sachdeva model. The seven VFM model variants were trained on production data from 10 petroleum wells on the asset Edvard Grieg (Lundin Energy Norway, 2020). A larger investigation into scientific consistency was conducted for all models. The results in Paper **D** were inconclusive concerning the suitability of different hybrid VFMs. They showed that balancing the task of learning from physics and learning from data is nontrivial and that the performance of the hybrid models was highly dependent on the available data, both the quality and the nonstationarity of the data. Nevertheless, the results indicated that hybrid VFMs can improve the performance over mechanistic VFMs and have a higher scientific consistency than data-driven VFMs.

Paper E

In theory, see Section 2.1, hybrid models should exploit the advantages of both the mechanistic and data-driven modeling domain. Yet, previous research in Paper **A**, **B**, and **D** showed that this is a nontrivial task, and the superiority of hybrid models over mechanistic and data-driven models has not been demonstrated. Therefore, Paper **E** examined four scenarios in which hybrid models were expected to excel: 1) under large process-model mismatch between a mechanistic model and the process, 2) with little available process data, 3) with an increasing noise level in the data, and 4) in nonstationary process conditions. The study was conducted with synthetic data to have complete control of the process and conditions. The simulator used to generate the data was an advanced version of the Sachdeva model in Section 2.3.5, which included slip in the equations. The results clearly showed that hybrid VFMs are advantageous in the two first scenarios and indicated a potential advantage in the fourth scenario.

Paper C

In Section 2.3.3, epistemic and aleatoric uncertainty are drawn out as two challenges of modeling VFMs in industrial scenarios. In Paper **C**, an approach to include both types of uncertainties in the modeling was described. The modeling of uncertainty can increase the robustness of the models and thereby promote trust in the predictions. The model type considered was a probabilistic model called a Bayesian neural network (BNN). The model was trained using variational inference (VI). The result was a model where the uncertainty bands on the predictions were obtained. Further, Paper **C** demonstrated that the approach obtained more robust models by modeling VFMs on data from 60 petroleum wells. On the other hand, the approach was also more computationally heavy than MAP estimation and was therefore not pursued further in the work leading to this thesis. Nevertheless, training hybrid models with VI is an interesting approach for future research.

Paper F

In Section 2.3.3, another potential challenge of VFM modeling is described: the inherent nonstationarity of the underlying process. Results from Paper **D** showed that the performance of VFMs tended to decrease with time, causing a poor long-term prediction performance. Therefore, Paper **F** suggested using passive learning algorithms as a method to sustain the long-term prediction accuracy. Passive learning algorithms assumes that the underlying process is continuously changing, advocating for a periodic updating of the model in time. Two passive learning algorithms were investigated: periodic batch learning and online learning. In the two algorithms, the model update frequency is different. The methods were used to train seven different VFM model types on 10 petroleum wells at Edvard Grieg. The results showed that frequent model updating was essential to obtain high long-term accuracy. Compared to previous work, the prediction accuracy obtained in this work was significantly improved. Furthermore, the results showed that if process measurements were intermittent and infrequent, hybrid modeling together with frequent updating achieved the best sustained long-term performance.

4 | Discussion

In this chapter, the contributions of the work leading up to this thesis are discussed. Firstly, the choices and research directions that have been made during the investigations of hybrid modeling of a VFM are reviewed. Thereafter, a conclusion of whether or not hybrid VFMs are recommended for utilization in the industry is given. Lastly, reflections on the applicability of hybrid models to applications outside virtual flow metering are presented before promising, future research directions within this topic are suggested.

4.1 Comments on the choices and research directions

As mentioned in Chapter 3, the top-level research question has a wide scope, and several choices and research directions were taken in the work behind the contributions of this thesis. To sum up, these are as follows:

- Focusing on developing hybrid models on the left side of the gray-scale in Figure 2.2.
- The neural network is the only data-driven model examined for utilization in the hybrid models.
- The production choke valve has been the chosen VFM component, disregarding the remaining production system.
- Only steady-state models have been considered to model the multiphase flow rate through the production choke.
- First-order gradient-based optimization algorithms have been the main choice of learning algorithms.

A short discussion on the items in the list follows below.

Hybrid models on the left side of the gray-scale

From the perspective of the petroleum engineer, hybrid models with high explainability and high scientific consistency are generally preferred as these properties promote trust in the model. Furthermore, hybrid models on the left side of the gray-scale in Figure 2.2 are easily integrated with existing simulators in the industry. The above arguments have been the main reason for focusing on these model types.

Future research could give more thought about the selection of hybrid model variants using the recommendations from Sansana et al. (2021) as described in Section 2.1.3. The recommendations state that the variant should be selected based on the accuracy of the available mechanistic model. In the work behind this thesis, the utilized mechanistic model of highest accuracy has been the Sachdeva model described in Section 2.3.5. Most likely, there exist mechanistic models of higher accuracy that would increase the overall accuracy of the developed hybrid models. Moreover, recent research on hybrid models on the right side of the gray-scale have shown promising results also for the VFM application (Bikmukhametov and Jäschke, 2020b; Staff et al., 2020) and it would be interesting to deep-dive more into these variants.

The neural network as the data-driven component

The neural network has been the only data-driven model type used in the hybrid models. The neural network was chosen due to its high flexibility to adapt to arbitrarily complex patterns in the data. Furthermore, the neural network was easily integrated with the developed framework for hybrid modeling described in Chapter 3.

In retrospect, examining different data-driven model types should have been considered. As mentioned in Section 2, VFM modeling is typically performed in the small data regime. Furthermore, neural networks generally require large amounts of data in training. Therefore, other data-driven model types could have offered hybrid models with higher accuracy.

On the other hand, the results in Papers **D** and **F** show that hybrid models with neural networks can achieve high accuracy in industrial scenarios with appropriate training algorithms. Furthermore, there are many different data-driven model types to choose from. Hence, to fairly compare different hybrid model structures against each other, the neural network was kept as the only data-driven model type used in the hybrid models.

The production choke valve as the VFM component

Throughout the work with this thesis, only the production choke valve has been considered to model the VFM, while the remaining production system has been neglected. There are several arguments behind this choice. Firstly, due to the inherent complex behavior of multiphase flow, mechanistic modeling of the production system is challenging and time-consuming in general. If the research had examined a larger part of the production system, much time would have passed on first-principle modeling only. For instance, production choke models are typically based on simplified mass and momentum balance equations, disregarding energy balance equations. On the other hand, energy balances can be of more importance for other components in the production system. Therefore, as the top-level research question of this thesis concerns hybrid VFM model types, the overhead of time spent on first-principle modeling was avoided. Secondly, real production data for the choke is typically more available than for other components in the production system. For instance, down-hole sensor measurements tend to be very noisy and biased, and even non-existing for many wells, making wellbore modeling difficult. Naturally, having data to use in modeling is essential for hybrid models. Hence, the production choke became the chosen component.

Nevertheless, when investigating the scientific consistency of the hybrid models in Paper **D**, it was shown that the available data reflected not only the physical behavior of a choke but the complete production system. Hence, the inclusion of a larger part of the production system could be beneficial to the performance of the hybrid models and should be considered in future research.

Steady-state models

All models developed in this work are steady-state models. The main reason is, as discussed in Section 2, because of the inherent complex multiphase flow characteristics, which make it challenging to develop dynamic models suitable for real-time applications due to the high computational burden. Furthermore, for the production choke, steady-state models can be suitable in short periods. Firstly, because the underlying data-generating process is slowly varying, and secondly, due to the short distance between the inlet and outlet of the choke. For other components in the petroleum production system, such as pipelines, dynamics are generally of more importance.

From the results in Paper **F**, it was seen that even though the nonstationarity of the underlying process does influence the performance of the VFM over time, the long-term performance can be sustained by frequent model training. Hence, a steady-state model of the production choke can offer adequate performance and is likely necessary for real-time applications due to the less computational burden of steady-state models contrary to dynamic models. Indeed, several commercially available VFMs are based on steady-state models (Amin, 2015; Bikmukhametov and Jäschke, 2020b).

First-order gradient-based optimization

First-order gradient-based optimization with Adam and the MAP objective function has been the standard approach to train the VFM models. The MAP objective function is used as it was easily modified to handle both data-driven and mechanistic model parameters, as described in Section 3. First-order gradient-based optimization with Adam was chosen due to previous good experiences with this setup on VFM modeling. Moreover, as mentioned in Section 2.2, utilization of second-order gradient-based algorithms requires the computation of the Hessian matrix, which can be challenging for complex models and with much data. The latter is an explanation for the popularity of first-order methods within the data-driven modeling domain. For these reasons and to avoid the overhead of time spent on examining different optimization settings, first-order gradient-based optimization with Adam was kept as the training method. On the other hand, there are several methods where an approximation of the Hessian matrix is applied in the learning algorithm to utilize some form of second-order gradient-based information. Such methods could be interesting to investigate in future research.

4.2 Conclusion

To conclude upon this thesis, the top-level research objective presented in Section 1.1 should be brought back into focus. The contributions of this thesis show that the hybrid modeling approach is suitable for virtual flow metering applications. They can provide a higher prediction accuracy than mechanistic models under process-model mismatch, and generally offer a larger scientific consistency than data-driven models. Furthermore, the hybrid model types investigated in this thesis are easily integrated with existing solutions for VFM in the industry today. However, the task of balancing learning from physics and learning from data is non-trivial. Even though hybrid modeling should exploit the advantages of both the mechanistic and data-driven modeling domain, it can easily adapt to the disadvantages of both domains. Hence, one should inquire whether the added complexity of handling two model types is worthwhile the possible increase in performance.

On the other hand, the latest work with hybrid VFM shows that utilizing both physics and data, together with frequent model updating, is essential to sustain the prediction accuracy over time. In particular, in industrial scenarios where the amount of available data for model development is small. Therefore, the contributions in this thesis support the recommendation of hybrid models for VFM in the industry. It is believed that further research into hybrid modeling will return lucrative results concerning the accuracy while retaining the scientific consistency and explainability of the VFM models.

4.3 Reflection

In this thesis, virtual flow metering has been the application for testing hybrid modeling. However, from the contributions in this thesis, it is believed that hybrid modeling could offer an increased performance of process models in general. In particular, in scenarios where parts of the process' physical behavior are unknown or challenging to model with first-principles, there is little available data, or both. Likely, the best type of hybrid model would differ for the different scenarios.

In literature, there are already several successful applications of hybrid modeling to other fields of research, for instance, within chemistry, climate systems, and biological sciences (Willard et al., 2020). Nevertheless, many industrial applications still use mechanistic modeling as their preferred strategy for process modeling. Therefore, a question remains on what it will take for industries to adopt hybrid modeling as their preferred strategy. Hopefully, the contributions of this thesis can spark enthusiasm and promote trust in hybrid modeling and, in time, aid engineers in exploring hybrid modeling for process systems.

4.4 Future work

The contributions of this thesis have not explored the whole domain of hybrid solutions to VFM and there are several promising research directions. First of all, it would be interesting to investigate the right side of the gray-scale in Figure 2.2 in more depth, for example, with science-guided neural networks. For petroleum assets with much available data, it is believed that such approaches would be beneficial to the VFM performance. Furthermore, a solution for including uncertainty into hybrid models, as was done for data-driven models in Paper **C**, should be pursued. Offering the uncertainty on the predictions and not just point estimates can yield more robust models and promote trust. Moreover, with a probabilistic approach for hybrid modeling, passive learning methods as investigated in Paper **F** could be developed further.

Another interesting direction is to develop hybrid models for utilization in control and optimization strategies. For instance, hybrid VFMs could be used to predict the flow rates over time, which thereafter could be used as input to reservoir optimization. Hopefully, high accuracy VFM models could aid in more accurate reservoir optimization enabling extraction of as much petroleum as possible from existing reservoirs. Another option is to develop hybrid models to use directly in control and optimization applications. For this, models must be developed for which control variables can be inferred from desired flow rates.

Nevertheless, the most promising direction for hybrid VFMs is believed to be the inclusion of data-driven models that enable learning across wells, for instance, multi-task learning models. Such methods would significantly increase the available data that a model can learn from while still exploiting the known physics of the process.

5 | Publications

This chapter contains the original publications that are the result of the work on this thesis. The papers are reprints reformatted to fit the thesis. They are ordered chronologically by the date of publication.

Publications

A Developing a Hybrid Data-Driven, Mechanistic Virtual Flow Meter - a Case Study

Postprint of Hotvedt, Grimstad, and Imsland (2020)

M. Hotvedt, B. Grimstad, and L. Imsland (2020). “Developing a Hybrid Data-Driven, Mechanistic Virtual Flow Meter - a Case Study”. In: *IFAC-PapersOnLine* 53 (2), pp. 11692–11697

Developing a Hybrid Data-Driven, Mechanistic Virtual Flow Meter - a Case Study

M. Hotvedt[1], B. Grimstad[2], and L. Imsland[3]

[1,3]Engineering Cybernetics Department, NTNU, Trondheim, Norway
[2]Solution Seeker

Abstract: Virtual flow meters, mathematical models predicting production flow rates in petroleum assets, are useful aids in production monitoring and optimization. Mechanistic models based on first-principles are most common, however, data-driven models exploiting patterns in measurements are gaining popularity. This research investigates a hybrid modeling approach, utilizing techniques from both the aforementioned areas of expertise, to model a well production choke. The choke is represented with a simplified set of first-principle equations and a neural network to estimate the valve flow coefficient. Historical production data from the petroleum platform *Edvard Grieg* is used for model validation. Additionally, a mechanistic and a data-driven model are constructed for comparison of performance. A practical framework for development of models with varying degree of hybridity and stochastic optimization of its parameters is established. Results of the hybrid model performance are promising albeit with considerable room for improvements.

Keywords: hybrid modeling, virtual flow metering, petroleum production systems

1 Introduction

For a petroleum asset to succeed economically, the operators have to make crucial decisions regarding optimization of the asset. Knowledge regarding the multiphase flow rates in the asset is therefore of high importance. The flow rates may be obtained with deduction well testing, test separators and multiphase flow meters (MPFM), however, these methods are costly and MPFMs call for well intervention upon failure (Marshall and Thomas, 2015). An alternative is virtual flow meters (VFM) that take advantage of measurements to describe the input-output relationship of a system with a mathematical model (Toskey, 2012).

There are several types of VFM models. Dependent on the amount of available process data and prior knowledge of the system, the types may be placed on a scale ranging from mechanistic models (M-models) derived from first-principles, to data-driven models (DD-models), which are generic mathematical models fitted to input-output data (Stosch et al., 2013), see Figure 1. Often, the two extremes are called white-box and black-box models, with reference to the extent of prior knowledge about the system, for example physical interpretation of parameters and relationship between variables. The models in between are hybrid models (H-models) or gray-box models, which utilize modeling techniques from both fields and have a mixture of physical and non-physical parameters.

In this research, an H-model of a well production choke is developed using historical production data from the petroleum platform *Edvard Grieg* (Lundin Energy Norway, 2020). In addition, an M-model and a DD-model are developed for comparison of performance. A practical framework facilitating development of models with varying degree of hybridity and stochastic optimization of model parameters is constructed and conveniently enables future research into the field of hybrid modeling. Background into VFM modeling and the contributions of this research is given in Section 2, the three model types of the production choke is presented in Section 3, the practical framework is outlined in Section 4, the Edvard Grieg case study is presented in Section 5, before simulation results and a conclusion is given in Sections 6 and 7.

2 Background

2.1 Virtual flow meter modeling approaches

The most common way to model VFM in today's oil and gas industry are with M-models, where some well known commercial VFM are Olga, K-Spice and FlowManager (Bikmukhametov and Jäschke, 2020). A great advantage with M-models is their way of representing prior knowledge through the use of first-principles, which leads to interpretable parameters and usually good extrapolation abilities. However, in order for M-models to be computationally feasible, model simplifications are usually a necessity and plant-model mismatch is unavoidable (Solle et al., 2016). Additionally, in complex processes, unknown physical relations are ofttimes present and difficult to capture. VFM with DD-models have shown promising performance suitable for real-time monitoring, without the need of prior knowledge about the system (AL-Qutami et al., 2018). Further, unknown phenomena may be captured if reflected in the process measurements. However, DD-models are data hungry (Figure 1), they struggle with extrapolation in unseen operational settings, parameters generally lack physical interpretation and incorporating process constraints may be challenging, although existent dependent on the DD-method (Pitarch, Sala, and Prada, 2019). Several industrial and academic M- and DD-models are reported in (Mokhtari and Wlatrich, 2016; Balaji et al., 2018; AL-Qutami et al., 2018; Bikmukhametov and Jäschke, 2020) and references therein.

An in-between solution designed to utilize the best of both worlds are H-models. First, notice that the expression "H-models" is widely used in literature for other concepts than combinations of M- and DD-models. Further, one should differ between a hybrid

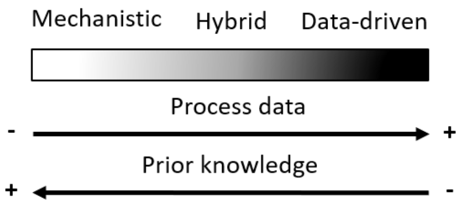


Figure 1: Range of VFM models from mechanistic to data-driven, white-box to black-box.

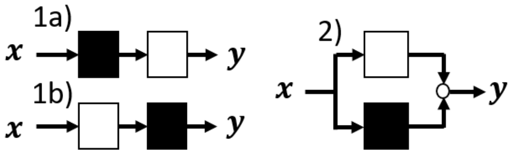


Figure 2: Illustration of hybrid model variants. Serial H-models (type 1) and parallel H-models (type 2).

model development procedure and a hybrid model in application. To clarify, most M-models use real data for parameter estimation. Thus, these models are hybrid in their development procedure, however, after development, parameters are fixed, and the model in application is an M-model. Likewise, a DD-model trained on generated data from an M-model would be hybrid in development, although not in application. Therefore, in this article, we define an H-model as follows:

Definition 2.1: Hybrid model

A hybrid model combines equations from first-principles with generic mathematical structures, both in model development and application.

Following the definition, an H-model is fundamentally categorized in two ways, serial or parallel, see Figure 2. Examples of serial models are online (that is, at each new state) parameter estimation with a DD-model (1a), a DD-model to capture unknown physical phenomena or modeling errors (1b) and physical equations utilized to construct specialized features as input to the DD-model, called feature engineering (1b). A parallel H-model (type 2) would be achieved if a composition of M- and DD-submodels are connected or used in an ensemble model. Naturally, combinations of the two fundamental ways will also be an H-model. Expectantly, compared to an M-model, the H-model should have an increased ability to capture unknown phenomena, yet have better interpretability than a DD-model through the inclusion of prior knowledge and physical parameters. Generally, the DD-part in the H-model will be smaller (in terms of number of parameters) than in a DD-model and should thus require fewer data samples to obtain a satisfactory approximation of the process (Psichogios and Ungar, 1992), see Figure 1.

2.2 Hybrid models in literature

Some of the earliest reported H-models are within the field of chemistry (Psichogios and Ungar, 1992; Kramer, Thomsom, and Bhagat, 1992). However, H-models for VFM are rare although some examples exist in literature. For instance, Xu et al. (2011) used feature engineering in a neural network for wet gas metering. Although feature engineering has shown to boost DD-models, choosing appropriate features is challenging (Sutton and Barto, 2018). Al-Rawahi et al. (2012) estimated the mixture density of multiphase flow using a neural network. However, the neural network required the underlying primary measurements from a MPFM, which may not be as readily available as other measurements. Additionally, MPFM are known to require frequent calibration and may yield high measurement error in-between calibrations (Falcone et al., 2013). Although not a VFM, Baraldi et al. (2014) used an ensemble H-model to detect degradation of production choke valves.

2.3 Contributions

The contributions of this research are two-fold:

- A practical and convenient framework to facilitate development of models with varying degree of hybridity and stochastic optimization of the model parameters.
- A hybrid VFM model for production chokes, developed and validated utilizing real historical production data with readily available measurements such as pressures, temperatures and choke openings.

It must be specified that the main ambition of this research has been to establish a convenient framework for development and utilization of hybrid models. In addition, this research attempts to highlight that H-models may offer advantages over M- and DD-models. Therefore, only one type H-model (type 1a) with parameter estimation using a neural network in an existing M-model has been developed. However, a notable feature with the framework is that, regardless of the hybrid model structure, the model may be trained requiring only measurements of the *output* variable.

3 Choke models

A production choke may be illustrated as in Figure 3, where the volumetric oil flow rate, Q_o, will be estimated using nearby measurements; pressures (p), temperatures

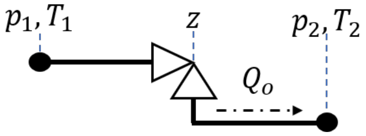


Figure 3: Illustration of the well production choke

(T) and choke opening (z). Three model types have been developed, M-, H- and DD-model. In short notation, these are represented with $\hat{\boldsymbol{y}}_\xi = f_\xi(\boldsymbol{x}_\xi; \boldsymbol{\theta}_\xi)$ where $\xi \in \{m, h, dd\}$, $\hat{\boldsymbol{y}}_\xi = \boldsymbol{Q}_o^e$ is the estimated oil flow rate, f_ξ are the set of model equations, $\boldsymbol{x}_\xi$ are the input measurements and $\boldsymbol{\theta}_\xi$ are learnable model parameters. In the model development, also called training procedure, an optimization algorithm finds the $\boldsymbol{\theta}_\xi$ that minimizes the deviation between estimated ($\boldsymbol{Q}_o^e$) and existing measurements ($\boldsymbol{Q}_o^m$) of the volumetric oil flow rate, see Section 4. The following Sections (3.1, 3.2, 3.3) briefly explain the three model types and Table 1 gives an overview of inputs and parameters to the three models and highlights the difference between the M- and H-model, in this case the form of the C_v-curve.

3.1 Mechanistic model

The M-model is from (Kittilsen, Fjalestad, and Aasheim, 2014), chosen for its simplicity, and described with the equations (1)-(7). More widespread choke models exist (e.g. Sachdeva, Hydro, Al-Safran, see (Haug, 2012) and references therein) and should be experimented with in future work.

$$Q_o = \frac{w_o \dot{m}}{\rho_{o,ST}} \qquad \left[\frac{Sm^3}{h}\right] \tag{1}$$

$$\dot{m} = N C_v(z) \sqrt{Y^2 \rho_m (p_1 - p_2)} \qquad \left[\frac{kg}{h}\right] \tag{2}$$

$$Y^2 = (1 - \frac{1}{3}\frac{x_{lim}}{x_{TP}})^2 \frac{x_{lim} p_1}{p_1 - p_2} \qquad [-] \tag{3}$$

$$\frac{1}{\rho_m} = \frac{w_g}{\rho_g} + \frac{w_o}{\rho_o} + \frac{1 - w_g - w_o}{\rho_w} \qquad \left[\frac{m^3}{kg}\right] \tag{4}$$

$$\rho_g = \frac{M_w p_1}{z_g R T_1} \qquad \left[\frac{kg}{m^3}\right] \tag{5}$$

$$x_{lim} = \min(x_P, x_{TP}) \qquad [-] \tag{6}$$

$$x_P = \frac{p_1 - p_2}{p_1}, x_{TP} = \frac{p_1 - p_2}{p_1}|_c \tag{7}$$

Two important assumptions are those of frozen flow and incompressible liquid; the mass phase fractions, $\boldsymbol{w} = [w_g, w_o, w_w]$ and liquid densities ρ_o and ρ_w are constant in a given operating point (ST for standard conditions). For this model, the valve flow coefficient; $C_v(z)$, is determined with linear interpolation between a given set of test points, which are usually from lab-experiments with water, yet calibrated to the multiphase flow once in place. Further nomenclature may be found in (Kittilsen, Fjalestad, and Aasheim, 2014). The learnable model parameters are chosen to be $\boldsymbol{\theta}_m = [\rho_o, \rho_w, a]$, where a allows the $C_v(z)$ to be shifted; $C_{v,new}(z) = aC_{v,old}(z)$.

3.2 Data-driven model

The DD-model is a fully-connected, feed forward neural network (NN) with the Rectified Linear Unit (ReLU) as activation function on each layer. See e.g. (Balaji et al., 2018) for description of neural networks. The learnable parameters are the weights and biases on each layer, $\boldsymbol{\theta}_{dd} = [\boldsymbol{W}_{dd}, \boldsymbol{b}_{dd}]$.

3.3 Hybrid model

The H-model (type 1a, Figure 2) is represented with the same equations as for the M-model (1)-(7), but with the C_v obtained from a fully-connected, feed forward, NN with ReLU as activation function on each layer. The mass fractions were included as inputs to the NN in an attempt to have the C_v-curve reflect well-specific properties. Thus, the learnable parameters are $\boldsymbol{\theta}_h = [\rho_o, \rho_w, \boldsymbol{W}_h, \boldsymbol{b}_h]$.

Table 1: Overview of parameters and inputs and overview of the C_v-curve form

	M-model	H-model	DD-model
$\boldsymbol{\theta}$	ρ_o, ρ_w, a	ρ_o, ρ_w, $\boldsymbol{W}_h$, $\boldsymbol{b}_h$	$\boldsymbol{W}_{dd}$, $\boldsymbol{b}_{dd}$
$\boldsymbol{x}$	$[p_1, p_2, T_1, z, w_g, w_o]$	$[p_1, p_2, T_1, z, w_g, w_o]$	$[p_1, p_2, T_1, T_2, z, w_g, w_o]$
$C_v(\boldsymbol{x'})$	Linear int. $\boldsymbol{x}'_m = [z]$	NN $\boldsymbol{x}'_h = [z, w_g, w_o]$	n.a

4 Modeling framework

To easily investigate different model types, a practical framework utilizing machine learning techniques is constructed[1]. The framework enables a smooth transition between training a fully M-model to a fully DD-model. It consist of several parts and will be defined in the following.

[1] We have utilized PyTorch, but other possibilities exist such as TensorFlow.

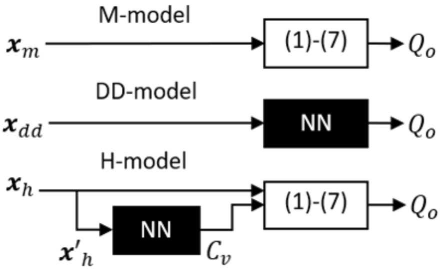


Figure 4: The forward pass illustrated for the three model types

4.1 Defining the model

This part enables a convenient way to implement models with varying degree of hybridity. Firstly, the model parameters must be defined, either as single, learnable parameters, as for the physical parameters, or as NN's with weights and biases. Thus, a model will effortlessly move on the gray-scale (Figure 1) dependent on the defined parameters. Thereafter, the forward pass, the propagation of input data through the model, will be defined as a computational graph, enabling access to the model derivatives through automatic differentiation. The forward pass for the different models is illustrated in Figure 4. A particularly appealing property with this framework is that measurements of the C_v are not required as the model is trained on the output, Q_o.

4.2 Defining the optimization problem

Once the model is defined, a general optimization problem to find the $\boldsymbol{\theta}_\xi$ that minimizes deviation between the model estimates $\hat{\boldsymbol{y}}_\xi = \boldsymbol{Q}^e_{o,\xi} = f_\xi(\boldsymbol{x}_\xi; \boldsymbol{\theta}_\xi)$ where $\xi \in \{m, h, dd\}$ and the measurements $\boldsymbol{y} = \boldsymbol{Q}^m_o$ may be set up as

$$\begin{aligned} \hat{\boldsymbol{\theta}}_\xi &= \arg\min_{\boldsymbol{\theta}_\xi} J(\boldsymbol{\theta}_\xi, \boldsymbol{\lambda}_\xi) \\ &= \arg\min_{\boldsymbol{\theta}_\xi} \Bigg(\tfrac{1}{n} \textstyle\sum_{i=1}^{n} \Big(y^{(i)} - f_\xi(\boldsymbol{x}_\xi^{(i)}; \boldsymbol{\theta}_\xi) \Big)^2 \\ &\qquad + \tfrac{1}{n} \textstyle\sum_{j=1}^{p} \lambda_{j,\xi} (\theta_{j,\xi} - \mu_{j,\xi})^2 \Bigg) \end{aligned} \tag{8}$$

The first term in eq. (8) is the mean square error (MSE) and the second is an ℓ_2-regularization term with regularization factors λ_i. For the physical parameters, the goal of regularization is to penalize deviation of the parameters from a prior (expected) value, μ_i, and maximum a posteriori (MAP) estimation has been set up to automatically calculate the λ_i factors, see Section 4.3. For the NN parameters, common practice is followed and μ_i is set to zero. If $\boldsymbol{Q}^m_o$ are available from different measurement sources, additional MSE terms may be added and weighted according to the uncertainty in the measurement source. In this research, only measurements from a MPFM has been utilized.

The framework solves the optimization problem in eq. (8) using iterative gradient-based optimization. The update formula may be stated as follows

$$\boldsymbol{\theta}_\xi^{k+1} = \boldsymbol{\theta}_\xi^k - \alpha^k \mathcal{M}(\boldsymbol{x}_\xi^{k'}; \boldsymbol{\theta}_\xi^k, \boldsymbol{\lambda}_\xi) \tag{9}$$

where α^k is the learning rate (or step-size), $\boldsymbol{x}_\xi^{k'}$ is a subset of the data samples and $\mathcal{M}$ is the set of equations calculating the step direction. Different algorithms may be selected, such as stochastic gradient descent (SGD), Adam among others (Bottou, Curtis, and Nocedal, 2018). Stochastic gradient-based optimization algorithms has the advantage of being well suited for large scale models, either in terms of large datasets or many parameters, where other optimization algorithms utilizing linesearch may be to computationally expensive (Bengio, 2012). In SGD, $\mathcal{M} = \nabla_{\boldsymbol{\theta}} \tilde{J}(\boldsymbol{x}; \boldsymbol{\theta}^k, \boldsymbol{\lambda})$, where $\nabla \tilde{J}$ may be calculated with different number of samples (batch size). Knowing which optimization algorithm yields the best result is challenging as it might be problem dependent. Therefore, the framework promote investigation of different optimization algorithms. In this research, Adam is used for all models.

4.3 Calculation of regularization parameters

The λ_i regularization factors for the physical parameters may be automatically calculated through MAP estimation. If one assumes a model of the form

$$y = f(x; \theta) + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2) \tag{10}$$

the MAP estimation may be set up as follows utilizing Bayes' rule, where (X, y) is the collection of data points

$$\hat{\theta}_{MAP} \quad = \arg\max_\theta \left(\log p(y|X, \theta) + \log p(\theta)\right) \tag{11}$$

If one additionally assumes independent Gaussian priors of the parameters $\theta_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, the MAP estimation will result in, after some rearrangements,

$$\begin{aligned} \hat{\theta}_{MAP} \quad &= \arg\min_\theta \Bigg(\textstyle\sum_{i=1}^n \left(y^{(i)} - f(x^{(i)}; \theta)\right)^2 \\ &\quad + \textstyle\sum_{i=1}^p \frac{\sigma_\epsilon^2}{\sigma_i^2} (\theta_i - \mu_i)^2 \Bigg) \end{aligned} \tag{12}$$

Dividing by n and setting $\lambda_i = \sigma_\epsilon^2 / \sigma_i^2$, the MAP estimation will be the same as the estimate in eq. (8). The σ_i may be determined based on physical bounds and if σ_ϵ is known, λ_i is automatically calculated. In practice, σ_ϵ must be tuned, however, the number of coefficients to determine decreases.

5 Case study - Edvard Grieg

Historical production data from *Edvard Grieg* has been utilized in the model development procedure and to analyze performance of the models. In addition to pressures, temperatures, and choke opening (see Figure 3), measurements from a MPFM located upstream the choke restriction was used for training the model, keeping in mind that MPFM measurements may be faulty and require frequent calibration (Falcone et al., 2013). Future work should include well-tests which in general have higher accuracy than MPFM measurements. The production data are from 10 oil wells, yielding a total of 30 models, over a period of 1248 days. Consequently, the assumption of constant physical parameters may be a rough approximation and future work should consider updating the models at certain intervals in time to account for changes in the true process.

The data was preprocessed in two steps before performing modeling. First, the raw production data was processed by Solution Seeker's data squashing technology (Grimstad et al., 2016). The data squashing algorithm partitions the data into intervals of steady-state operation. The data in each interval is then compressed to mean values using statistics suitable for time-series data. The result is a compressed data set of steady-state operating points, suitable for steady-state modeling. In the second preprocessing step, samples considered invalid, such as samples with unrealistically large well head pressures or negative flow rates, were removed and some samples were slightly modified, for instance small negative flow rates, where measurement noise was the likely cause of error. The second step resulted in a variable number of samples per well, in the range 612-2175. Further, the mass fractions were calculated using MPFM flow rates and standard densities. In an industrial setting, the mass fractions are often calculated from sparse well-test samples, thus to mimic this setting, a mass fraction update time of 30 days was employed, using an average of the last 20 samples.

The data set of each well was divided into two, training (75%) and test (25%), where 15% of the latest training data was used as a validation set to decide upon the hyperparameters in the training procedure. An ambition was for the three model types to generalize well across all wells of the asset. Consequently, the same set of hyperparameters was used for a model type, instead of individual tuning of each model type for each well. However, one should expect a lower overall error by individual tuning due to dissimilar well operating conditions and variable sample numbers, and this should be considered in future work. The average root mean square error (RMSE) and average mean absolute error (MAE) of the 10 wells were monitored and the best set of hyperparameters was chosen based on the minimum obtained averages. However, if prominent overfitting occurred in a well for a set of hyperparameters, that is, if the validation error increased when the training error decreased towards the end of training, the next best set of hyperparameters was chosen. Practical recommendations from (Bengio, 2012) was followed in the tuning process.

For all models, the learning rate (α) was thoroughly experimented with as this often is the most important hyperparameter to tune (Bengio, 2012). Further, for the M-model, the physical parameters had to converge within the specified bounds, thus, the number of epochs (E), that is, the number of loops through the training set, and σ_ϵ were tuned thereafter. For the H-model, both physical and NN parameters had to be found. However, E may be high and the NN architecture (width/depth) large without leading to overfitting of the NN as long as regularization of the NN parameters is applied (Bengio, 2012). Hence, E was set sufficiently high and σ_ϵ adjusted for convergence of the physical parameters within bounds, the width/depth was set to 20/2 and combinations of α and the NN regularization factor, $\lambda_{i,nn}$, were tested. For the DD-model, the same recommendations were followed. The E was set high and combinations of α and $\lambda_{i,nn}$ investigated. The width/depth was set to 70/2. Lastly, the batch size (B) is often tuned independently of the other hyperparameters (Bengio, 2012) and was thus tuned last. Even though considerable effort was put into fair tuning of the three models, a Bayesian optimization approach will be investigated in the future to avoid (non-intentional) advantage to either model.

An overview of the final hyperparameters are given in Table 2. Observe that the M-model required a larger σ_ϵ than the H-model for the physical parameters to converge within specified bounds, indicating that the H-model accounts for some of the measurements noise with the DD-part. Further, the best performance for the H-model was obtained with a low batch number, however, only small differences in average error lead to this choice.

Table 2: Overview of the final model hyperparameters

	M-model	H-model	DD-model
E	5000	2000	2000
B	150	32	150
α	0.01	0.01	0.01
σ_ϵ	25	10	-
$\lambda_{i,nn}$	-	0.01	0.001
width/depth	-	20/2	70/2

6 Simulation results

The simulation results are shown in Figure 5, where the RMSE and MAE of the test set for the 10 wells are illustrated, and Figure 6 which is a cumulative deviation plot (CDP) (Corneliussen et al., 2005) indicating the accuracy of the developed VFM models, that is, how many of the test points fall within a certain deviation from the measurement. There are several interesting observations to be made from the results. Firstly, notice the extreme outlier that is present in the DD-model performance in Figure 5. The outlier is caused by one of the wells which had an operational setting very different from the setting in the training set. As mentioned in Section 2, DD-models may struggle with extrapolation in unseen operational settings which may

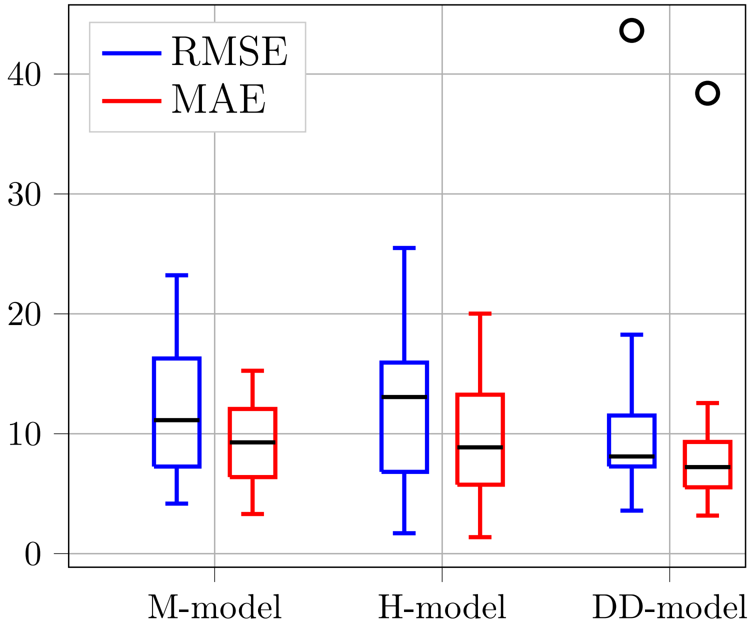


Figure 5: Boxplot overview with median for the three model types across the 10 wells.

explain the outlier. In that case, the results indicate that the H-model has preserved some of the extrapolation power of the M-model which do not have the extreme outlier. However, despite the H-model obtaining the lowest errors for some of the wells, the M-model performs better than the H-model due to less spread between the quartiles. Now, the only difference between the two models is the form of the C_v-curve. This indicates that our prior belief of the C_v-curve in the M-model was good and that no flexibility was added to the H-model by having the C_v as an NN. Naturally, these results are preliminary and further investigations are necessary. In particular, different H-model variants may better leverage the advantages of both M- and DD-models.

Generally, the results show a higher error than expected. Other studies have reported almost 90% performance for 20% deviation in the CDP (e.g. (AL-Qutami et al., 2018)), whereas in this paper about 70% performance for 20% deviation is achieved. There may be several causes for the large error. Firstly, preprocessing of the data could be improved by for instance further outlier removals. In addition, MFPM measurements was used for mass fraction calculation and in training despite a possibility of being faulty in between calibrations. Further, the mass fractions was updated every 30 days to mimic an industrial setting, however, in training, continuous mass fraction updates could be utilized. Hence, future work should include measurement sources with higher accuracy, such as well-tests, and analyze performance with continuous mass fraction updates. Secondly, the model types were generalized across all wells

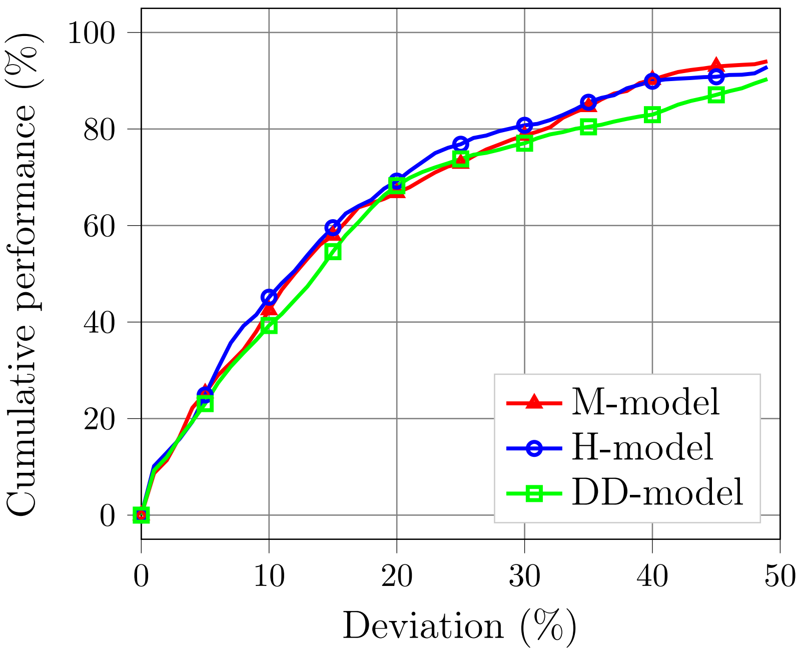


Figure 6: Cumulative deviation plot of all test samples across the 10 wells.

and improved results would most likely be achieved with individual tuning. Further, utilization of a more accurate mechanistic choke model or optimization of additional physical parameters may decrease the error in the M- and H-model. Thirdly, the number of days for which the models are used in prediction should be taken into account. For some of the wells, the test set covered more than 200 days, whereupon the true process could have changed significantly and the models lack validity. Future work should consider online training of the models at regular intervals in time. Nonetheless, the main goal of this research was not to find exceptional models, but to illustrate that an H-model may offer advantages over M- and DD-models and to establish a framework for convenient future research.

7 Conclusion

Results in Section 6 indicate that hybrid modeling is promising and may offer advantages over both mechanistic and data-driven modeling. However, results are preliminary and there is considerable room for improvements. Future work should put more effort into preprocessing of the data set, analysis of the mass fraction calculation influence on performance and inclusion of well tests in training and validation. Further, individual tuning of each well should be investigated and the models should be tested with different data sets, for instance from other petroleum assets. Last but not least, future work should explore different hybrid model variants, for which the presented framework is convenient and highly suitable.

Acknowledgment

This research is a part of BRU21 - NTNU Research and Innovation Program on Digital and Automation Solutions for the Oil and Gas Industry (www.ntnu.edu/bru21) and supported by Lundin Norway AS.

References

Balaji, K. et al. (2018). “Status of data-driven methods and their application in oil and gas industry”. In: *SPE Europec featured at 80th EAGE Conference and Exhibition.*

Baraldi, P. et al. (2014). “A hybrid ensemble-based approach for process parameter estimation and degradation assesment in offshore oil platforms”. In: *International Journal of Performability Engineering* 10 (5), pp. 497–509.

Bengio, Y. (2012). “Practical Recommendations for Gradient-Based Training of Deep Architectures.” In: *arXiv:1206.5533v2*, pp. 1–33.

Bikmukhametov, Timur and Johannes Jäschke (2020). “First Principles and Machine Learning Virtual Flow Metering: A Literature Review”. In: *Journal of Petroleum Science and Engineering* 184.

Bottou, L., F.E. Curtis, and J. Nocedal (2018). “Optimization Methods for Large-Scale Machine Learning”. In: *Society for Industrial and Applied Mathematics* 60(2), pp. 223–311.

Corneliussen, Sidsel et al. (2005). *Handbook of multiphase flow metering.* The Norwegian Society for Oil and Gas Measurements.

Falcone, G. et al. (2013). “Multiphase flow metering: current trends and future developments”. In: *SPE Annual Technical Conference and Exhibition.*

Grimstad, B. et al. (Sept. 2016). “A Simple Data-Driven Approach to Production Estimation and Optimization”. In: *In: SPE Intelligent Energy International Conference and Exhibition.*

Haug, R. (2012). “Multiphase Flow Through Chokes”. In: *NTNU open.*

Kittilsen, P., K. Fjalestad, and R. Aasheim (2014). “Stabilized and Increased Well Production Using Automatic Choke Control”. In: *Society of Petroleum Engineers.*

Kramer, M. A., M. L. Thomsom, and P. M. Bhagat (1992). “Embedding Theoretical models in neural networks”. In: *Proceedings of American Control Conference* 1 (1), pp. 475–479.

Lundin Energy Norway (2020). *Edvard Grieg.* https://lundin-energy-norway.com/edvard-grieg/. Acccessed: 18.01.2020.

Marshall, C. and A. Thomas (2015). “Maximising Economic Recovery - a review of well test procedures in the North Sea”. In: *Conference: SPE Offshore Europe Conference and Exhibition.*

Mokhtari, K. and P. Wlatrich (2016). “Performance Evaluation Of Multiphase Flow, Models Applied To Virtual Flow Metering”. In: *WIT Transactions on Engineering Sciences.*

Pitarch, J., A. Sala, and C. de Prada (2019). “A systematic grey-box modeling methodology via data reconciliation and SOS constrained regression”. In: *Processes* 7.3, pp. 1–23.

Psichogios, D.C. and L.H. Ungar (1992). “A Hybrid Neural Network-First Principles Approach to Process Modeling”. In: *AIChE Journal* 38(10), pp. 1499–1511.

AL-Qutami, T.A et al. (2018). “Virtual multiphase flow metering using diverse neural network ensemble and adaptive simulated annealing”. In: *Expert Systems With Applications* 93, pp. 72–85.

Al-Rawahi, N. et al. (2012). “A neural network algorithm for density measurement of multiphase flow”. In: *Multiphase Science and Technology* 24 (2), pp. 89–103.

Solle, D. et al. (2016). “Between the Poles of Data-Driven and Mechanistic Modeling for Process Operation”. In: *Chemie Ingenieur Technik.*

Stosch, M.v et al. (2013). “Hybrid semi-parametric modeling in process system engineering: Past, present and future”. In: *Computers and Chemical Engineering.*

Sutton, R.S. and A.G. Barto (2018). *Reinforcement Learning, an introduction.* Cambridge, Massachusetts, London, England: The MIT Press.

Toskey, E.D. (2012). “Improvements to Deepwater Subsea Measurements RPSEA Program: Evaluation of Flow Modeling”. In: *Offshore Technology Conference.*

Xu, L. et al. (2011). “Wet gas metering using a revised venturi meter and soft-computing approximation techniques”. In: *IEE transactions on instrumentation and measurement* 60 (3), pp. 947–956.

B Identifiability and physical interpretability of hybrid, gray-box models - a case study

Postprint of Hotvedt, Grimstad, and Imsland (2021)

M. Hotvedt, B. Grimstad, and L. Imsland (2021). “Identifiability and physical interpretability of hybrid, gray-box models - a case study”. In: *IFAC-PapersOnLine* 54 (3), pp. 389–394

Identifiability and physical interpretability of hybrid, gray-box models - a case study

M. Hotvedt[1], B. Grimstad[2], and L. Imsland[3]

[1,2, 3]Engineering Cybernetics Department, NTNU, Trondheim, Norway
[2]Solution Seeker

Abstract: Model identifiability concerns the uniqueness of uncertain model parameters to be estimated from available process data and is often thought of as a prerequisite for the physical interpretability of a model. Nevertheless, model identifiability may be challenging to obtain in practice due to both stochastic and deterministic uncertainties, e.g. low data variability, noisy measurements, erroneous model structure, and stochasticity and locality of the optimization algorithm. For gray-box, hybrid models, model identifiability is rarely obtainable due to a high number of parameters. We illustrate through an industrial case study – modeling of a production choke valve in a petroleum well – that physical interpretability may be preserved even for non-identifiable models with adequate parameter regularization in the estimation problem. To this end, in a real industrial scenario, it may be beneficial for the model's predictive performance to develop hybrid over mechanistic models, as the model flexibility is higher. Modeling of six petroleum wells on the asset Edvard Grieg using historical production data show a 35% reduction in the median prediction error across the wells comparing a hybrid to a mechanistic model. On the other hand, both the predictive performance and physical interpretability of the developed models are influenced by the available data. The findings encourage research into online learning and other hybrid model variants to improve the results.

Keywords: Gray-box, hybrid model, identifiability, interpretability, choke modeling

1 Introduction

Mathematical modeling of physical processes is an important aspect of many engineering fields and may aid in the analysis and prediction of a process response to changes in state and control variables. Therefore, a model should be a good representation of the underlying process. However, during mathematical modeling, there is often a compromise between complexity and intractable models (Zendehboudi, Rezaei, and Lohi, 2018). Usually, a higher model complexity yields a better representation of the process but is harder to solve. Another trade-off is how much prior knowledge should be incorporated into the model and how much should be learned from process

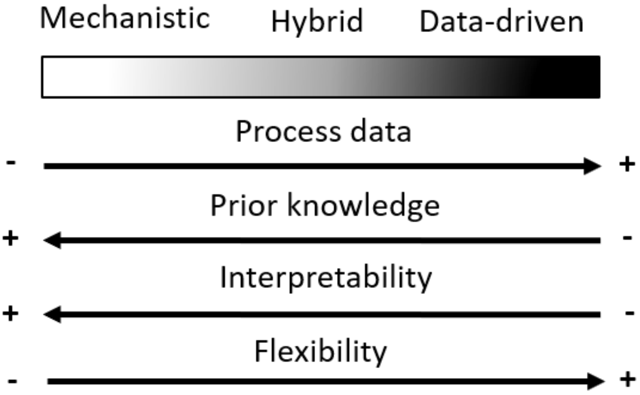


Figure 1: Gray-scale of mathematical models, ranging from mechanistic to data-driven models.

data. Typically, mathematical models may be placed on a gray-scale ranging from mechanistic to data-driven models, or, from white- to black-box models respectively, see Fig. 1.

Mechanistic models are built from first-principle equations, with possible empirical closure relations, and require considerable understanding of the physical behavior of the process. These models are often of high complexity and call for simplifications to be computationally feasible. Data-driven models are constructed from generic mathematical equations fitted to process data and require no prior knowledge about the process. These model types have a high degree of *flexibility*, which is the ability to adapt to arbitrarily complex patterns in data. Therefore, contrary to mechanistic models, data-driven models may capture unmodeled or unknown process behavior as long as these are reflected in the available data. However, data-driven models are data-hungry and require the data to be sufficiently rich to represent the process behavior appropriately. If care is not taken, overfitting is a frequent outcome resulting in poor extrapolation power to future process conditions (Solle et al., 2016). As a result of using first-principle equations, mechanistic models are typically better at extrapolation.

Hybrid modeling combines mechanistic and data-driven modeling techniques and attempts to preserve the advantages while diminishing the disadvantages of the two approaches. There are several ways to construct hybrid models and much literature on the topic (e.g. Psichogios and Ungar (1992), Kramer, Thomsom, and Bhagat (1992), Xu et al. (2011), Zendehboudi, Rezaei, and Lohi (2018), Bikmukhametov and Jäschke (2020), and Hotvedt, Grimstad, and Imsland (2020)). Hybrid modeling may be approached from either side of the gray-scale in Fig. 1. Models on the white side of the gray-scale typically have a higher degree of *interpretability* than models on the black side. In literature, there are numerous definitions of, and several aspects connected to interpretability. According to Roscher et al. (2020) is interpretability the ability to present a model in understandable terms to a human. The focus in this article is on the aspect *physical* interpretability. A model will be physically interpretable if 1) its output behavior is in line with existing physical principles, and 2) any model parameter representing a physical quantity has a physically feasible value. From a process analysis perspective, physically interpretable models are of great value as the models are often better at extrapolation and the analysis of the model response to changes in variables is simplified. This promotes trust in the model. To this end, this article will focus on hybrid models on the white side of the gray-scale in Fig. 1. A mechanistic model is used as a baseline and a data-driven model is inserted to increase model flexibility.

According to Deconinck and Roels (2017), a prerequisite to obtaining physically interpretable models is model *identifiability*. With this property, uncertain model parameters may be *uniquely* determined from the available process data (Goodfellow, Bengio, and Courville, 2016). However, when a model is moved on the gray-scale from mechanistic towards data-driven, the number of parameters increases and the model generally becomes non-identifiable given the available data. This is particularly the case for deep neural networks which are often designed to have a complexity above the interpolation threshold. Furthermore, identifiability is influenced by both stochastic uncertainties, e.g. noisy data and stochasticity of the estimation problem, and deterministic uncertainties, e.g. erroneous model structure and local optimization algorithms, making most models non-identifiable in practice. For these reasons, in the data-driven modeling domain, a common practice is to seek a model that generalizes well and has a high degree of flexibility, rather than seeking identifiable models. On the other hand, data-driven models typically have non-physical parameters. Ensuring unique parameter values are therefore not as important as for hybrid and mechanistic models where there exist physical parameters whose values should be physically consistent to obtain physically interpretable models. Identifiability and interpretability have been topic for many studies, e.g. (Brun, Reichert, and Künsch, 2001; Raue et al., 2009; Deconinck and Roels, 2017; Brastein, Sharma, and Skeie, 2019), but few have studied this for hybrid, gray-box models.

This article will study the identifiability, physical interpretability, and predictive performance of a mechanistic and a hybrid petroleum well production choke valve model. The first two concepts are investigated through a synthetic case study where we have perfect information about the underlying process. Thereafter, the two latter concepts are examined in an industrial scenario where real and historical production data from six petroleum wells on Edvard Grieg (Lundin Energy Norway, 2020) are utilized in the model development.

2 Parameter estimation problem

For all models on the gray-scale, parameter estimation is essential to obtain adequate model predictions of the process. Consider a dataset $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$ with n observations of the process state and control variables $\boldsymbol{x}_i \in \mathbb{R}^d$, also called explanatory variables, and target variable $y_i \in \mathbb{R}$. Assume the process to be described by

$$y_i = f(\boldsymbol{x}_i; \boldsymbol{\phi}) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2), \tag{1}$$

where $\hat{y}_i = f(\boldsymbol{x}_i; \boldsymbol{\phi})$ is a general steady-state model with parameters $\boldsymbol{\phi} \in \mathbb{R}^m$, and ϵ_i is normally distributed measurement noise. In general, the parameter estimation problem may be posed as an optimization problem

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} J(\boldsymbol{y}, \hat{\boldsymbol{y}}), \tag{2}$$

where J is a scalar, non-negative objective function, and $\boldsymbol{y}, \hat{\boldsymbol{y}} \in \mathbb{R}^n$ are the observations of target variable measurements and estimates respectively. A common approach to solving the estimation problem is through least squares estimation, which is equivalent to maximum likelihood estimation (MLE) of the model in (1). Denoting the design matrix by $X \in \mathbb{R}^{n\times d}$ with rows $X_{i,*} = \boldsymbol{x}_i$, the MLE may be written compactly as

$$\boldsymbol{\phi}_{MLE}^* = \arg\min_{\boldsymbol{\phi}} \left(\boldsymbol{y} - f(X; \boldsymbol{\phi})\right)^\top \left(\boldsymbol{y} - f(X; \boldsymbol{\phi})\right). \tag{3}$$

This is a nonlinear, nonconvex optimization problem that generally has several solutions, which all fulfill

$$\nabla_\phi J_{MLE} = -2\nabla_\phi f(X; \boldsymbol{\phi})^\top (\boldsymbol{y} - f(X; \boldsymbol{\phi})) = 0. \tag{4}$$

If the sensitivity matrix

$$\nabla_\phi f(X; \boldsymbol{\phi})|_\phi = \begin{bmatrix} \frac{\partial f}{\partial \phi_1}(X) & \dots & \frac{\partial f}{\partial \phi_m}(X) \end{bmatrix} |_\phi \tag{5}$$

has full column rank for all ϕ, we may say that the problem is identifiable and there exist a unique solution to (3). Nevertheless, determining the rank of (5) for the general nonlinear problem is nontrivial.

The sensitivity matrix is highly influenced by the span, or variability, of X, and uncertainty in the measurements of the explanatory variables. If the model is non-identifiable, a natural approach is to increase the variability of X. However, for petroleum production systems this is not trivial as the production data is influenced by the operational practices of the operator. Further, designed experiments are often unaffordable due to the operation of the asset at a non-optimal operating point. Other approaches to obtain identifiability of non-identifiable over-parameterized models are parameter ranking methods (Chan and Hansen, 1992) and parameter reduction methods (White, 2003). The first method estimates the parameters that are most influential to the model output while fixing the remaining parameters at constant prior values. The latter method removes redundant parameters. Nevertheless, in a real-life setting, the underlying process and parameter values are unknown, and fixing parameters at constant values may lead to bias estimates. For hybrid models, many of the parameters are non-physical. Therefore, finding redundant parameters or good priors to fix the less influential parameters is challenging.

On the other hand, even if the sensitivity matrix has full rank, identifiability may be influenced by other aspects. For instance, for highly nonlinear, non-convex problems, iterative, stochastic, and local optimization algorithms are often required to solve the estimation problem in reasonable time (Bottou, Curtis, and Nocedal, 2018). The stochasticity and locality of these algorithms may prevent model identifiability. For instance, with stochastic gradient descent (SGD) the parameters are updated with

$$\boldsymbol{\phi}^{k+1} = \boldsymbol{\phi}^k - \alpha^k \mathcal{M}^k(\boldsymbol{y}^k, \hat{\boldsymbol{y}}^k), \tag{6}$$

where $\alpha \leq 1$ is the learning rate and $\mathcal{M}$ is a stochastic gradient commonly calculated using a subset or mini-batch of the data samples. When a model increases in complexity, both in terms of model parameters and available process data, batch SGD methods scale better and are computationally feasible (Bottou, Curtis, and Nocedal, 2018). The totality of these issues implies that model identifiability is challenging to obtain in practice for complex models, and the physical interpretability of the model may be easily lost.

An alternative approach to counteract the loss of physical interpretability in non-identifiable models is using parameter regularization in the estimation problem. Regularization of the parameters is achieved by setting up maximum a posterior (MAP) estimation of the parameters, instead of MLE. MAP estimation attempts to find the mode of the posterior probability distribution of the model parameters given the data $\mathcal{D}$

$$\boldsymbol{\phi}^*_{MAP} = \arg\max_{\boldsymbol{\phi}} p(\boldsymbol{\phi}|\mathcal{D}). \tag{7}$$

Through utilization of Bayes' theorem and assuming normally distributed parameter priors $\phi_i \sim \mathcal{N}(\mu_i, \sigma_i^2), i \in \{1, .., m\}$, the optimization problem

$$\begin{aligned} \boldsymbol{\phi}^*_{MAP} = \arg\min{}_{\boldsymbol{\phi}} \Bigg[& \underbrace{\left(\boldsymbol{y} - f(X;\boldsymbol{\phi})\right)^\top \left(\boldsymbol{y} - f(X;\boldsymbol{\phi})\right)}_{\text{MLE}} \\ + & \underbrace{(\boldsymbol{\phi} - \boldsymbol{\mu})^\top \Pi (\boldsymbol{\phi} - \boldsymbol{\mu})}_{\text{parameter regularization}} \Bigg], \quad \Pi = diag(\tfrac{\sigma_\epsilon^2}{\sigma_1^2}, .., \tfrac{\sigma_\epsilon^2}{\sigma_m^2}) \end{aligned} \tag{8}$$

is derived. In short, MAP estimation is a trade-off between minimizing the deviation between model estimates and measurements, and penalization of the deviation of the parameters away from their prior mean value $\boldsymbol{\mu}$. This type of regularization method is called Tikhonov or ℓ_2-regularization and is referred to as a shrinking method (Hastie, Tibshirani, and Friedman, 2009). Dependent on Π, ℓ_2-regularization may make the MAP problem nonsingular with one unique solution when the MLE problem is not.

To see this, consider the linear case where

$$\hat{\boldsymbol{y}} = f(X; \boldsymbol{\phi}) = X\boldsymbol{\phi}. \tag{9}$$

The solutions of MLE in (3), and MAP in (8), for (9) are respectively

$$\begin{aligned} \boldsymbol{\phi}^{\star}_{MLE} &= \left(X^{\top}X\right)^{-1} X^{\top}\boldsymbol{y}, \\ \boldsymbol{\phi}^{\star}_{MAP} &= \left(\Pi + X^{\top}X\right)^{-1} \left(X^{\top}\boldsymbol{y} + \Pi\boldsymbol{\mu}\right). \end{aligned} \tag{10}$$

The MAP solution adds a diagonal, positive definite matrix to $X^{\top}X$ before inversion. Dependent on Π, the MAP problem may be nonsingular even if the $X^{\top}X$ does not have full column rank. If the elements in Π are set sufficiently high, the MAP solution approximate to

$$\boldsymbol{\phi}^{\star}_{MAP} \approx \boldsymbol{\mu}. \tag{11}$$

This indicates the importance of good parameter priors. An appropriate selection of Π will allow parameter deviation away from the prior mean value while keeping the parameters within a feasible range, thereby preserving physical interpretability. The same results may be obtained (locally) for the general nonlinear system through a first-order Taylor approximation of f.

In this study, MAP estimation is used to train the mechanistic and hybrid models. For hybrid models, the parameter regularization term in (8) is divided into two, one each for the physical and non-physical parameters. For the physical parameters, the prior parameter distributions $\{\mu_i, \sigma_i^2\}_{i=1}^{m}$ need to be specified. The variance of the parameters may be determined using physical bounds. Notice, if $\sigma_i^2 \to \infty$ then $\Pi \approx \mathbf{0}$, and the MAP estimation becomes an MLE problem in practice. The same effect may be achieved by setting σ_ϵ^2 small. Ideally, σ_ϵ^2 should be determined prior to training, however, it may be used as a regularization tuning constant. For the non-physical parameters, common practice is to penalize large parameters by setting the same prior $\mu_i = 0$ and $\Pi = \lambda\mathbb{I}$ on all parameters (Goodfellow, Bengio, and Courville, 2016). The optimizer Adam (Kingma and Ba, 2015) is utilized for all models. A hyperparameter search with Bayesian Optimization (Klein et al., 2017) is used to find λ and the learning rate α in (6).

3 Choke models

A petroleum well production choke valve may be illustrated as in Figure 2. Available sensor measurements are typically pressures (p) and temperatures (T), upstream (1) and downstream (2) the choke valve, and choke openings (u). In parameter estimation, measurements of the model output, in this research the oil volumetric flow rate, (Q_O), is also required, for instance from a multiphase flow meter or test separator. The mass fractions are treated as known and as inputs to the model. The mechanistic model used in this study is taken from Sachdeva et al. (1986). The choke model is developed

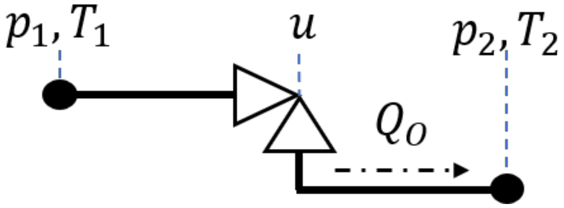


Figure 2: The petroleum well production choke valve with available measurement sensors.

from the combined steady-state mass and momentum balance along a streamline and the equation for the mass flow rate through the choke is given as

$$\dot{m} = C_D A_2(u)\Bigg(2\rho_2^2 p_1 \left(\tfrac{\kappa}{\kappa-1}\eta_G \left(\tfrac{1}{\rho_{G,1}} - \tfrac{p_r}{\rho_{G,2}} \right) + \left(\tfrac{\eta_O}{\rho_O} + \tfrac{\eta_W}{\rho_W} \right)(1 - p_r) \right) \Bigg)^{1/2}, \tag{12}$$

where A is the flow area of the choke, typically a nonlinear function of u, ρ is the mixture density, $\eta_i, \rho_i, i \in \{G, O, W\}$ are the mass fractions and densities of gas, oil, and water, respectively, κ is the adiabatic gas expansion coefficient, p_r is the downstream to upstream pressure ratio, and C_D is a discharge coefficient commonly introduced to account for modeling errors. The oil volumetric flow rate in standard conditions is obtained with

$$Q_O = \frac{\eta_O \dot{m}}{\rho_{O,ST}}. \tag{13}$$

In model development, Sachdeva et al. (1986) assumes 1) the upstream gas density may be described with the real gas law

$$\rho_{G1} = \frac{p_1 M_G}{Z_1 R T_1}, \tag{14}$$

where M_G is the molar mass of gas, Z is the gas compressibility factor, and R is the universal gas constant, 2) the gas expansion across the choke may be assumed adiabatic

$$\frac{1}{\rho_{G,2}} = \frac{1}{\rho_{G,1}} \left(\frac{p_1}{p_2} \right)^{\frac{1}{\kappa}}, \tag{15}$$

3) the liquid is incompressible such that the oil and water densities remain constant across the choke, and 4) the mixture density may be assumed homogeneous

$$\frac{1}{\rho} = \frac{\eta_G}{\rho_G} + \frac{\eta_O}{\rho_O} + \frac{\eta_W}{\rho_W}, \quad \eta_W = 1 - \eta_G - \eta_O. \tag{16}$$

The Sachdeva et al. model distinguishes between sub-critical and critical flow[1] through the choke restriction by defining the pressure ratio as

$$p_r = \begin{cases} \frac{p_2}{p_1} & \frac{p_2}{p_1} \geq p_{r,c} \\ p_{r,c} & otherwise \end{cases} \tag{17}$$

For multiphase flow, a rule of thumb for the critical pressure ratio is $p_{r,c} \approx 0.6$ (Jansen, 2015).

In this article, the Sachdeva et al. choke model will be referred to as the mechanistic model (MM) and we specify the model parameters as

$$\boldsymbol{\phi}_{MM} = [\rho_O, \rho_W, \kappa, M_G, p_{r,c}, C_D]. \tag{18}$$

To hybridize the MM, we introduce a neural network (NN) into the model equations. The NN is a collection of *layers*, where each layer, $i \in \{1, ..K\}$, outputs (z_i) a transformation of the inputs (z_{i-1}). We use a piece-wise linear transformation with weights $\boldsymbol{W}_i$ and bias b_i, and the rectified linear unit (ReLU) as activation function, a.

$$\begin{aligned} \boldsymbol{z}_i &= a(\boldsymbol{W}_i \boldsymbol{z}_{i-1} + b_i) \\ &= \max\{0, \boldsymbol{W}_i \boldsymbol{z}_{i-1} + b_i\}, \quad i \in \{1, ...K\} \end{aligned} \tag{19}$$

In this research, the flow area function of the outlet, $A_2(u)$, will be represented with a neural network

$$A_2 = g(u; \boldsymbol{\phi}_{DD}), \tag{20}$$

where the choke opening is used as input to the network and the network parameters are the collection of weights and biases on all layers $\boldsymbol{\phi}_{DD} = \{(\boldsymbol{W}_1, b_1), ..., (\boldsymbol{W}_K, b_K)\}$. The rest of the mechanistic equations from the MM remain as before. One may think of this model design as a way to alleviate the MM assumption of the shape of the area function. Naturally, relaxation of other MM assumptions such as the real gas law or adiabatic gas expansion is another possible model design. We incorporate the constant discharge coefficient C_D into the above function such that the remaining physical parameters of the HM are

$$\boldsymbol{\phi}_{HM} = [\rho_O, \rho_W, \kappa, M_G, p_{r,c}]. \tag{21}$$

4 Case study - synthetic data

In this case study, we investigate the identifiability and interpretability of the MM and HM presented in Section 3. To have perfect information about the underlying process, we generate noise-free, synthetic data of the oil volumetric flow rate through

[1] Critical flow through a choke occur when a reduction in p_2 for a fixed p_1 does not increase the flow rate (Jansen, 2015).

the choke with the MM from Section 3. A set of realistic combinations of the inputs $\boldsymbol{x}$ taken from two anonymous petroleum fields are used. If the model is identifiable, we should expect the physical parameters to converge to the true underlying parameter values of the data-generating process *without* regularizing the parameters.

Table 1 gives the true parameter values of the data generating process and the prior parameter distributions used for initialization and regularization of the physical model parameters. The neural network depth and width in the HM are chosen 3×100, and the non-physical parameters are initialized with He-initialization (He et al., 2015). Due to the stochastic nature of the optimization algorithm, the models are trained several times, and the median mean absolute error (MAE), absolute percentage error (MAPE), and parameter values are reported. The errors are calculated using an independent test set. Initially, we set σ_ϵ small to obtain $\Pi \approx \mathbf{0}$, resulting in negligible regularization. It turns out that neither the MM nor the HM obtains convergence of the physical parameters to the true underlying values, see the two first columns of Table 2. Furthermore, some of the parameters converge to physically infeasible values. On the other side, through experimentation, it was found that fixing the value of C_D in the MM at its true value resulted in the convergence of the remaining parameters to their true value. Nevertheless, in a real-life setting, the true underlying parameter values are unknown and the process will never be perfectly represented by a model. Further, the available data is often limited. Therefore, in particular for hybrid models with a large number of non-physical parameters, a better approach may be to include all uncertain parameters in the estimation problem instead of fixing some of them and use parameter regularization to ensure the preservation of the physical interpretability of the model. The two last columns of Table 2 illustrates the results of the MM and HM with sufficient regularization of the model parameters. The learned HM area function with and without regularization is illustrated in Fig. 3. Observe how the curve is close to that of the MM, thereby retaining the physical interpretability of the model. Pay in mind, it was not expected that the HM should obtain a lower error than the MM as the underlying model structure is perfectly known and matches the MM. In a practical case, one should expect an overall decrease in error due to the increased model flexibility of the HM.

Table 1: Values of the true parameters and the prior mean and standard deviation.

ϕ	True	μ	σ
ρ_O	760	800	33.3
ρ_W	1010	1025	8.33
κ	1.30	1.32	0.033
M_G	0.021	0.027	0.003
$p_{r,c}$	0.55	0.6	0.067
C_D	1.0	0.9	0.25

Table 2: The median error and parameter values of the MM and HM without regularization for the synthetic data.

	Without reg. MM	Without reg. HM	With reg. MM	With reg. HM
MAE	0.0	0.2	0.0	0.1
MAPE	0.0	0.4	0.0	0.1
ρ_O	979	737	777	772
ρ_W	1302	976	1027	1025
κ	1.31	1.30	1.30	1.30
M_G	0.028	0.021	0.022	0.022
$p_{r,c}$	0.55	0.55	0.55	0.55
C_D	0.88	n.a.	0.99	n.a.

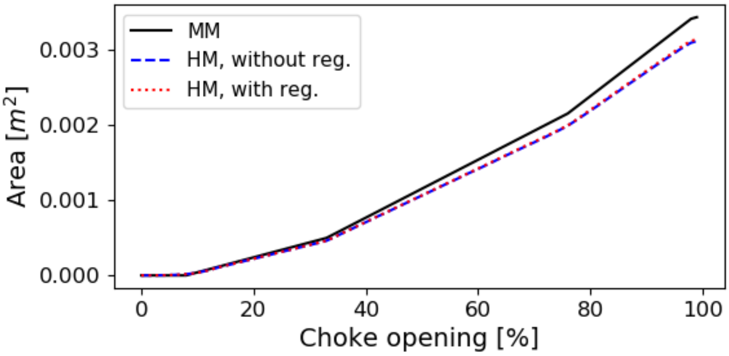


Figure 3: The neural network area function in the HM for different degree of regularization. The mechanistic area function is illustrated for reference.

5 Case study - Edvard Grieg

In this case study, the physical interpretability and predictive performance are investigated for petroleum production choke valve models developed using real production data. We use historical data from six petroleum wells on the asset Edvard Grieg (Lundin Energy Norway, 2020) and train the MM and the HM for each well. The data have been preprocessed in two steps. Firstly, the processing technology in (Grimstad et al., 2016) is used to produce a compressed data set of steady-state operating points suitable for steady-state modeling. Secondly, a set of filters are applied to remove significant outliers such as negative pressures or flow rates. For each well, the historical data is split into training and test data, using the three latest months as test data. This is to mimic an industrial case where the model is used to predict future flow rates.

As in Section 4, the physical parameter initial values are drawn from prior parameter normal distributions, see Table 1, and regularization is applied to enforce convergence of the physical parameters within feasible bounds. As before, the models are trained several times due to the stochasticity of the estimation problem. Table 3 gives the minimum, median, and maximum MAE, MAPE, and parameter values across the six wells grouped on model type. Fig. 4 illustrates the learned neural network area function in the HM for the six wells.

Table 3: The minimum, median, and maximum error and parameter values for the six wells, grouped on model type.

	MM			HM		
	min	med.	max	min	med.	max
MAE	3.4	8.8	19.9	2.3	4.8	11.5
MAPE	7.3	14.0	68.3	2.3	9.1	47.8
ρ_O	608	719	812	634	682	707
ρ_W	1025	1025	1025	1025	1025	1025
κ	1.29	1.33	1.49	1.33	1.35	1.36
M_G	0.027	0.033	0.041	0.029	0.037	0.045
$p_{r,c}$	0.48	0.66	0.93	0.65	0.68	0.88
C_D	0.72	0.84	0.91	n.a.	n.a.	n.a.

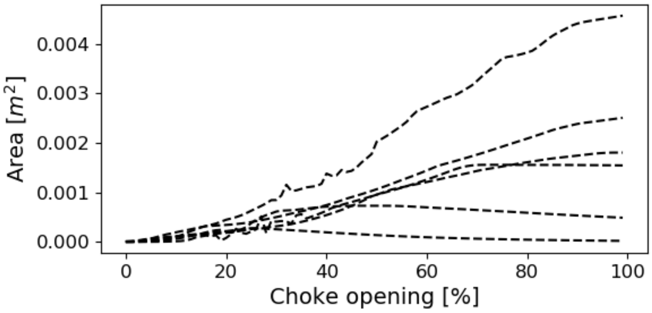


Figure 4: The neural network area function in the HM for the different wells. We see that there are large individual differences for each well.

First of all, notice that the results indicate a significant reduction in prediction error comparing the HM to the MM. Also notice that the values for the physical parameters mostly stay feasible, with some exceptions. On the other hand, there are large variations in the errors for each individual model. The overall best performing model is an HM with 2.3% MAPE whereas the overall worst performing model is an MM with 68.3% MAPE. Furthermore, observe from Fig. 4 that some of the learned area functions are in line with the expected physical trend (an increase in the opening gives an increase in area) whereas other converges to zero.

There may be multiple causes for the large variations. Firstly, if the available data are noisy, faulty, or lacking, the model performance and physical interpretability may be influenced. Investigations of the dataset show that for the wells where the learned neural network area function converges to zero, there are lacking measurements of the choke opening in certain operating regions. Pretraining the neural network could be beneficial in such situations. Further, inappropriate data may easily lead to overfitting of the model parameters and commonly results in poor predictive performance on unseen data.

Secondly, the model structure may be inappropriate. The baseline mechanistic model is a simplified model and may not capture physical effects equally well in all flow regimes and operating conditions. Mechanistic models with higher complexity or models that include effects of the remaining production system such as the wellbore, would likely decrease the error. Further, in this research, the HM only alleviates the mechanistic model assumption of the area function shape. Other assumptions that could be alleviated are for instance the real gas law, the homogeneous mixture density, or the adiabatic gas expansion. Yet, introducing additional data-driven elements could influence model interpretability and should be carefully considered.

Lastly, the utilized process data originate from a non-stationary process, the reservoir. Therefore, using three months as test data in which the model and parameters remain constant may be inappropriate and may cause large prediction errors. Updating the model more frequently, for instance in an online learning fashion, would likely improve the performance.

6 Conclusion

This study argues that model identifiability is challenging to obtain in practice due to both deterministic and stochastic uncertainties. In particular in situations where the underlying process is complex and highly nonlinear, as is the case in petroleum production systems, and where the available data is limited. Even though identifiability is often thought of as a prerequisite for obtaining physically interpretable models, we have illustrated through an industrial case study that physical interpretability may be preserved for non-identifiable hybrid models by the inclusion of sufficient regularization of the model parameters in the estimation problem. Further, hybrid models may improve the predictive performance compared to a mechanistic model due to the increased model flexibility. This is demonstrated in a case study on real production data from six petroleum wells on the asset Edvard Grieg. The hybrid model decreases the median MAPE across the six wells by 35% compared to a mechanistic model while staying physically interpretable. On the other hand, the study shows that the predictive performance and the physical interpretability are influenced by the available data, and there are large variations in the results on a well-level. Future research should look into other hybrid model variants and online learning to improve predictive performance and physical interpretability. Certainly, any general conclusions cannot be drawn as we have only experimented with two different model types and with historical data from one petroleum field. Experimentation with data from other wells and fields would greatly benefit the results in this study.

Acknowledgment

This research is a part of BRU21 - NTNU Research and Innovation Program on Digital and Automation Solutions for the Oil and Gas Industry (`www.ntnu.edu/bru21`) and supported by Lundin Energy Norway.

References

Bikmukhametov, Timur and Johannes Jäschke (2020). “First Principles and Machine Learning Virtual Flow Metering: A Literature Review”. In: *Journal of Petroleum Science and Engineering* 184.

Bottou, L., F.E. Curtis, and J. Nocedal (2018). “Optimization Methods for Large-Scale Machine Learning”. In: *Society for Industrial and Applied Mathematics* 60(2), pp. 223–311.

Brastein, O. M., R. Sharma, and N.-O. Skeie (2019). “Sensor placement and parameter identifiability in grey-box models of building thermal behaviour”. In: *Proc. of The 60th SIMS Conf. on Sim. and Modelling* 170, pp. 51–58.

Brun, R., P. Reichert, and H.R. Künsch (2001). “Practical identifiability analysis of large environmental simulation models”. In: *Water Resources Research* 37(4), pp. 1015–1030.

Chan, T. F. and P.C. Hansen (1992). “Some applications of the rank revealing qr factorization.” In: *SIAM Journal on Scientific and Statistical Computing* 13(3), pp. 727–741.

Deconinck, A.-H. and S. Roels (2017). “Is stochastic grey-box modelling suited for physical properties estimation of building components from on-site measurements?” In: *Journal of Building Physics* 40(5), pp. 444–471.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning.* Cambridge, Massachusetts. London, England.: The MIT Press.

Grimstad, B. et al. (Sept. 2016). “A Simple Data-Driven Approach to Production Estimation and Optimization”. In: *In: SPE Intelligent Energy International Conference and Exhibition.*

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning.* New York, USA: Springer.

He, Kaiming et al. (2015). “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

Hotvedt, M., B. Grimstad, and L. Imsland (2020). “Developing a Hybrid Data-Driven, Mechanistic Virtual Flow Meter - a Case Study”. In: *IFAC-PapersOnLine* 53 (2), pp. 11692–11697.

Jansen, Jan-Dirk (2015). *Nodal Analysis of Oil and Gas Wells - Theory and Numerical Implementation.* TU Delft, The Netherlands: Delft University of Technology.

Kingma, D.P. and J. L. Ba (2015). "Adam: a method for stochastic optimization". In: *International conference on learning representations.*

Klein, A. et al. (2017). "Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets". In: *In: Proceedings of the 20th International Conference on Artificial In- telligence and Statistics (AISTATS)* 54.

Kramer, M. A., M. L. Thomsom, and P. M. Bhagat (1992). "Embedding Theoretical models in neural networks". In: *Proceedings of American Control Conference* 1 (1), pp. 475–479.

Lundin Energy Norway (2020). *Edvard Grieg.* `https://lundin-energy-norway.com/edvard-grieg/`. Acccessed: 18.01.2020.

Psichogios, D.C. and L.H. Ungar (1992). "A Hybrid Neural Network-First Principles Approach to Process Modeling". In: *AIChE Journal* 38(10), pp. 1499–1511.

Raue, A. et al. (2009). "Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood." In: *Bioinfromatics* 25 (15), pp. 1923–1929.

Roscher, R. et al. (2020). "Explainable Machine Learning for Scientific Insights and Discoveries." In: *arXiv:1905.08883v3*, pp. 1–29.

Sachdeva, R. et al. (1986). "Two-phase flow through chokes". In: *Society of Petroleum Engineers, Annual Technical Conference and Exhibition.*

Solle, D. et al. (2016). "Between the Poles of Data-Driven and Mechanistic Modeling for Process Operation". In: *Chemie Ingenieur Technik.*

White, F.M. (2003). *Fluid Mechanics, 5th edition.* New York, USA: McGraw-Hill.

Xu, L. et al. (2011). "Wet gas metering using a revised venturi meter and soft-computing approximation techniques". In: *IEE transactions on instrumentation and measurement* 60 (3), pp. 947–956.

Zendehboudi, S., N. Rezaei, and A. Lohi (2018). "Application of hybrid models in chemical, petroleum, and energy systems: A systematic review". In: *Applied Energy* 228, pp. 2539–2566.

C Bayesian Neural Networks for Virtual Flow Metering: An Empirical Study

Postprint of Grimstad, Hotvedt, et al. (2021)

B. Grimstad, M. Hotvedt, A.T. Sandnes, O. Kolbjørnsen, and L. Imsland (2021). "Bayesian Neural Networks for Virtual Flow Metering: An Empirical Study". In: *Applied Soft Computing* 112 (1)

Bayesian Neural Networks for Virtual Flow Metering: An Empirical Study

B. Grimstad[1,2], M. Hotvedt[1,2], A. T. Sandnes[1,3], O. Kolbjørnsen[3], and L. Imsland[2]

[1]Solution Seeker
[2]Engineering Cybernetics Department, NTNU, Trondheim, Norway
[3]Department of Mathematics, University of Oslo, Norway

Abstract: Recent works have presented promising results from the application of machine learning (ML) to the modeling of flow rates in oil and gas wells. Encouraging results and advantageous properties of ML models, such as computationally cheap evaluation and ease of calibration to new data, have sparked optimism for the development of data-driven virtual flow meters (VFMs). Data-driven VFMs are developed in the small data regime, where it is important to question the uncertainty and robustness of models. The modeling of uncertainty may help to build trust in models, which is a prerequisite for industrial applications. The contribution of this paper is the introduction of a probabilistic VFM based on Bayesian neural networks. Uncertainty in the model and measurements is described, and the paper shows how to perform approximate Bayesian inference using variational inference. The method is studied by modeling on a large and heterogeneous dataset, consisting of 60 wells across five different oil and gas assets. The predictive performance is analyzed on historical and future test data, where an average error of 4-6% and 8-13% is achieved for the 50% best performing models, respectively. Variational inference appears to provide more robust predictions than the reference approach on future data. Prediction performance and uncertainty calibration is explored in detail and discussed in light of four data challenges. The findings motivate the development of alternative strategies to improve the robustness of data-driven VFMs.

Keywords: Neural Network, Bayesian Inference, Variational Inference, Virtual Flow Metering, Heteroscedasticity

1 Introduction

Knowledge of multiphase flow rates is essential to efficiently operate a petroleum production asset. Measured or predicted flow rates provide situational awareness and flow assurance, enable production optimization, and improve reservoir management and planning. However, multiphase flow rates are challenging to obtain with great accuracy due to uncertain subsurface fluid properties and complex multiphase flow dynamics (Jansen, 2015). In most production systems, flow rates are measured using well testing. While these measurements are of high accuracy, they are intermittent and infrequent (Monteiro, Duque, et al., 2020). Some production systems have multiphase flow meters (MPFMs) installed at strategic locations to continuously measure flow rates. Yet, these devices are expensive, and typically have lower accuracy than well testing. An alternative approach is to compute flow rates using virtual flow metering (VFM). VFM is a soft-sensing technology that infers the flow rates in the production system using mathematical models and ancillary measurements (Toskey, 2012). Many fields today use some form of VFM technology complementary to flow rate measurements. There are two main applications of a VFM: i) real-time prediction of flow rates, and ii) prediction of historical flow rates. The second application is relevant to the prediction of missing measurements due to sensor failure or lacking measurements in between well tests.

VFMs are commonly labeled based on their use of either mechanistic or data-driven models (Timur Bikmukhametov and Johannes Jäschke, 2020). Both model types can be either dynamic or steady-state models. Mechanistic VFM models are derived from prior knowledge about the internal structure of the process (Solle et al., 2016). Physical, first-principle laws such as mass, energy, and momentum-balance equations, along with empirical closure relations, are utilized to describe the relationship between the system variables. Mechanistic modeling is the most common approach in today's industry and some commercial VFMs are Prosper, ValiPerformance, LedaFlow, Flow-Manager, and Olga (Amin, 2015).

In contrary to mechanistic models, data-driven models exploit patterns in process data and attempt to find relationships between the system variables with generic mathematical models. In other words, data-driven models attempt to model the process without utilizing explicit prior knowledge (Solle et al., 2016). In recent years, there has been an increasing number of publications on data-driven VFMs (Timur Bikmukhametov and Johannes Jäschke, 2020). The developments are motivated by the increasing amount of sensor data due to improved instrumentation of petroleum fields, better data availability, more computing power, better machine learning tools and more practitioners (Balaji et al., 2018). Additionally, data-driven VFMs may require less maintenance than a mechanistic VFMs (T. A. AL-Qutami, Ibrahim, Ismail, and Ishak, 2017b). Even so, commercial data-driven VFMs are rare. This is arguably due to the following data challenges, which must be overcome by data-driven VFMs:

1. Low data volume
2. Low data variety
3. Poor measurement quality
4. Non-stationarity of the underlying process

The first two challenges are due to data-driven methods, especially neural networks, being data-hungry, and require substantial data volume and variety to achieve high accuracy (Mishra and Datta-Gupta, 2018). Petroleum production data do not generally fulfill these requirements. For petroleum fields without continuous monitoring of the flow rates, new data is obtained at most 1-2 times per month during well testing (Monteiro, Duque, et al., 2020), yielding low data volume. For fields with continuous measurements, the data volume may be higher, yet, the second challenge of low variety remains. Low data variety relates to the way production systems are operated and how it affects the information content in historical production data. The production from a well is often kept fairly constant by the operator, in particular during plateau production, i.e., when the production rate is limited by surface conditions such as the processing capacity. When a field later enters the phase of production decline, the operator compensates for falling pressures and production rates by gradually opening the production choke valves. This can introduce correlations among the measured variables which are unfortunate for data-driven models. A common consequence of modeling in the small data regime is overfitting which decreases the generalization ability of the model, that is, the models struggle with extrapolation to unseen operating conditions (Solle et al., 2016). Nonetheless, one should be able to model the dominant behavior of the well and make meaningful predictions close to the observed data if care is taken to prevent overfitting (T. AL-Qutami et al., 2018).

The third challenge, poor measurement quality, highly influences the predictive abilities of data-driven VFMs. Common issues with measurement devices in petroleum wells include measurement noise, bias, and drift. Additionally, equipment or communication failures may lead to temporarily or permanently missing data. Common practices to improve data quality include device calibration, data preprocessing and data reconciliation (Câmara et al., 2017). In model development, methods such as parameter regularization and model selection techniques prevent overfitting of the model in the presence of noisy data. However, some of the above issues and practices may be challenging to handle in a data-driven model.

Lastly, the underlying process in petroleum production systems, the reservoir, is nonstationary. The pressure in the reservoir decreases as the reservoir is drained and the composition of the produced fluid changes with time (Foss, Knudsen, and Grimstad, 2018). Time-varying boundary conditions make it more difficult to predict future process behavior for data-driven VFMs as they often struggle with extrapolation. As mentioned above, methods to prevent overfitting to the training data in model development may (and should) be utilized to improve extrapolation abilities to the near future, and frequent model updating or online learning would contribute to better predictive abilities for larger changes in the underlying process.

As the above discussion reflects, data-driven VFMs are influenced by uncertainty. Both model (epistemic) uncertainty and measurement (aleatoric) uncertainty are present (Hüllermeier and Waegeman, 2021). The first type originates from the model not being a perfect realization of the true process and there are uncertainties related to the model structure and parameters. The latter type is a cause of noisy data due to imprecision in measurements (Gal, 2016). Accounting for uncertainty is important to petroleum production engineers as they are often concerned with worst- and best-case scenarios. Further, information about the prediction uncertainty may aid the production engineers to decide whether the model predictions may be trusted. According to a recent survey (Timur Bikmukhametov and Johannes Jäschke, 2020), uncertainty estimation must be addressed by future research on VFM.

The motivation of this paper is to address uncertainty by introducing a probabilistic, data-driven VFM based on Bayesian neural networks. With this approach, epistemic uncertainty is modeled by considering the weights and biases of the neural network as random variables. Aleatoric uncertainty can be accommodated by a homoscedastic or heteroscedastic model of the measurement noise. This allows the modeler to separately specify priors related to the two uncertainty types. This can be beneficial when having knowledge of the measurement devices that produced the data modeled on.

Historically, the difficulty of performing Bayesian inference with neural networks has been a hurdle to practitioners. We thus provide a description of how to train the model using variational inference. Variational inference provides the means to perform efficient, approximate Bayesian inference and results in a posterior distribution over the model parameters (Blei, Kucukelbir, and McAuliffe, 2017). The method has shown promising results in terms of quantifying prediction uncertainty on other problems subject to small datasets and dataset shift (Ovadia et al., 2019). We also consider maximum a posteriori estimation, which serves as a non-probabilistic reference method. Although it computes a point estimate of the parameters, as opposed to a posterior distribution, it more closely resembles the maximum likelihood methods used in the majority of previous works on data-driven VFM. The reference method enables us to investigate if a probabilistic method, i.e. variational inference, may improve robustness over a non-probabilistic method. We test the proposed VFM by performing a large-scale empirical study on data from a diverse set of 60 petroleum wells.

The paper is organized as follows. In Section 2 we briefly survey related works on data-driven VFM, with a focus on applications of neural networks. This section also gives some relevant background on probabilistic modeling. In Section 3 we describe how flow rates are measured and the dataset used in the case study. The probabilistic model for data-driven VFM is presented in Section 4 and in Section 5 we discuss methods for Bayesian inference. The case study is presented in Section 6 and discussed in Section 7. In Section 8 we conclude and give our recommendations for future research on data-driven VFM based on our findings.

2 Related work

2.1 Traditional data-driven modeling

In literature, several data-driven methods have been proposed for VFM modeling, for instance, linear and nonlinear regression, principal component regression, random forest, support vector machines and the gradient boosting machine learning algorithm (Zangl, Hermann, and Schweiger, 2014; Bello, Ade-Jacob, and Yuan, 2014; Xu et al., 2011; T. Bikmukhametov and J. Jäschke, 2019). One of the most popular and promising data-driven methods for VFM are neural networks (NN). In (Zangl, Hermann, and Schweiger, 2014), the oil flow rate from three wells was modeled using NNs, and an error as low as 0.15% was reported. However, well-step tests were used to generate data with sufficient variety, and the time-span of the data covered only 30 hours. The three studies, (S. M. Berneti and Shahbazian, 2011; Ahmadi et al., 2013; Hasanvand and S. Berneti, 2015), investigated NNs for the oil flow rate from a reservoir using data samples from 31-50 wells. All used a neural network architecture with one hidden layer and 7 hidden neurons. In the two first, the imperialist competitive algorithm was used to find the NN weights. All of the three studies reported a very small mean squared error, of less than 0.05. Yet, the data was limited to a time-span of 3 months and did not include measurements of the choke openings of the petroleum wells. This will strongly affect the future model performance when reservoir conditions change and the choke openings are adjusted.

A particularly noticeable series of studies on VFM and NN, using historical well measurements with a time-span of more than a year, are (T. AL-Qutami et al., 2018; T. A. AL-Qutami, Ibrahim, Ismail, and Ishak, 2017a; T. A. AL-Qutami, Ibrahim, and Ismail, 2017; T. A. AL-Qutami, Ibrahim, Ismail, and Ishak, 2017b). In (T. A. AL-Qutami, Ibrahim, Ismail, and Ishak, 2017a), the oil and gas flow rates were modeled using two individual feed-forward NN, with one hidden layer and 6 and 7 neurons respectively, and with early stopping to prevent overfitting. An error of 4.2% and 2.3% for the oil and gas flow rates were reported. In (T. A. AL-Qutami, Ibrahim, Ismail, and Ishak, 2017b), a radial basis function network was utilized to model the gas flow rate from four gas condensate wells, and the Orthogonal Least Squares algorithm was applied to find the optimal number of neurons (≤ 80) in the hidden layer of the network. The study reported an error of 5.9%. In (T. A. AL-Qutami, Ibrahim, and Ismail, 2017; T. AL-Qutami et al., 2018), ensemble neural networks were used to excel the learning from sparse data. In the first, the neural network architecture was limited to one hidden layer but the number of hidden neurons was randomly chosen in the range 3-15. Errors of 1.5%, 6.5%, and 4.7% for gas, oil, and water flow rate predictions were achieved. The second paper considered 1-2 hidden layers with 1-25 neurons. Errors of 4.7% and 2.4% were obtained for liquid and gas flow rates respectively.

2.2 Probabilistic modeling

A common approach in today's industry and literature is to study the sensitivity of the model to changes in parameter values, thus to a certain extent approaching epistemic uncertainty, e.g. (Bieker, Slupphaug, and Johansen, 2007; Fonseca, Gonçalves, and Azevedo, 2009; Zangl, Hermann, and Schweiger, 2014; Monteiro, Chaves, et al., 2017; Monteiro, Duque, et al., 2020). By approximating probability distributions for some of the model parameters from available process data and using sampling methods to propagate realizations of the parameters through the model, a predictive distribution of the output with respect to the uncertainty in the parameter may be analyzed.

Probabilistic modeling offers a more principled way to model uncertainty, e.g. by considering model parameters and measurement noise as random variables (Ghahramani, 2015). With Bayesian inference, a posterior distribution of the model output is found that takes into account both observed process data and prior beliefs of the model parameters (Hastie, Tibshirani, and Friedman, 2009). The result is a predictive model that averages over all likely models that fit the data and a model that offers a natural parameter regularization scheme through the use of priors. This is in contrast to traditional data-driven modeling where the concern is often to find the maximum likelihood estimate (Ghahramani, 2015). Although probabilistic models and Bayesian inference are well-known in other fields of research, probabilistic VFMs are rare, yet existent (Lorentzen, Stordal, Nævdal, et al., 2014; Lorentzen, Stordal, Luo, et al., 2016; Luo et al., 2014; Bassamzadeh and Ghanem, 2018).

The following series of studies, (Lorentzen, Stordal, Nævdal, et al., 2014; Luo et al., 2014; Lorentzen, Stordal, Luo, et al., 2016), constructed a mechanistic, probabilistic model of the flow rate in petroleum wellbores. A method for probabilistic, data-driven models is Bayesian neural networks (BNNs). BNNs are similar to traditional neural networks but with each parameter represented with a probability distribution (Hastie, Tibshirani, and Friedman, 2009; Polson and Sokolov, 2017). Bayesian methods have shown to be efficient in finding high accuracy predictors in small data regimes and in the presence of measurement noise without overfitting to the data (Snoek, Larochelle, and Adams, 2012). Further, Bayesian methods lend themselves to online model updating and could quickly improve the model's predictive ability when introduced to new operating regions. Yet, there are disadvantages with probabilistic modeling and Bayesian inference. Except in special cases, inferring the posterior probability distribution of the model consists of solving intractable integrals and inference is slow for large datasets (Blei, Kucukelbir, and McAuliffe, 2017). However, methods for approximation of the posterior distribution exist such as Markov Chain Monte Carlo (MCMC) sampling and variational inference (VI). Comparing these two approximation methods, VI has shown to scale better to large datasets and inference tends to be faster. Additionally, it simplifies posterior updating in the presence of new data. Nevertheless, the approximation with VI is in most cases bounded away from the true distribution, whereas MCMC methods will in principle converge towards the true distribution (Blei, Kucukelbir, and McAuliffe, 2017). A challenge for data-driven probabilistic models, such as Bayesian neural networks, is that the model parameters are generally non-physical, and setting the parameter priors is nontrivial. Despite neural networks being among the more popular data-driven methods for VFM modeling, to the extent of the authors' knowledge, there has been no attempt at using BNNs for VFM. There are, however, examples of BNNs being used for data-driven prediction in similar applications (Liu et al., 2012; Humphrey et al., 2016).

3 Flow rate measurements and dataset

A petroleum production well is illustrated in Figure 1. Produced fluids flow from the reservoir, up to the wellhead, and through the choke valve. The choke valve opening (u) is operated to control the production from the well. The fluids thereafter enter the separator which separates the multiphase flow into the three single phases of oil, gas, and water $\boldsymbol{q} = (q_{\text{oil}}, q_{\text{gas}}, q_{\text{wat}})$. On well-instrumented wells, pressure (p) and temperature (T) is measured upstream and downstream the choke valve.

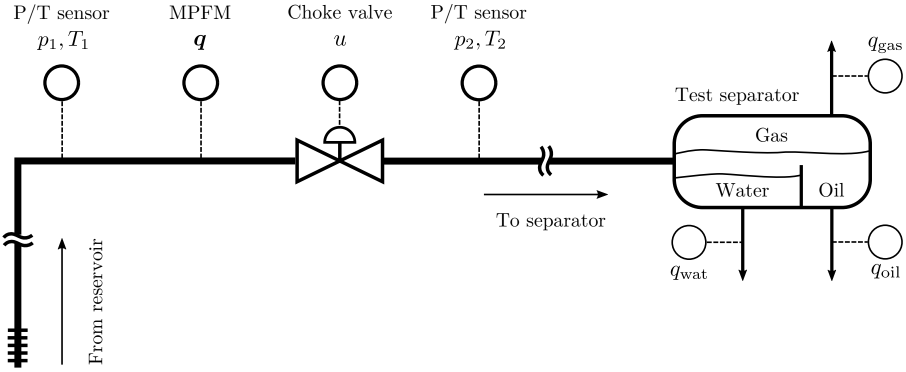


Figure 1: Sensor placement in a typical production well. A MPFM measures multiphase flow rates in the well. During well testing, single phase flow rates are measured with high accuracy after fluid separation.

The two main devices to measure multiphase flow rates in a well are the multiphase flow meter (MPFM) and test separator, both illustrated in Figure 1. MPFMs are complex devices based on several measurement principles and offer continuous measurements of the multiphase flow rate. Unfortunately, MPFMs have limited operation range, struggle with complex flow patterns, and are subject to drift over time (Corneliussen et al., 2005). Additionally, PVT (pressure-volume-temperature) data are used as part of the MPFM calculations and should be accurate and up-to-date for high accuracy MPFM measurements. On the other hand, well-testing is performed by routing the multiphase flow to a test separator whereby the separated flows are measured using single-phase measurement devices over a period of time (typically a few hours). Compared to the MPFM, well tests are performed infrequently, usually 1-2 times a month (Monteiro, Duque, et al., 2020).

Normally, measurements of the multiphase flow rate obtained through well-testing have higher accuracy than the measurements from the MPFMs. This is due to the use of single-phase measurement devices in well-testing. According to (Corneliussen et al., 2005; Marshall and Thomas, 2015), the uncertainty, in terms of mean absolute percentage error, of well tests, may potentially be as low as 2% and 1% for gas and oil respectively, whereas MPFM uncertainty is often reported to be around 10%. The error statistics are calculated with respect to reference measurements. For measurements of pressure, temperature, and choke openings, we assume that the sensors' accuracy is high, typically with an uncertainty of 1% or less, and measurement error in these measurements are therefore neglected.

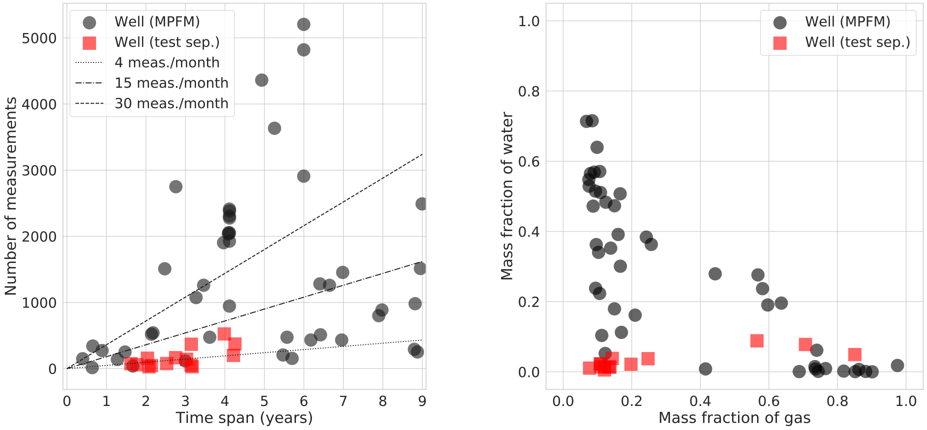
The flow rates are often given as volumetric flow rates under standard conditions, e.g. as standard cubic meter per hour (Sm^3/h). Standard conditions make it easier to compare to reference measurements or measurements at other locations in the process as the volume of the fluid changes with pressure and temperature. Flow rates may be converted from actual conditions to standard conditions using PVT data (Krejbjerg et al., 2019). If the density of the fluid at standard conditions is known, the standard volumetric flow rate may be converted to mass flow rate, and the phasic mass fractions, $\boldsymbol{\eta} = (\eta_{\text{oil}}, \eta_{\text{gas}}, \eta_{\text{wat}})$, may be calculated. We assume steady-state production, frozen flow, and incompressible liquid such that the phasic volumetric flow rate and mass fractions are constant through the system, from the reservoir to the separator.

3.1 Dataset

The dataset used in this study consists of 66 367 data points from 60 wells producing from five oil and gas fields. The dataset was produced from raw measurement data using a data squashing technology (Grimstad et al., 2016). The squashing procedure averages raw measurement data in periods of steady-state operation to avoid short-scale instabilities. The resulting data points, which we refer to as measurements henceforth, are suitable for modeling of steady-state production rates.

For each well we have a sequence of measurements in time. The time span from the first to last measurement is plotted for each well in Figure 2a. The figure shows that the measurement frequency varies from a handful to hundreds of measurements per year. There are 14 wells with test separator measurements, for which the average number of measurements is 163. The other 46 wells have MPFM measurements, and the average number of measurements is 1393. The 60 wells are quite different from each other in terms of produced fluids. Figure 2b illustrates the spread in mass fractions among the wells.

In the following, we model the multiphase flow through the production choke valve,



(a) Measurement frequency

(b) Average mass fractions

Figure 2: The number of measurements is plotted against the time span from the first to last measurement in (a). The average gas and water mass fraction is shown for all wells in (b).

a crucial component in any VFM. We consider ideal conditions, in the sense that all measurements required by a reasonable choke model are available (Hotvedt, Grimstad, and Imsland, 2020). For each well, we collect the corresponding measurements in a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$. We will only consider one well at the time and simply refer to the dataset as $\mathcal{D}$. The target variable is the total volumetric flow rate, $y_i = q_{\text{oil},i} + q_{\text{gas},i} + q_{\text{wat},i} \in \mathbb{R}$, measured *either* by a test separator *or* a MPFM. The explanatory variables,

$$\boldsymbol{x}_i = (u_i, p_{1,i}, p_{2,i}, T_{1,i}, T_{2,i}, \eta_{\text{oil},i}, \eta_{\text{gas},i}) \in \mathbb{R}^7,$$

are the measured choke opening, the pressures and temperatures upstream and downstream the choke valve, and the mass fractions of oil and gas. No experimental set-up was used to affect the data variety; for example, we did not consider step well tests as in (Zangl, Hermann, and Schweiger, 2014).

4 Probabilistic flow model

Consider the following probabilistic model for the total multiphase flow rate:

$$\begin{aligned}
&\left.\begin{aligned}
y_i &= z_i + \epsilon_i \\
z_i &= f(\boldsymbol{x}_i, \boldsymbol{\phi}) \\
s_i &= g(z_i, \boldsymbol{\psi}) \\
\epsilon_i &\sim \mathcal{N}(0, s_i^2)
\end{aligned}\right\} i = 1, \ldots, N, \\
&\boldsymbol{\phi} \sim p(\boldsymbol{\phi}) = \prod_{i=1}^{K_\phi} \mathcal{N}(\phi_i \,|\, a_i, b_i^2), \\
&\boldsymbol{\psi} \sim p(\boldsymbol{\psi}) = \prod_{i=1}^{K_\psi} \mathcal{N}(\psi_i \,|\, c_i, d_i^2),
\end{aligned} \tag{1}$$

where y_i is a measurement of the multiphase flow rate z_i subject to additive measurement noise ϵ_i. The nonlinear dependence of z_i on $\boldsymbol{x}_i$ is approximated by a Bayesian neural network $f(\boldsymbol{x}_i, \boldsymbol{\phi})$ with weights and biases represented by latent (random) variables $\boldsymbol{\phi}$. The neural network is composed of L functions, $f = f^{(L)} \circ \cdots \circ f^{(1)}$, where $f^{(1)}$ to $f^{(L-1)}$ are called the hidden layers of f, and $f^{(L)}$ is the output layer (Goodfellow, Y. Bengio, and Courville, 2016). A commonly used form of a hidden layer l is $f^{(l)}(\boldsymbol{x}) = \mathrm{ReLU}(W^{(l)}\boldsymbol{x} + \boldsymbol{b}^{(l)})$, where the rectified linear unit (ReLU) operator is given as $\mathrm{ReLU}(\boldsymbol{z})_i = \max\{z_i, 0\}$, $W^{(l)}$ is a weight matrix, and $\boldsymbol{b}^{(l)}$ is a vector of biases. For regression tasks the output layer is usually taken to be an affine mapping, $f^{(L)}(\boldsymbol{x}) = W^{(L)}\boldsymbol{x} + \boldsymbol{b}^{(L)}$. The layer weights and biases are collected in $\boldsymbol{\phi} = \{(W^{(l)}, \boldsymbol{b}^{(l)})\}_{l=1}^{L}$ to enable the compact notation $f(\boldsymbol{x}_i, \boldsymbol{\phi})$. With a slight abuse of this notation, an element ϕ_i of $\boldsymbol{\phi}$ represents a scalar weight or bias for $i \in \{1, \ldots, K_\phi\}$, where K_ϕ is the total number of weights and biases in the neural network. The distinguishing feature of a Bayesian neural network is that the weights and biases, $\boldsymbol{\phi}$, are modeled as random variables with a prior distribution $p(\boldsymbol{\phi})$.

We assume the noise to be normally distributed with standard deviation $g(z_i, \boldsymbol{\psi}) > 0$, and we consider different functions g of z_i and latent variables $\boldsymbol{\psi}$. We discuss the priors on the latent variables, $p(\boldsymbol{\phi})$ and $p(\boldsymbol{\psi})$, in the subsequent sections. The probabilistic model is illustrated graphically in Figure 3.

Given $\boldsymbol{\phi}$, $\boldsymbol{\psi}$ and explanatory variables $\boldsymbol{x}$, the conditional flow rate $z = f(\boldsymbol{x}, \boldsymbol{\phi})$ and a measurement y is generated as

$$y \,|\, z, \boldsymbol{\psi} \sim \mathcal{N}(y \,|\, z, g(z, \boldsymbol{\psi})^2). \tag{2}$$

The flow rate measurement y is subject to epistemic (model) uncertainty in $f(\boldsymbol{x}, \boldsymbol{\phi})$ and aleatoric (measurement) uncertainty via $g(z, \boldsymbol{\psi})$. We differ between homoscedastic and heteroscedastic measurement noise. Heteroscedasticity is when the structure of the noise in a signal is dependent on the structure of the signal itself and is more

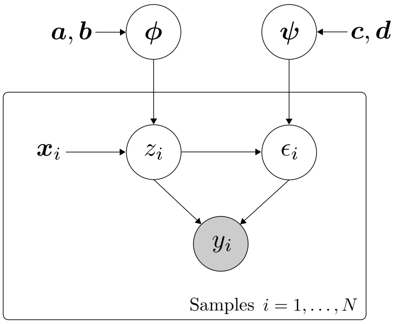

$\boldsymbol{a}, \boldsymbol{b}$
$\boldsymbol{\phi}$
$\boldsymbol{\psi}$
$\boldsymbol{c}, \boldsymbol{d}$
$\boldsymbol{x}_i$
z_i
ϵ_i
y_i
Samples $i = 1, \dots, N$

Figure 3: A probabilistic graphical model for flow rates. Random variables are inscribed by a circle. A gray-filled circle means that the random variable is observed. The dependence $z_i \to \epsilon_i$ indicates that the noise is heteroscedastic, while the dependence $\boldsymbol{\psi} \to \epsilon_i$ indicates that the noise model is learned from data.

difficult to capture (Woodward, Alsberg, and Kell, 1998). Homoscedasticity is the lack of heteroscedasticity.

The flow model in (1) is a quite generic regression model, but it restricts the modeling of the measurement noise. The model allows the noise to be heteroscedastic, with the noise level being a function of the flow rate z, or homoscedastic for which the noise level is fixed. In the latter case, $g(z, \boldsymbol{\psi}) = \sigma_n$, where σ_n is a fixed noise level. If the noise level is unknown, it can be learned with the following homoscedastic noise model:

$$\begin{aligned} g(z_i, \boldsymbol{\psi}) &= \exp(\psi_1), \\ \psi_1 &\sim \mathcal{N}(c_1, d_1^2), \end{aligned} \tag{3}$$

where ψ_1 is a normally distributed latent variable and the noise level is log-normal. The exponential ensures that $g(z_i, \boldsymbol{\psi}) > 0$.

The homoscedastic noise model in (3) may be unrealistic for flow meters with a heteroscedastic noise profile. As described earlier, the uncertainty of the flow rate measurement is often given in relative terms. To model this property of the data, we augment (3) with a multiplicative term to get the following heteroscedastic noise model:

$$\begin{aligned} g(z_i, \boldsymbol{\psi}) &= \exp(\psi_2) \cdot |z_i| + \exp(\psi_1), \\ \psi_1 &\sim \mathcal{N}(c_1, d_1^2), \\ \psi_2 &\sim \mathcal{N}(c_2, d_2^2), \end{aligned} \tag{4}$$

where ψ_1 and ψ_2 are normally distributed latent variables[1]. Both $\exp(\psi_1)$ and $\exp(\psi_2)$ are log-normal, and are hence strictly positive. It follows from $|z| \geq 0$ that the noise standard deviation $g(z, \boldsymbol{\psi}) > 0$.

[1] We assume that we have one flow rate instrument for each well. Yet, several instruments may be handled by having separate noise models for each instrument.

4.1 Prior for the noise model, $p(\boldsymbol{\psi})$

The prior for the noise model is assumed to be a factorized normal

$$p(\boldsymbol{\psi}) = \prod_{i=1}^{K_\psi} \mathcal{N}(\psi_i \,|\, c_i, d_i^2), \tag{5}$$

where $K_\psi = 1$ for the homoscedastic noise model in (3) and $K_\psi = 2$ for the heteroscedastic noise model in (4).

The accuracy of an instrument measuring flow rate is commonly given as a mean absolute percentage error (MAPE) to a reference measurement. More precisely, the expected measurement error is specified as

$$\mathbf{E}_{y \,|\, z}\left[\frac{|y-z|}{|z|}\right] = E_r, \tag{6}$$

where y is the measurement, $z > 0$ is the reference measurement, and E_r is the MAPE, e.g. $E_r = 0.1$ for a MAPE of 10%. We wish to translate such statements to a prior $p(\boldsymbol{\psi})$.

Assuming a perfect reference measurement z, normal noise ϵ, and an additive noise model $y = z + \epsilon$, we obtain from (6) a noise standard deviation $g(z) = \sqrt{\pi/2}E_r|z|$. We recognize this as the first term in the heteroscedastic noise model (4). We derive prior parameters of $\psi_2 \sim \mathcal{N}(c_2, d_2^2)$ that correspond to a log-normal distribution $\exp(\psi_2)$ with mean $\sqrt{\pi/2}E_r$ by solving:

$$c_2 = \log(\sqrt{\pi/2}E_r) - d_2^2/2, \tag{7}$$

where we can adjust the variance d_2^2 to express our uncertainty in the value of E_r.

The specification of a relative measurement error E_r cannot be translated directly to a fixed noise level, as required by the homoscedastic noise model in (3). However, we can obtain a reasonable approximation by using the above procedure. If we set $z = \bar{z}$, where $\bar{z}$ is the mean production of a well, we can calculate prior parameters for ψ_1 as follows:

$$c_1 = \log(\sqrt{\pi/2}E_r\bar{z}) - d_1^2/2. \tag{8}$$

We express our uncertainty about the noise level by adjusting the variance d_1^2.

4.2 Prior for the neural network weights, $p(\boldsymbol{\phi})$

We encode our initial belief of the parameters $\boldsymbol{\phi}$ with a fully factorized normal prior

$$p(\boldsymbol{\phi}) = \prod_{i=1}^{K_\phi} \mathcal{N}(\phi_i \,|\, a_i, b_i^2), \tag{9}$$

where K_ϕ is the number of weights and biases in the neural networks f. We assume a zero mean for the weights and biases, that is $a_i = 0$, as is common practice for neural networks. One interpretation of the prior standard deviations is that they encode the (believed) frequencies of the underlying function, with low values of $\boldsymbol{b}$ inducing slow-varying (low frequency) functions, and high values inducing fast-varying (high frequency) functions (Gal, 2016). While this interpretation can give us some intuition about the effect of the prior, it is not sufficiently developed to guide the specification of a reasonable prior. We refrain from learning the prior from the data (as with empirical Bayes) and therefore treat $\boldsymbol{b}$ as hyperparameters to be prespecified.

For deep neural networks it is common practice to randomly sample the initial weights so that the output has a variance of one for a standard normal distributed input (Glorot and Yoshua Bengio, 2010; He et al., 2015). For example, He-initialization (He et al., 2015) is often used for neural networks with ReLU activation functions. With He-initialization, the weights of layer l are drawn from the distribution $\mathcal{N}(0, \sigma_l^2)$ with $\sigma_l = \sqrt{2/n_l}$, where n_l is the number of layer inputs. The weights in the first hidden layer are initialized with $\sigma_l = \sqrt{1/n_l}$ since no ReLU activation is applied to the network's input. With layer biases set to zero, this initialization scheme yields a unit variance for the output.

The objective of weight initialization is similar to that of prior specification; a goal in both settings is to find a good initial model. In this work, we use the standard deviations $b_i = \sigma_l$ as a starting point for the prior specification (for weight i in layer l of a ReLU network). We call this the He-prior. The resulting standard deviations can then be increased (or decreased) if one believes that the underlying function amplifies (or diminishes) the input signal.

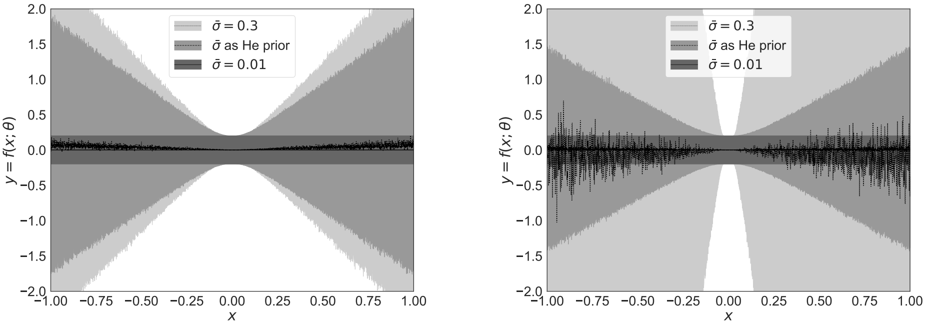
Figure 4 shows the effect of $\boldsymbol{b}$ on the predictive uncertainty of a Bayesian neural network. With a common prior standard deviation (same for all weights), the output variance is sensitive to the network size (depth and width). This sensitivity complicates the prior specification, as illustrated for different network depths in the figure. The He-prior retains a unit output variance for different network sizes.

4.3 A fully factorized normal prior on the latent variables

The prior of model (1) is a fully factorized normal distribution, $p(\boldsymbol{\phi})p(\boldsymbol{\psi})$. To simplify the notation in the rest of this paper we collect the latent variables in $\boldsymbol{\theta} = (\boldsymbol{\phi}, \boldsymbol{\psi}) \in \mathbb{R}^K$, where $K = K_\phi + K_\psi$. This allows us to state the prior on $\boldsymbol{\theta}$ as $p(\boldsymbol{\theta}) = p(\boldsymbol{\phi})p(\boldsymbol{\psi})$, where

$$p(\boldsymbol{\theta}) = \prod_{i=1}^{K} \mathcal{N}(\theta_i \,|\, \bar{\mu}_i, \bar{\sigma}_i^2), \tag{10}$$

with means $\bar{\boldsymbol{\mu}} = (\bar{\mu}_1, \dots, \bar{\mu}_K) = (a_1, \dots, a_{K_\phi}, c_1, \dots, c_{K_\psi}) \in \mathbb{R}^K$ and standard deviations $\bar{\boldsymbol{\sigma}} = (\bar{\sigma}_1, \dots, \bar{\sigma}_K) = (b_1, \dots, b_{K_\phi}, d_1, \dots, d_{K_\psi}) \in \mathbb{R}^K$. The total number of model parameters ($\bar{\boldsymbol{\mu}}$ and $\bar{\boldsymbol{\sigma}}$) is $2K$.



(a) Two hidden layers, each with 50 nodes

(b) Five hidden layers, each with 50 nodes

Figure 4: Prediction uncertainty (two sigma) for different priors $b_i = \bar{\sigma}$ on a neural network's weights. Two networks are trained on a dataset $\mathcal{D} = \{(0, y_i)\}_{i=1}^{100}$, where $y_i \sim \mathcal{N}(0, \sigma_n^2)$ and the noise level $\sigma_n = 0.1$ is known. The figure shows that the epistemic (model) uncertainty is explained away for $x = 0$ and increasing with the distance to $x = 0$. Away from the data, the increase in epistemic uncertainty depends on the prior variance and network depth.

5 Methods

We wish to infer the latent variables $\boldsymbol{\theta}$ of the flow rate model in (1) from observed data. With Bayesian inference, the initial belief of $\boldsymbol{\theta}$, captured by the prior distribution $p(\boldsymbol{\theta})$ in (10), is updated to a posterior distribution $p(\boldsymbol{\theta} \,|\, \mathcal{D})$ after observing data $\mathcal{D}$. The update is performed according to Bayes' rule:

$$p(\boldsymbol{\theta} \,|\, \mathcal{D}) = \frac{p(\mathcal{D} \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D})}, \tag{11}$$

where $p(\mathcal{D})$ is the evidence and the likelihood is given by

$$p(\mathcal{D} \,|\, \boldsymbol{\theta}) = \prod_{i=1}^{N} p(y_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}). \tag{12}$$

The log-likelihood of the model in (1) is shown in A.1.

From the posterior distribution, we can form the predictive posterior distribution

$$p(y^+ \,|\, \boldsymbol{x}^+, \mathcal{D}) = \int p(y^+ \,|\, \boldsymbol{x}^+, \boldsymbol{\theta}) p(\boldsymbol{\theta} \,|\, \mathcal{D}) d\boldsymbol{\theta} \tag{13}$$

to make a prediction y^+ for a new data point $\boldsymbol{x}^+$.

The posterior in (11) involves intractable integrals that prevents a direct application of Bayes' rule (Blei, Kucukelbir, and McAuliffe, 2017). In the following sections, we review two methods that circumvent this issue, namely maximum a posteriori (MAP) estimation and variational inference. With MAP estimation inference is simplified by considering only the mode of $p(\boldsymbol{\theta} \,|\, \mathcal{D})$, and with variational inference the posterior distribution is approximated. In the latter case, we can form an approximated predictive posterior distribution by replacing the posterior in (13) with its approximation. Statistics of this distribution, such as the mean and variance, can be estimated using Monte-Carlo sampling (Gal, 2016).

5.1 MAP estimation

With maximum a posteriori (MAP) estimation we attempt to compute:

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta} \,|\, \mathcal{D}), \tag{14}$$

where $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ is the mode of the posterior distribution in (11). For the model in (1) with a fixed and constant noise variance σ_n^2 and $\bar{\sigma}_i^2$ is the (prior) variance of θ_i, we have that

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{\text{MAP}} &= \arg\max_{\boldsymbol{\theta}} \; \log p(\mathcal{D} \,|\, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \\ &= \arg\min_{\boldsymbol{\theta}} \; \frac{1}{2\sigma_n^2} \sum_{i=1}^{N} (y_i - f(\boldsymbol{x}_i, \boldsymbol{\theta}))^2 + \sum_{i=1}^{K} \frac{1}{2\bar{\sigma}_i^2} \theta_i^2, \end{aligned} \tag{15}$$

From (15), we see that MAP estimation is equivalent to maximum likelihood estimation with L^2-regularization (Hastie, Tibshirani, and Friedman, 2009).

While MAP estimation allows us to incorporate prior information about the model, it provides only a point estimate $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ and will not capture the epistemic uncertainty of the model. To obtain a posterior distribution of $\boldsymbol{\theta}$ we consider the method of variational inference.

5.2 Variational inference

With variational inference, the posterior in (11) is approximated by solving an optimization problem, cf. (Blei, Kucukelbir, and McAuliffe, 2017). Consider a variational posterior density $q(\boldsymbol{\theta} \,|\, \boldsymbol{\lambda})$, parameterized by a real vector $\boldsymbol{\lambda}$. The objective of the optimization is to find a density $q^\star = q(\boldsymbol{\theta} \,|\, \boldsymbol{\lambda}^\star)$ that minimizes the Kullback-Leibler (KL) divergence to the exact posterior, i.e.

$$\boldsymbol{\lambda}^\star = \arg\min_{\boldsymbol{\lambda}} D_{\text{KL}} \left(q(\boldsymbol{\theta} \,|\, \boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta} \,|\, \mathcal{D}) \right). \tag{16}$$

A direct approach to solve (16) is not practical since it includes the intractable posterior. In practice, the KL divergence is instead minimized indirectly by maximizing the evidence lower bound (ELBO):

$$\mathcal{L}(\boldsymbol{\lambda}) := \log p(\mathcal{D}) - D_{\mathrm{KL}}\left(q(\boldsymbol{\theta} \,|\, \boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta} \,|\, \mathcal{D})\right) \tag{17}$$

$$= \mathbf{E}_q\left[\log p(\mathcal{D}|\boldsymbol{\theta})\right] - D_{\mathrm{KL}}\left(q(\boldsymbol{\theta} \,|\, \boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta})\right), \tag{18}$$

where the expectation $\mathbf{E}_q[\cdot]$ is taken with respect to $q(\boldsymbol{\theta} \,|\, \boldsymbol{\lambda})$. From the ELBO loss in (18), we see that an optimal variational distribution maximizes the expected log-likelihood on the dataset, while obtaining similarity to the prior via the regularizing term $D_{\mathrm{KL}}\left(q(\boldsymbol{\theta} \,|\, \boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta})\right)$.

5.2.1 Stochastic gradient variational Bayes

Stochastic gradient variational Bayes (SGVB) or Bayes by backprop is an efficient method for gradient-based optimization of the ELBO loss in (18), cf. (Diederik P Kingma and Welling, 2014; Blundell et al., 2015).

Suppose that the variational posterior $q(\boldsymbol{\theta} \,|\, \boldsymbol{\lambda})$ is a mean-field (diagonal) normal distribution with mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$. Let the variational parameters be $\boldsymbol{\lambda} = (\boldsymbol{\mu}, \boldsymbol{\rho})$ and compute $\boldsymbol{\sigma} = \log(1 + \exp(\boldsymbol{\rho}))$, where we use an elementwise softplus mapping to ensure that $\sigma_i > 0$.

The basic idea of SGVB is to reparameterize the latent variables to $\boldsymbol{\theta} = h(\boldsymbol{\zeta}, \boldsymbol{\lambda}) = \boldsymbol{\mu} + \log(1 + \exp(\boldsymbol{\rho})) \circ \boldsymbol{\zeta}$, where $\circ$ denotes pointwise multiplication and $\boldsymbol{\zeta} \sim \mathcal{N}(0, I)$. With this formulation, the stochasticity of $\boldsymbol{\theta}$ is described by a standard normal noise $\boldsymbol{\zeta}$ which is shifted by $\boldsymbol{\mu}$ and scaled by $\boldsymbol{\sigma}$. The reparameterization allows us to compute the gradient of the ELBO (18) as follows:

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}) &= \nabla_{\boldsymbol{\lambda}} \mathbf{E}_q\left[\log p(\mathcal{D}|\boldsymbol{\theta})\right] - \nabla_{\boldsymbol{\lambda}} D_{\mathrm{KL}}\left(q(\boldsymbol{\theta} \,|\, \boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta})\right) \\ &= \mathbf{E}_{\zeta}\left[\nabla_{\boldsymbol{\theta}} \log p(\mathcal{D} \,|\, \boldsymbol{\theta}) \nabla_{\boldsymbol{\lambda}} h(\boldsymbol{\zeta}, \boldsymbol{\lambda})\right] - \nabla_{\boldsymbol{\lambda}} D_{\mathrm{KL}}\left(q(\boldsymbol{\theta} \,|\, \boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta})\right)\end{aligned} \tag{19}$$

The expectation in (19) can be approximated by Monte-Carlo sampling the noise: $\boldsymbol{\zeta}_i \sim \mathcal{N}(0, I)$ for $i = 1, \ldots, M$. If we also approximate the likelihood by considering a mini-batch $\mathcal{B} \subset \mathcal{D}$ of size $B \leq N$, we obtain the unbiased SGVB estimator of the ELBO gradient:

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}) \simeq \nabla_{\boldsymbol{\lambda}} \hat{\mathcal{L}}(\boldsymbol{\lambda}) := \frac{N}{B} \frac{1}{M} \sum_{i=1}^{M} \nabla_{\boldsymbol{\theta}} \log p(\mathcal{B} \,|\, \boldsymbol{\theta}) \nabla_{\boldsymbol{\lambda}} h(\boldsymbol{\zeta}_i, \boldsymbol{\lambda}) \\ - \nabla_{\boldsymbol{\lambda}} D_{\mathrm{KL}}\left(q(\boldsymbol{\theta} \,|\, \boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta})\right).\end{aligned} \tag{20}$$

An advantage with the SGVB estimator in (20) is that we can utilize the gradient of the model $\nabla_{\boldsymbol{\theta}} \log p(\mathcal{B}|\boldsymbol{\theta})$ as computed by back-propagation. When both the variational posterior and prior are mean-field normals, as is the case for our model, $D_{\mathrm{KL}}\left(q(\boldsymbol{\theta} \,|\, \boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta})\right)$ can be computed analytically as shown in A.2.

In Algorithm 1 we summarize the basic SGVB algorithm for mean-field normals and Monte-Carlo sample size of $M = 1$. We finally note that for variables representing weights of a neural network, we implement the local reparameterization trick in (Diederik P. Kingma, Salimans, and Welling, 2015) to reduce gradient variance and save computations (not shown in Algorithm 1).

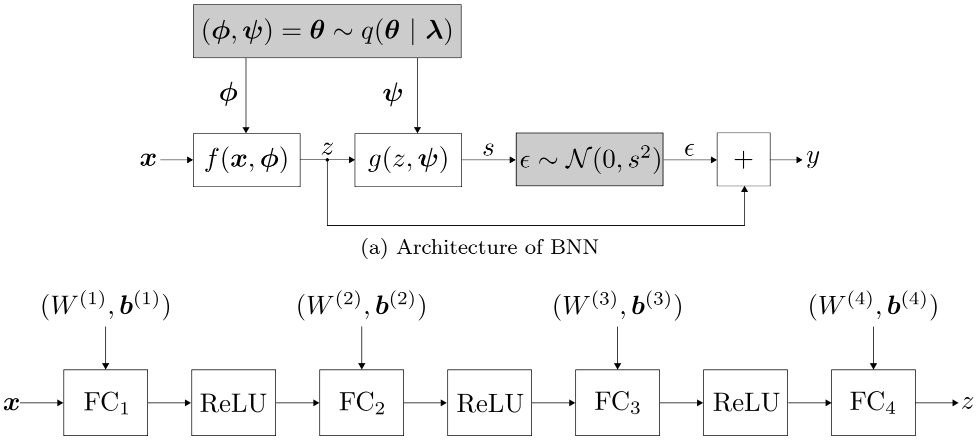
Algorithm 1 Basic implementation of SGVB for mean-field normals ($M = 1$)

```
Require: data $\mathcal{D}$, model $p(\mathcal{D}, \boldsymbol{\theta}) = p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$, parameters $\boldsymbol{\lambda} = (\boldsymbol{\mu}, \boldsymbol{\rho})$, learning
    rate $\alpha$.
repeat
    Sample mini-batch $\mathcal{B}$ from $\mathcal{D}$
    Sample $\boldsymbol{\zeta} \sim \mathcal{N}(0, I)$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\mu} + \log(1 + \exp(\boldsymbol{\rho})) \circ \boldsymbol{\zeta}$
    Compute $\nabla_{\boldsymbol{\lambda}} \hat{\mathcal{L}}(\boldsymbol{\lambda})$ using (20)
    $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \alpha \nabla_{\boldsymbol{\lambda}} \hat{\mathcal{L}}(\boldsymbol{\lambda})$
until no improvement in ELBO
return $\boldsymbol{\lambda}$
```

6 Case study

The goal of the case study was to investigate the predictive performance and generalization ability of the proposed VFM. The study was designed to test the predictive performance on historical data and on future data, which reflect the two main applications of a VFM. If the models generalize well, a similar performance across all wells for each model type should be expected on both historical and future data. To cast light on the data challenges in Section 1, the results differentiate between wells with test separator and MPFM measurements, which have different measurement accuracy and frequency. The prediction uncertainty of the models was also analyzed and the effect of training set size on prediction performance was investigated.

The probabilistic flow rate models in Section 4 were developed using the dataset described in Section 3.1. The conditional mean flow rate, $f(\boldsymbol{x}, \boldsymbol{\phi})$, was modeled using a feed-forward neural network. Three different noise models were considered: a homoscedastic model with fixed noise standard deviation $g(z, \boldsymbol{\psi}) = \sigma_n = \text{const.}$, a homoscedastic model with learned noise standard deviation (3), and a heteroscedastic model with learned noise standard deviation (4). For each of the three model types and the 60 wells in the dataset, the neural network was trained using the SGVB method in Section 5.2.1. These models will be referred to by the label VI-NN. For comparison, a neural network for each of the 60 wells was trained using the MAP estimation method in Section 5.1. For these models we considered the measurement noise to be homoscedastic with a fixed noise standard deviation (σ_n). We label these models as MAP-NN. The He-prior was used for the hidden layers to initialize and regularize the parameters, see Section 4.2. For the noise models, we set the priors as described in Section 4.1, differentiating between wells with MPFM and test separator measurements.


$(\boldsymbol{\phi}, \boldsymbol{\psi}) = \boldsymbol{\theta} \sim q(\boldsymbol{\theta} \mid \boldsymbol{\lambda})$
$\boldsymbol{\phi}$
$\boldsymbol{\psi}$
$\boldsymbol{x}$
$f(\boldsymbol{x}, \boldsymbol{\phi})$
z
$g(z, \boldsymbol{\psi})$
s
$\epsilon \sim \mathcal{N}(0, s^2)$
ϵ
$+$
y
$(W^{(1)}, \boldsymbol{b}^{(1)})$
$(W^{(2)}, \boldsymbol{b}^{(2)})$
$(W^{(3)}, \boldsymbol{b}^{(3)})$
$(W^{(4)}, \boldsymbol{b}^{(4)})$
$\boldsymbol{x}$
FC_1
ReLU
FC_2
ReLU
FC_3
ReLU
FC_4
z

(a) Architecture of BNN

(b) Composition of $f(\boldsymbol{x}, \boldsymbol{\phi})$

Figure 5: The architecture of the BNNs used in this study is illustrated in (a). Probabilistic computations are colored grey. Variables $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are drawn from the approximate posterior and used to compute the conditional mean flow rate, $f(\boldsymbol{x}, \boldsymbol{\phi})$, and noise standard deviation, $g(z, \boldsymbol{\psi})$. The composition of $f(\boldsymbol{x}, \boldsymbol{\phi})$ with four layers (three hidden) and $\boldsymbol{\phi} = \{(W^{(l)}, \boldsymbol{b}^{(l)})\}_{l=1}^{4}$ is shown in (b). Fully connected blocks perform the operation $\mathrm{FC}_l(\boldsymbol{x}) = W^{(l)}\boldsymbol{x} + \boldsymbol{b}^{(l)}$.

A schematic representation of the Bayesian neural network is shown in Figure 5. The network architecture was fixed to three hidden layers, each with 50 nodes to which we apply the ReLU activation function (Glorot, Bordes, and Yoshua Bengio, 2011). Using practical recommendations in (Y. Bengio, 2012), the network architecture may be large as long as regularization is used to prevent overfitting. The Adam optimizer (Diederik P. Kingma and Ba, 2015) with the learning rate set to 0.001 was used to train all networks. Early stopping with a validation dataset was used to determine an appropriate number of epochs to train the models to avoid overfitting (Goodfellow, Y. Bengio, and Courville, 2016). The hyper-parameters were chosen by experimentation and using best practices. The models were implemented and trained using PyTorch (Paszke et al., 2019).

6.1 Prediction performance on historical data

To examine the predictive performance on historical data, a three months long period of contiguous data located in the middle of the dataset, when ordered chronologically, was set aside for testing. The rest of the data was used to train the models. During model development, a random sample of 20% of the training data was used for model validation. The performance of each model type across the 60 wells was analyzed.

Table 1 shows the P_{10}, P_{25}, P_{50} (median), P_{75}, and P_{90} percentiles of the MAPE across all wells. Detailed results which differentiate between test separator and MPFM measurements are reported in B, Table 4.

Table 1: Prediction performance in terms of mean absolute percentage error on historical test data. The percentiles show the variation in performance among all wells.

Method and model	P_{10}	P_{25}	P_{50}	P_{75}	P_{90}
MAP-NN fixed homosc.	1.8	2.8	5.1	8.3	16.0
VI-NN fixed homosc.	1.4	2.6	4.8	8.5	12.8
VI-NN learned homosc.	1.3	2.4	5.3	8.4	13.3
VI-NN learned heterosc.	1.7	3.5	5.9	9.7	11.5

The results show that the four model types achieve similar performance to each other for the 75th and lower percentiles. The median MAPEs (P_{50}) lie in the range 4-6%. A comparison of the 90th percentile performance indicates that models trained by variational inference are more robust in terms of modeling difficult wells. Regardless of the model type used, there are large variations in the performance on different wells, as seen by comparing the 10th and 90th percentiles. The best performing model achieved an error of 0.3% for one of the wells. Yet, some models obtain an unsatisfactory large error. The overall worst-performing model (MAP-NN) achieved an error of 72.1% for one of the wells.

The cumulative performance of the four models is plotted in Figure 6. The cumulative performance plot shows the percentage of test points that fall within a certain percent deviation from the actual measurements (Corneliussen et al., 2005). The figure shows that the models perform better on wells with MPFM measurements than on wells with test separator measurements. Again, similar performance of the four model types is observed.

6.2 Prediction performance on future data

The last three months of measurements were used to test the predictive performance on future data. The rest of the data was used to train the models. During model development, a random sample of 20% of the training data was used for model validation. Table 2 shows the percentiles of the MAPE for the different models on all 60 wells. Detailed results which differentiate between MPFM and test separator measurements are given in B, Figure 5.

Similarly to the case with historical test data, the performance of the four model types is comparable for the 50th and lower percentiles. The median MAPEs (P_{50}) lie in the range 8-13%. For all model types, the 25% best-performing models achieved a MAPE of less than 6%. The best performing model obtained a MAPE of 1.1% on one of the wells. This is in line with some of the best reported results in the literature; see Section 2.1. Nevertheless, for each model type there is a large variation

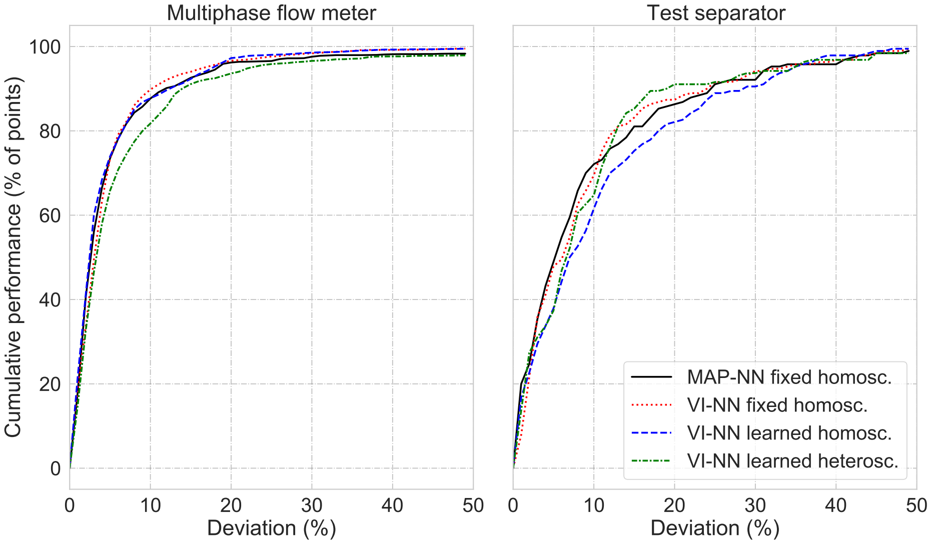


Figure 6: Cumulative performance of the four models on historical test data. The cumulative performance is shown for wells with (left) MPFM and (right) test separator measurements.

Table 2: Prediction performance in terms of mean absolute percentage error on future test data. The percentiles show the variation in performance among all wells.

Method and model	P_{10}	P_{25}	P_{50}	P_{75}	P_{90}
MAP-NN fixed homosc.	3.7	5.6	12.4	24.1	40.0
VI-NN fixed homosc.	4.0	5.6	9.6	18.2	29.3
VI-NN learned homosc.	4.0	6.0	8.9	22.5	32.5
VI-NN learned heterosc.	4.0	5.0	9.2	15.7	24.3

in performance among wells. The overall worst performing model achieved a MAPE of 48.7%.

Comparing the performance for either the 75th or 90th percentile again indicates that models trained by variational inference are more robust in terms of modeling difficult wells. In this regard, the heteroscedastic VI-NN performs particularly well compared to the other model types.

As seen from the cumulative performance plot in Figure 7, the four model types have similar performance to each other. The exception is the heteroscedastic VI-NN, which outperforms the other model types for wells with test separator measurements. As seen in the case of historical test data, the models perform better on wells with MPFM measurements than on well with test separator measurements.

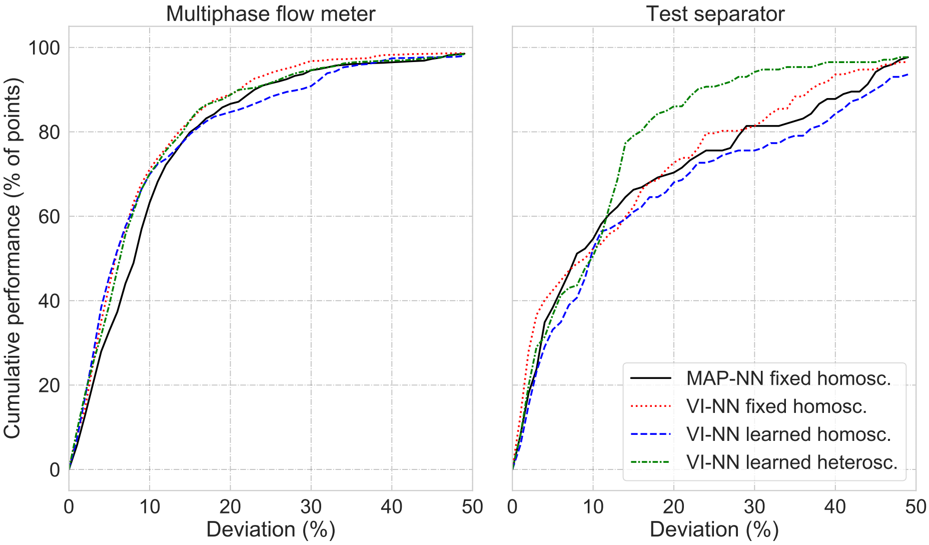


Figure 7: Cumulative performance of the four models on future test data. The cumulative performance is shown for wells with (left) MPFM and (right) test separator measurements.

6.3 Comparison of performance on historical and future data

A comparison of the MAPEs on historical and future data is illustrated in Figure 8. The plots differentiate wells with MPFM and test separator measurements. In general, the prediction error is larger on future test data than on historical test data. There is also a larger variance in the performance on future test data. This indicates that it is harder to make predictions on future data, than on historical data. Further, observe that the errors are smaller for the wells with MPFM measurements than for the wells with test separator measurements in both the historical and future test data case.

6.4 Uncertainty quantification and analysis

In contrary to the MAP-NN models, the VI-NN models quantify the uncertainty in their predictions. To study the quality of the prediction uncertainty, we generated a calibration plot for the three different noise models using the test datasets from

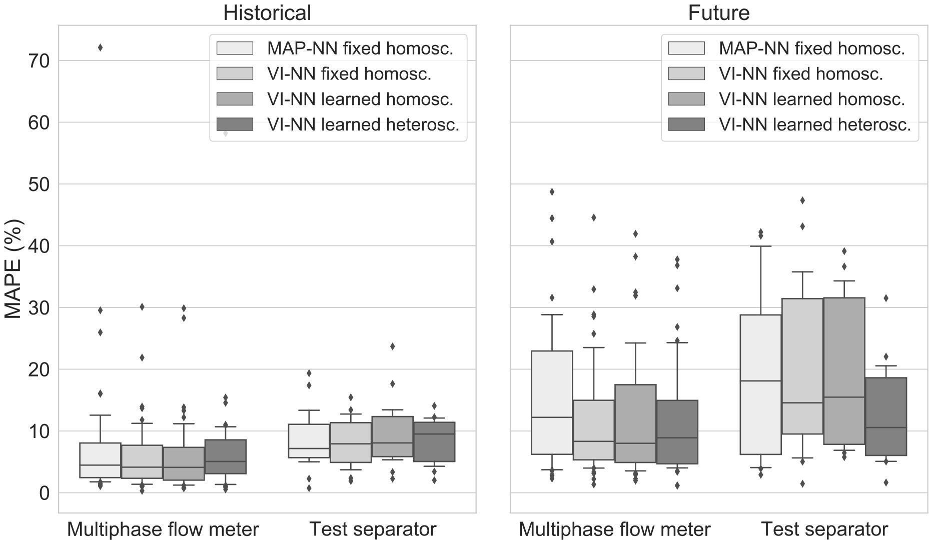
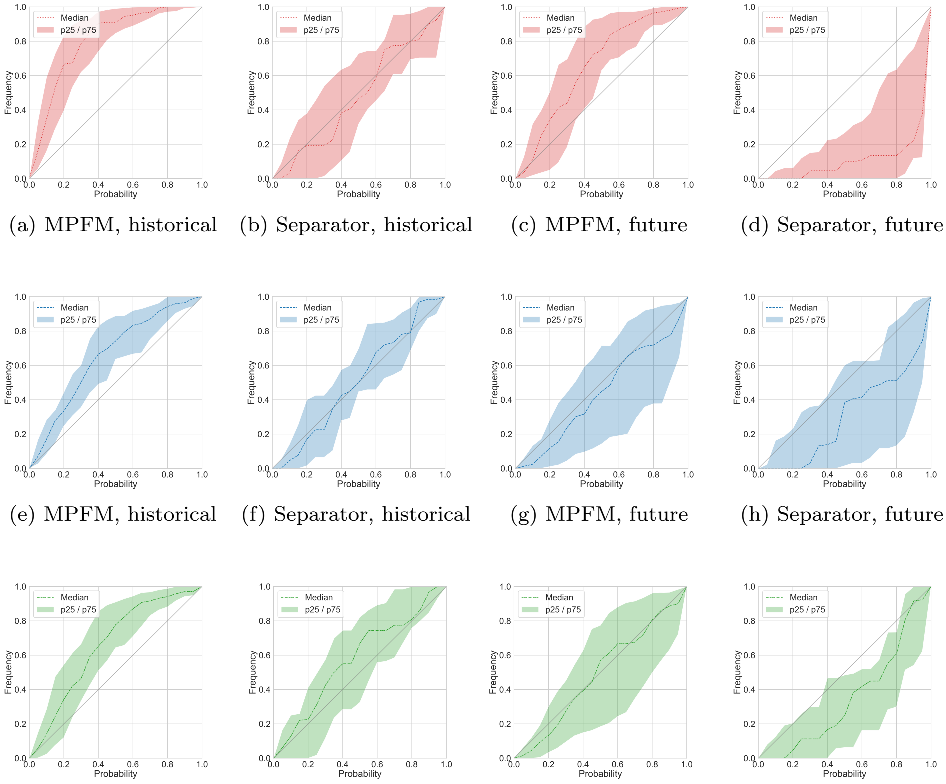


Figure 8: Comparison of performance on historical and future data for the different models. The box plots differentiate between wells with multiphase flow meter and test separator measurements. The boxes show the P_{25}, P_{50} (median), and P_{75} percentiles. The whiskers show the P_{10} and P_{90} percentiles.

Section 6.1 and 6.2; see Figure 9. The plot shows the frequency of residuals lying within varying posterior intervals. For instance, for a perfectly calibrated model, 20% of the test points is expected to lie in the 20% posterior interval centered about the posterior mean. In other words, the calibration curve of a perfectly calibrated model will lie on the diagonal gray line illustrated in the figures. The calibration of a model may vary across wells. To visualize the variance in model calibration, we have illustrated the (point-wise) 25th and 75th percentiles of the calibration curves obtained across wells.

On historical data, the models trained on test separator measurements seem to be best calibrated. The models trained on MPFM measurements overestimate the uncertainty in their predictions. On future data, the results are reversed. The models trained on MPFM measurements are better calibrated and the models trained on test separator measurements all underestimate the prediction uncertainty. Overall, the calibration improves when the noise model is learned. This is seen clearly when comparing the fixed homoscedastic noise to the learned heteroscedastic noise model. The results are summarized in Table 3, which shows the coverage probabilities for the 95% posterior interval (using the point-wise median in the calibration plots).



(a) MPFM, historical (b) Separator, historical (c) MPFM, future (d) Separator, future

(e) MPFM, historical (f) Separator, historical (g) MPFM, future (h) Separator, future

(i) MPFM, historical (j) Separator, historical (k) MPFM, future (l) Separator, future

Figure 9: Calibration plots for fixed homoscedastic noise (a-d), learned homoscedastic noise (e-h), and learned heteroscedastic noise (i-l). Wells are grouped by measurement device, multiphase flow meter or test separator, and the calibration on historical test data (Section 6.1) and future test data (Section 6.2) are shown. The median frequency is shown as a dashed line for each posterior interval (x-axis). The 25th and 75th percentiles (colored bands) show the variation in calibration across wells. A perfectly calibrated model would lie on the diagonal line $y = x$.

6.5 Effect of training set size on prediction performance

When analyzing the prediction performance of the four model types in Section 6.1 and 6.2, it was noticed that the prediction error tended to decrease as the training set size increased. This is illustrated in Figure 10, which shows the MAPEs for the different models and corresponding regression lines with negative slopes. This tendency is generally expected of machine learning models. On the other hand, previous studies such as (T. AL-Qutami et al., 2018), indicate that model performance does not necessarily

Table 3: Coverage probability (95%)

Case	Method and model	Test sep. (%)	MPFM (%)
Future prediction	VI-NN fixed homosc.	37.5	99.5
	VI-NN learned homosc.	81.0	87.7
	VI-NN learned heterosc.	92.3	90.0
Historical prediction	VI-NN fixed homosc.	92.4	100.0
	VI-NN learned homosc.	98.5	99.1
	VI-NN learned heterosc.	100.0	97.2

improve when including data that is several years old. To closer inspect this effect, we compared models developed on successively larger training sets.

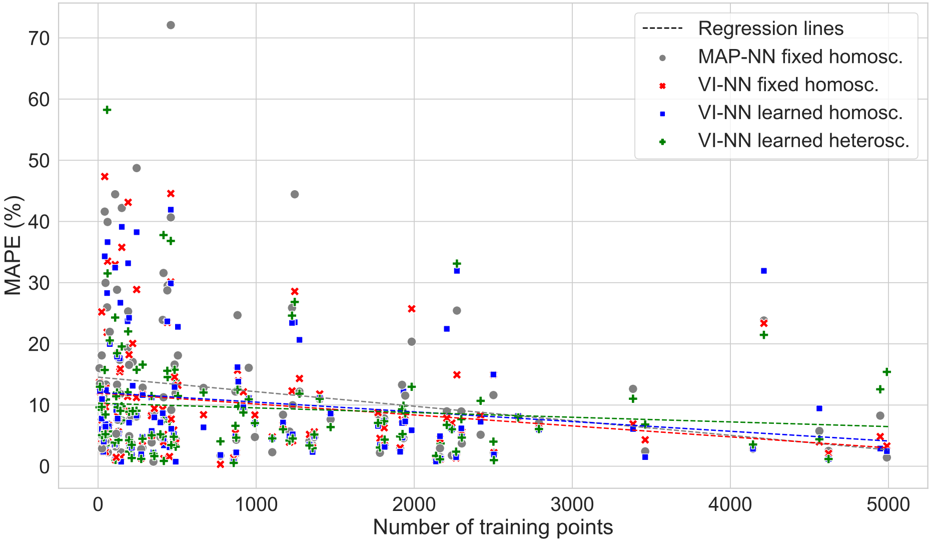


Figure 10: The plot shows the mean absolute percentage error of the four models on historical and future test data for all wells. A regression line for each model shows the tendency of the error as the number of training points varies.

To allow for an interesting range of dataset sizes a subset of 21 wells with 1200 or more MPFM measurements was considered. In a number of trials, a well from the subset and an instant of time at which to split the dataset into a training and test set, were randomly picked. Keeping the test set fixed, a sequence of training sets of increasing size was generated. The training sets were extended backwards in time with data preceding the test data. The following training set sizes were considered: 150, 200, 300, . . ., 1100, where the increment is 100 between 300 and 1100. A MAP-

NN model was developed for each of these training sets, using early stopping and validating against the last 100 data points. The test set size was also set to 100 data points, spanning on average 90 days of production.

Denoting the test MAPE of the models by E_{150}, E_{200}, E_{300}, ..., E_{1100}, we computed relative MAPEs

$$R_k = \frac{E_k}{E_{150}}, \text{ for } k \in \{150, 200, 300, \ldots, 1100\}. \tag{21}$$

The relative errors indicate how the performance develops as the training set size increases, with a baseline at $R_{150} = 1$. The result of 400 trials is shown in Figure 11.

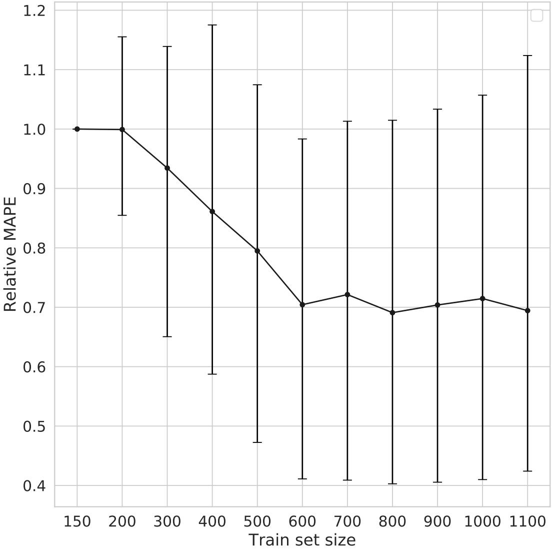


Figure 11: Relative test errors of the MAP-NN model for increasing training set sizes. Shown are the medians and 50% intervals of 400 trials.

7 Discussion

In Section 1 some of the challenges faced by data-driven VFMs were discussed. These were: (1) low data volume, (2) low data variety (3) poor measurement quality, and (4) non-stationarity of the underlying process. Here we discuss the results in light of these challenges. All results are discussed in terms of MAPE values.

No widely used standard exists for VFM performance specification or requirements. Thus, the following performance requirements have been set by the authors to assess the commercial viability of a VFM: 1) predictive performance in terms of mean absolute percentage error on test data of 10% or less, and 2) robustness in terms of achieving the above predictive performance for at least 90% of wells. While these simple requirements lack a specification of the test data, we find them useful in the assessment of VFM performance. A VFM failing to meet these requirements would not be practical to use in industrial applications.

7.1 Performance on historical and future test data

First, we discuss the concern about the non-stationarity of the underlying process. This means the distribution of values seen during training is not necessarily the same as the distribution of values used for testing. The effect of this is best observed when comparing the performance on historical and future data, see Table 1 and 2 and Figure 8. Looking at the upper and lower percentiles, we see the different models achieve performance in the range of 1-16% error on historical data and 3-40% error on future data. Since the strength of data-driven models lies with interpolation, rather than extrapolation, it is natural that the performance is worse on the future data case. Considering the VFM performance requirement of 10% MAPE for 90% of the wells, the performance is not acceptable for the historical or for the future data case. This indicates that the robustness of the models is inadequate for use in a commercial VFM. For real time applications, frequent model updates are likely required to achieve the VFM performance requirement. This raises the technical challenge of implementing a data-driven modeling approach.

The study on dataset size in Section 6.5 further explores the development of data distributions and the effect older data has on future prediction errors. The result, seen in Figure 11, indicates that additional data is only valuable up to a certain point, after which older data will no longer be useful when predicting future values. The point where this happens will naturally vary between wells. For the wells included here, this happens at 600 data points on average, for which the additional data is approximately 18 months or longer into the past. Looking at Figure 10, we again see the trend that wells with more data perform better, but only up to a certain point. We remark that insufficient model capacity would have a similar effect on the performance. However, we find this to be unlikely in this case study due to the high capacity and low training errors of the neural networks used.

At this point we remark that, for two observations $D_1, D_2 \in \mathcal{D}$, we model conditional independence ($D_1 \perp\!\!\!\perp D_2 \mid \boldsymbol{\theta}$). While the observations result from preprocessing measurement data in a way that removes transients and decorrelates observations, we cannot guarantee independence due to the non-stationary process. With dependant observations, the modeling assumption of conditional independence is not satisfied since the models lack temporal dependencies. This is also true for most, if not all published models for data-driven VFM. Models that include temporal dependencies may be better suited to learn from past data.

A second concern raised was related to small data regimes, both in terms of data volume and data variety. The results mentioned above also illustrate the effect of small data. Looking at Figure 10, higher variance in performance is seen among wells with less than 700 data points. This is concerning because many of the wells, in particular those with test separator measurements as their primary source of data, have very few data points. Based on the median MAPE values in Figure 8, also given in Table 4 and 5, models trained on MPFM data outperforms the models trained on test separator data. This indicates that data quantity may outweigh data quality in the small-data regime. The difference in performance is also evident in the cumulative performance plots, see Figure 6 and 7.

The wells that lie in the top quarter of performance achieved MAPE values comparable to the earlier works discussed in Section 2.1. However, this performance seems difficult to achieve for the full set of wells. The difficulty in generalizing a single model architecture to a broad set of wells is troublesome for the potential commercialization of data-driven VFM.

7.2 Noise models

The last concern raised was poor data quality. In particular uncertainty in flow rate measurements, and potential gross errors in MPFM measurements.

The three different noise models perform similarly in terms of MAPE, on both historical and future data. The only exception being the learned heteroscedastic noise model, which performed better than the others on historical and future test data case when judged by the 90th percentile. This is believed to be because the heteroscedastic error term gives the objective function some added robustness towards large errors.

From the calibration plots in Figure 9, we see that learning the noise model improves the calibration. The calibration curves for models trained on MPFM data generally lie above the curves for models trained on test separator data, both for historical and future predictions. This means that models trained on MPFM measurements are less confident in their predictions, even though they are trained on more data. It was suspected that models trained on MPFM data would reflect the increased uncertainty present in these measurements, but this is difficult to observe from the results. It is worth noting that the MPFM models are tested on MPFM data, so any systematic errors present in the MPFM measurements themselves will not be detected.

Because the models have potentially large prediction errors, especially for future data, it is desirable that the model can assess its performance. The coverage probabilities reported in Table 3 give us some confidence in the uncertainty estimates for the learned noise models, especially for the historical cases.

Neither the homoscedastic or heteroscedastic noise models in (3) and (4), respectively, can capture complex noise profiles that depend on the flow conditions $\boldsymbol{x}$. As most flow meters are specialized to accurately measure flow rates for certain compositions and flow regimes, this is a potential drawback of the models. We leave it to later works to address such limitations, but note that with few adjustments the flow model in (1) can accommodate heteroscedasticity of a rather general form.

7.3 Bayesian neural networks

As stated in Section 1, setting the priors on the parameters in the model is not a trivial task. In several papers, the Kullback-Leibler divergence term of the ELBO loss in (18) is down-weighted to improve model performance due to poor priors (Wenzel et al., 2020). This remains a research question, however, in Section 4.2 one way of approaching prior specification in BNNs is described. The difficulty of setting priors combined with small data sets may make it difficult to successfully train models of this complexity. Still, the results are reasonable in the historical data case, and the estimated uncertainty is still better than only relying on point estimates.

8 Concluding remarks

MAP estimation and VI for a probabilistic, data-driven VFM was presented and explored in a case study with 60 wells. The models achieve acceptable performance on future test data for approximately half of the studied wells. It is observed that models trained on historical data lack robustness in a changing environment. Frequent model updates are therefore likely required, which pose a technical challenge in terms of VFM maintenance.

Of the presented data challenges, the non-stationary data distribution is the most concerning. It means that models must have decent extrapolating properties if they are to be used in real-time applications. This is inherently challenging for data-driven approaches, and limits the performance of all the models considered in this paper. Of the models explored here, VI provided more robust predictions than MAP estimation on future test data.

The BNN approach is promising due to its ability to provide uncertainty estimates. Among these models, the heteroscedastic model had the best performance, indicating that a heteroscedastic model can be advantageous for flow rate measurements. However, it is challenging to obtain well-calibrated models due to the difficulty of setting meaningful priors on neural network weights, and the fact that priors play a significant role in small data regimes. As a result, the uncertainty estimates provided by the BNNs should be used with caution.

8.1 Recommendations for future research

We would suggest future research on data-driven VFM to focus on ways to overcome the challenges related to small data and non-stationary data distributions. Advances on these problems are likely required to improve the robustness and extrapolation capabilities of models to be used in real-time applications. We believe promising avenues of research to be: i) hybrid data-driven, physics-based models that allows for stronger priors; ii) data-driven architectures that enables learning from more data, for instance by sharing parameters between well models; iii) online learning to enable frequent model updates; and iv) modeling of temporal dependencies, for example using sequence models, to better capture time-varying boundary conditions.

Acknowledgements

This work was supported by Solution Seeker AS.

A Derivations

A.1 Log-likelihood of the flow rate model

The log-likelihood of the flow model in (1) with parameters $\boldsymbol{\theta} = (\boldsymbol{\phi}, \boldsymbol{\psi})$ on a dataset $\mathcal{D} = (X, \boldsymbol{y}) = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ is given by

$$\begin{aligned}
\log p(\boldsymbol{y} \mid X, \boldsymbol{\theta}) &= \sum_{i=1}^{N} \log p(y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}) \\
&= \sum_{i=1}^{N} \log \mathcal{N}(y_i \mid f(\boldsymbol{x}_i, \boldsymbol{\phi}), g(f(\boldsymbol{x}_i, \boldsymbol{\phi}), \boldsymbol{\psi})^2) \\
&= -\frac{N}{2} \log(2\pi) - \sum_{i=1}^{N} \log g(f(\boldsymbol{x}_i, \boldsymbol{\phi}), \boldsymbol{\psi}) - \frac{1}{2} \left(\frac{y_i - f(\boldsymbol{x}_i, \boldsymbol{\phi})}{g(f(\boldsymbol{x}_i, \boldsymbol{\phi}), \boldsymbol{\psi})} \right)^2 .
\end{aligned} \tag{22}$$

With a homoscedastic noise model $g(z, \boldsymbol{\psi}) = \sigma_n = \text{const.}$, the log-likelihood simplifies to:

$$\log p(\boldsymbol{y} \mid X, \boldsymbol{\theta}) = -\frac{N}{2} \log(2\pi\sigma_n^2) - \frac{1}{2\sigma_n^2} \sum_{i=1}^{N} (y_i - f(\boldsymbol{x}_i, \boldsymbol{\phi}))^2 . \tag{23}$$

A.2 Kullback-Leibler divergence term, $D_{\mathbf{KL}}\left(q(\boldsymbol{\theta} \mid \boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta})\right)$

Let the approximation $q(\boldsymbol{\theta} \mid \boldsymbol{\lambda})$ and prior $p(\boldsymbol{\theta})$ be mean-field normal distributions of the random variables $\boldsymbol{\theta} \in \mathbb{R}^K$. Assume that the approximation is parameterized with $\boldsymbol{\lambda} = (\boldsymbol{\mu}, \boldsymbol{\rho})$, where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\sigma} = \log(1 + \exp(\boldsymbol{\rho}))$ is the standard deviation of q. Then, the Kullback-Leibler divergence is given as:

$$\begin{aligned}
&D_{\mathrm{KL}}\left(q(\boldsymbol{\theta}\,|\,\boldsymbol{\lambda})\,\|\,p(\boldsymbol{\theta})\right) = \mathbf{E}_q\left[\log q(\boldsymbol{\theta}\,|\,\boldsymbol{\lambda}) - \log p(\boldsymbol{\theta})\right] \\
&= \mathbf{E}_q\left[\sum_{i=1}^{K} \log q(\theta_i\,|\,\lambda_i) - \log p(\theta_i)\right] \\
&= \frac{1}{2}\mathbf{E}_q\left[\sum_{i=1}^{K} - \log(2\pi\sigma_i^2) - \left(\frac{\theta_i - \mu_i}{\sigma_i}\right)^2 + \log(2\pi\bar{\sigma}_i^2) + \left(\frac{\theta_i - \bar{\mu}_i}{\bar{\sigma}_i}\right)^2\right] \\
&= \frac{1}{2}\left[\sum_{i=1}^{K} -2\log\frac{\sigma_i}{\bar{\sigma}_i} - \frac{1}{\sigma_i^2}\underbrace{\mathbf{E}_{q_i}\left[(\theta_i - \mu_i)^2\right]}_{=\sigma_i^2} + \frac{1}{\bar{\sigma}_i^2}\mathbf{E}_{q_i}\left[(\theta_i - \bar{\mu}_i)^2\right]\right] \qquad (24) \\
&= \frac{1}{2}\sum_{i=1}^{K}\left[-1 - 2\log\frac{\sigma_i}{\bar{\sigma}_i} + \frac{1}{\bar{\sigma}_i^2}\mathbf{E}_{q_i}\left[(\theta_i - \bar{\mu}_i)^2\right]\right] \\
&= \frac{1}{2}\sum_{i=1}^{K}\left[-1 - 2\log\frac{\sigma_i}{\bar{\sigma}_i} + \left(\frac{\mu_i - \bar{\mu}_i}{\bar{\sigma}_i}\right)^2 + \left(\frac{\sigma_i}{\bar{\sigma}_i}\right)^2\right]
\end{aligned}$$

B Results

Table 4: Prediction performance on historical test data for each well group. Reported values are the P_{10}, P_{25}, P_{50}, P_{75}, and P_{90} percentiles for the statistics root mean square error (RMSE) and mean absolute percentage error (MAPE).

Well group	Method and model	RMSE	MAPE %
All	MAP-NN fixed homosc.	0.4, 0.7, 1.1, 1.7, 3.0	1.8, 2.8, 5.1, 8.3, 16.0
	VI-NN fixed homosc.	0.3, 0.5, 1.0, 2.1, 3.0	1.4, 2.6, 4.8, 8.5, 12.8
	VI-NN learned homosc.	0.3, 0.5, 1.0, 2.0, 3.0	1.3, 2.4, 5.3, 8.4, 13.3
	VI-NN learned heterosc.	0.4, 0.6, 1.2, 1.9, 3.0	1.7, 3.5, 5.9, 9.7, 11.5
Test sep.	MAP-NN fixed homosc.	0.4, 0.8, 1.5, 1.7, 3.0	3.1, 5.7, 7.2, 11.1, 16.2
	VI-NN fixed homosc.	0.5, 0.8, 1.6, 2.2, 4.3	2.8, 4.9, 7.9, 11.3, 13.2
	VI-NN learned homosc.	0.6, 1.1, 1.7, 2.1, 3.1	3.9, 5.8, 8.1, 12.3, 16.4
	VI-NN learned heterosc.	0.5, 1.0, 1.7, 2.1, 3.9	3.7, 5.1, 9.5, 11.4, 12.2
MPFM	MAP-NN fixed homosc.	0.3, 0.6, 1.0, 1.6, 2.8	1.8, 2.4, 4.5, 8.1, 14.3
	VI-NN fixed homosc.	0.3, 0.4, 1.0, 1.9, 2.9	1.3, 2.3, 4.1, 7.7, 11.5
	VI-NN learned homosc.	0.3, 0.4, 0.7, 1.6, 3.0	1.2, 2.0, 4.1, 7.3, 11.7
	VI-NN learned heterosc.	0.4, 0.5, 1.2, 1.5, 2.9	1.3, 3.1, 5.1, 8.6, 10.8

Table 5: Prediction performance on future test data for each well group. Reported values are the P_{10}, P_{25}, P_{50}, P_{75}, and P_{90} percentiles for the statistics root mean square error (RMSE) and mean absolute percentage error (MAPE).

Well group	Method and model	RMSE	MAPE %
All	MAP-NN fixed homosc.	0.8, 1.2, 2.1, 4.0, 6.1	3.7, 5.6, 12.4, 24.1, 40.0
	VI-NN fixed homosc.	0.6, 1.1, 1.8, 3.5, 5.2	4.0, 5.6, 9.6, 18.2, 29.3
	VI-NN learned homosc.	0.7, 1.2, 1.9, 3.3, 5.5	4.0, 6.0, 8.9, 22.5, 32.5
	VI-NN learned heterosc.	0.6, 1.1, 1.7, 3.1, 4.5	4.0, 5.0, 9.2, 15.7, 24.3
Test sep.	MAP-NN fixed homosc.	0.8, 1.0, 1.6, 3.0, 6.7	3.9, 6.2, 18.1, 28.8, 41.1
	VI-NN fixed homosc.	0.3, 1.0, 2.1, 3.2, 8.0	5.2, 9.5, 14.6, 31.4, 40.9
	VI-NN learned homosc.	0.6, 1.3, 1.9, 3.6, 5.9	6.6, 7.8, 15.5, 31.6, 35.9
	VI-NN learned heterosc.	0.4, 1.2, 1.6, 2.3, 2.9	5.1, 6.0, 10.6, 18.6, 21.6
MPFM	MAP-NN fixed homosc.	0.9, 1.2, 2.4, 4.2, 5.7	3.7, 6.2, 12.2, 23.0, 30.2
	VI-NN fixed homosc.	0.8, 1.3, 1.8, 3.5, 4.6	4.0, 5.3, 8.3, 15.0, 24.6
	VI-NN learned homosc.	0.7, 1.1, 1.9, 3.1, 5.2	3.4, 4.9, 8.0, 17.5, 28.1
	VI-NN learned heterosc.	0.7, 1.0, 1.8, 3.3, 4.6	3.8, 4.7, 8.9, 14.9, 24.5

References

Ahmadi, M. A. et al. (2013). “Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir”. In: *Applied Soft Computing* 13 (2), pp. 1085–1098.

Amin, A. (2015). “Evaluation of Commercially Available Virtual Flow Meters (VFMs)”. In: *Proceedings of the Annual Offshore Technology Conference*, pp. 1293–1318.

Balaji, K. et al. (2018). “Status of data-driven methods and their application in oil and gas industry”. In: *SPE Europec featured at 80th EAGE Conference and Exhibition.*

Bassamzadeh, Nastaran and Roger Ghanem (2018). “Probabilistic data-driven prediction of wellbore signatures in high-dimensional data using Bayesian networks”. In: *Society of Petroleum Engineers.*

Bello, O., S. Ade-Jacob, and K. Yuan (2014). “Development of Hybrid Intelligent System for Virtual Flow Metering in Production Wells.” In: *Society of Petroleum Engineers, Intelligent Energy Conference and Exhibition.*

Bengio, Y. (2012). “Practical Recommendations for Gradient-Based Training of Deep Architectures.” In: *arXiv:1206.5533v2*, pp. 1–33.

Berneti, S. M. and M. Shahbazian (2011). “An Imperialist Competitive Algorithm - Artificial Neural Network Method to Predict Oil Flow Rate of the Wells”. In: *International Journal of Computer Applications* 26 (10), pp. 47–50.

Bieker, Hans Petter, Olav Slupphaug, and Tor Arne Johansen (2007). “Well management under uncertain gas or water oil ratios”. In: *SPE Digital Energy Conference and Exhibition.* Society of Petroleum Engineers.

Bikmukhametov, T. and J. Jäschke (2019). “Oil Production Monitoring using Gradient Boosting Machine Learning Algorithm.” In: *IFAC-PapersOnLine* 52 (1), pp. 514–519.

Bikmukhametov, Timur and Johannes Jäschke (2020). “Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models”. In: *Computers and Chemical Engineering* 138.

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe (2017). “Variational Inference: A Review for Statisticians”. In: *J. Am. Stat. Assoc.* 112.518, pp. 859–877.

Blundell, Charles et al. (2015). “Weight Uncertainty in Neural Networks”. In: *32nd International Conference on Machine Learning.* Vol. 37, pp. 1613–1622. eprint: `1505.05424`.

Câmara, Mauricio Melo et al. (2017). “Numerical Aspects of Data Reconciliation in Industrial Applications”. In: *Processes* 5 (4).

Corneliussen, Sidsel et al. (2005). *Handbook of multiphase flow metering.* The Norwegian Society for Oil and Gas Measurements.

Fonseca, Junior R.D., M.D.A.L. Gonçalves, and L.F.A. Azevedo (2009). “Consideration of uncertainty in simulation and flow. In Portugese: Consideração de incerteza nas simulações de elevação e Escoamento”. In: *An. do IV Semin. Elev. Artif. e Escoamento.*

Foss, B., B. R. Knudsen, and B. Grimstad (2018). “Petroleum production optimization - A static or dynamic problem”. In: *Computers and Chemical Engineering* 114, pp. 245–253.

Gal, Yarin (2016). “Uncertainty in Deep Learning”. PhD Thesis. University of Cambridge, p. 174.

Ghahramani, Zoubin (2015). “Probabilistic machine learning and artificial intelligence”. In: *Nature* 521 (7553), pp. 452–459.

Glorot, Xavier and Yoshua Bengio (2010). “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the 13th international conference on artificial intelligence and statistics.* Vol. 9, pp. 249–256.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). “Deep Sparse Rectifier Neural Networks”. In: *14th International Conference on Artificial Intelligence and Statistics.* Vol. 15, pp. 315–323.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning.* Cambridge, Massachusetts. London, England.: The MIT Press.

Grimstad, B. et al. (Sept. 2016). “A Simple Data-Driven Approach to Production Estimation and Optimization”. In: *In: SPE Intelligent Energy International Conference and Exhibition.*

Hasanvand, Mahdi Zeinali and S.M. Berneti (2015). “Predicting Oil Flow Rate due to Multiphase Flow Meter by Using an Artificial Neural Network”. In: *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* 37 (8), pp. 840–845.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning.* New York, USA: Springer.

He, Kaiming et al. (2015). “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

Hotvedt, M., B. Grimstad, and L. Imsland (2020). "Developing a Hybrid Data-Driven, Mechanistic Virtual Flow Meter - a Case Study". In: *IFAC-PapersOnLine* 53 (2), pp. 11692–11697.

Hüllermeier, Eyke and Willem Waegeman (Mar. 2021). "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods". In: *Machine Learning* 110.3, pp. 457–506.

Humphrey, Greer B. et al. (2016). "A hybrid approach to monthly streamflow forecasting: Integrating hydrological model outputs into a Bayesian artificial neural network". In: *Journal of Hydrology* 540, pp. 623–640.

Jansen, Jan-Dirk (2015). *Nodal Analysis of Oil and Gas Wells - Theory and Numerical Implementation.* TU Delft, The Netherlands: Delft University of Technology.

Kingma, Diederik P and Max Welling (2014). "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations (ICLR).* eprint: 1312.6114.

Kingma, Diederik P. and Jimmy Lei Ba (2015). "Adam: A method for stochastic optimization". In: *3rd International Conference on Learning Representations (ICLR)*, pp. 1–15. eprint: 1412.6980.

Kingma, Diederik P., Tim Salimans, and Max Welling (2015). "Variational dropout and the local reparameterization trick". In: *28th International Conference on Neural Information Processing Systems.* Vol. 2, pp. 2575–2583.

Krejbjerg, Kristian et al. (2019). *Conversion of multiphase meter flowrates.* The Norwegian Society for Oil and Gas Measurements (NFOGM).

Liu, Ying et al. (2012). "Data-driven based model for flow prediction of steam system in steel industry". In: *Information Sciences* 193, pp. 104–114.

Lorentzen, Rolf J., Andreas S. Stordal, Xiaodong Luo, et al. (2016). "Estimation of Production Rates by Use of Transient Well-Flow Modeling and the Auxiliary Particle Filter: Full-Scale Applications". In: *SPE Production and Operations* 31(2), pp. 163–175.

Lorentzen, Rolf J., Andreas S. Stordal, Geir Nævdal, et al. (2014). "Estimation of Production Rates With Transient Well-Flow Modeling and the Auxiliary Particle Filter". In: *Society of Petroleum Engineers* 19(1), pp. 172–180.

Luo, Xiaodong et al. (2014). "Toward an enhanced Bayesian estimation framework for multiphase flow soft-sensing". In: *Inverse Problems* 30.

Marshall, C. and A. Thomas (2015). "Maximising Economic Recovery - a review of well test procedures in the North Sea". In: *Conference: SPE Offshore Europe Conference and Exhibition.*

Mishra, Srikanta and Akhil Datta-Gupta (2018). *Applied Statistical Modeling and Data Analytics - A Practical Guide for the Petroleum Geosciences.* Elsevier.

Monteiro, Danielle D., Gabriela S. Chaves, et al. (2017). "Uncertainty analysis for production forecast in oil wells". In: *SPE Latin American And Caribbean Petroleum Engineering Conference.*

Monteiro, Danielle D., Maria Machado Duque, et al. (2020). "Using Data analytics to quantify the impact of production test uncertainty on oil flow rate forecast". In: *IFP Energies Nouvelles* 75 (7), pp. 1–15.

Ovadia, Yaniv et al. (2019). “Can You Trust Your Model’s Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift”. In: *33rd Conference on Neural Information Processing Systems*. arXiv: 1906.02530.

Paszke, Adam et al. (2019). “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems* 32, pp. 8026–8037. eprint: 1912.01703.

Polson, Nicholas G. and Vadim Sokolov (2017). “Deep learning: A Bayesian perspective”. In: *Bayesian Analysis* 12.4, pp. 1275–1304.

AL-Qutami, T.A et al. (2018). “Virtual multiphase flow metering using diverse neural network ensemble and adaptive simulated annealing”. In: *Expert Syst. Appl.* 93, pp. 72–85.

AL-Qutami, Tareq Aziz, Rosdiazli Ibrahim, and Idris Ismail (2017). “Hybrid neural network and regression tree ensemble pruned by simulated annealing for virtual flow metering application.” In: *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 304–309.

AL-Qutami, Tareq Aziz, Rosdiazli Ibrahim, Idris Ismail, and Mohd Azmin Ishak (2017a). “Development of soft sensor to estimate multiphase flow rates using neural networks and early stopping”. In: *Int. J. Smart Sens. Intell. Syst.* Vol. 10, pp. 199–222.

— (2017b). “Radial basis function network to predict gas flow rate in multiphase flow.” In: *Proceedings of the 9th International Conference on Machine Learning and Computing*, pp. 141–146.

Snoek, J., H. Larochelle, and R.P. Adams (2012). “Practical Bayesian Optimization of Machine Learning algorithms”. In: *arXiv:1206.2944v2*.

Solle, D. et al. (2016). “Between the Poles of Data-Driven and Mechanistic Modeling for Process Operation”. In: *Chemie Ingenieur Technik*.

Toskey, E.D. (2012). “Improvements to Deepwater Subsea Measurements RPSEA Program: Evaluation of Flow Modeling”. In: *Offshore Technology Conference*.

Wenzel, Florian et al. (2020). “How Good is the Bayes Posterior in Deep Neural Networks Really?” In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119, pp. 10248–10259. eprint: 2002.02405.

Woodward, Andrew M., Bjørn K. Alsberg, and Douglas B. Kell (1998). “The effect of heteroscedastic noise on the chemometric modelling of frequency domain data”. In: *Chemometrics and Intelligent Laboratory Systems* 40, pp. 101–107.

Xu, L. et al. (2011). “Wet gas metering using a revised venturi meter and soft-computing approximation techniques”. In: *IEE transactions on instrumentation and measurement* 60 (3), pp. 947–956.

Zangl, G., R. Hermann, and C. Schweiger (2014). “Comparison of Methods for Stochastic Multiphase Flow Rate Estimation”. In: *Society of Petroleum Engineers*.

D On gray-box modeling for virtual flow metering

Postprint of Hotvedt, Grimstad, Ljungquist, et al. (2022a)

M. Hotvedt, B. Grimstad, D. Ljungquist, and L. Imsland (2022a). "On gray-box modeling for virtual flow metering". In: *Control Engineering Practice* 118 (1)

On gray-box modeling for virtual flow metering

M. Hotvedt[1], B. Grimstad[1, 2], L. Imsland[1], and D. Ljungquist[3]

[1]Engineering Cybernetics Department, NTNU, Trondheim, Norway
[2]Solution Seeker
[3]TechnipFMC

Abstract: A virtual flow meter (VFM) enables continuous prediction of flow rates in petroleum production systems. The predicted flow rates may aid the daily control and optimization of a petroleum asset. Gray-box modeling is an approach that combines mechanistic and data-driven modeling. The objective is to create a computationally feasible VFM for use in real-time applications, with high prediction accuracy and scientifically consistent behavior. This article investigates five different gray-box model types in an industrial case study using real, historical production data from 10 petroleum wells, spanning at most four years of production. The results are diverse with an oil flow rate prediction error in the range of 1.8%-40.6%. Further, the study casts light upon the nontrivial task of balancing learning from both physics and data. Therefore, providing general recommendations towards the suitability of different hybrid models is challenging. Nevertheless, the results are promising and indicate that gray-box VFMs can reduce the prediction error of a mechanistic VFM while remaining scientifically consistent. The findings motivate further experimentation with gray-box VFM models and suggest several future research directions to improve upon the performance and scientific consistency.

Keywords: Gray-box, virtual flow meter, multiphase flow, neural network

1 Introduction

To optimally control a petroleum asset and maximize the recovery of oil and gas, it is necessary to have an adequate understanding of the behavior of the petroleum production system. This consists of the reservoir, wells, flowlines, pipelines, and separators. Commonly, a mathematical model of the flow through the production system is developed as an aid to information gathering and analysis of the system response to changes in control variables. Such a model is often referred to as a virtual flow meter (VFM) (Toskey, 2012). A VFM aims to continuously predict the multiphase flow rates (mixture of gas, oil, and water) at strategic locations in the asset, for instance in individual wells. The characteristics of multiphase flow represents a particular challenge to prediction. Several types of VFM models exist, ranging from mechanistic to data-driven, thus, from white-box to black-box, respectively (Prada et al., 2018). Depending on the prior knowledge about the system and the available process data, one model type can be more suitable than others, see Figure 1.

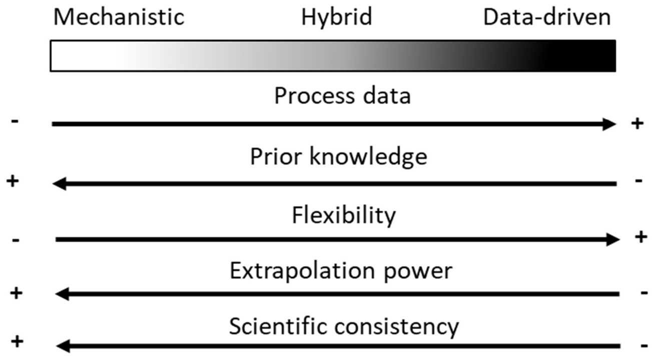


Figure 1: The range of model types from mechanistic, white-box models to data-driven, black-box models and a few of their characteristics.

1.1 Virtual flow meter models

Mechanistic models are based on prior knowledge about the process and utilize first-principle laws, with possible empirical closure relations, to describe the relationship between the process input, internal, and output variables (Shippen, 2012). Contrarily, data-driven models require no prior knowledge of the process, and rather exploit patterns in available process data to describe the input-output relationship. Therefore, data-driven models often lack scientific consistency. A model may be considered scientifically consistent if the output of the model is plausible and in line with existing scientific principles (Roscher et al., 2020). Although this concept is hard to quantify and dependent on the user's scientific knowledge, it is an important characteristic as it promotes trust in the model. As mechanistic models are derived from physical laws, their scientific consistency is high. On the other hand, assumptions and simplifications of the process physics are typically necessary for a mechanistic model to be computationally feasible and suitable for use in real-time control and optimization applications (Solle et al., 2016). Accordingly, mechanistic models often lack flexibility, which is the ability to adapt to unknown and unmodeled physical phenomena. Oppositely, due to the generic structure of data-driven models, the flexibility is high and the models may adapt to arbitrary complex physical behavior as long as this is reflected in the available data. Yet, data-driven models are data-hungry and sensitive to the quality and variability of the data. If care is not taken, overfitting of the model to data is a frequent outcome that results in poor extrapolation abilities to future process conditions (Solle et al., 2016).

Gray-box models, or hybrid models, are a combination of mechanistic and data-driven models. The goal is to achieve a computationally feasible model that have a high flexibility and a scientifically consistent behavior. There exist numerous ways of constructing hybrid models. According to Willard et al. (2020), gray-box models can be divided into two domains: 1) data-driven modeling to advance first-principle models, or 2) utilization of first principles to guide data-driven models. The two domains correspond to either side of the gray-scale illustrated in Figure 1 and will be referred to as the white-to-gray and the black-to-gray approach. Taking VFM as an example, a white-to-gray model is obtained if a mechanistic model is used as a baseline whereupon data-driven models are inserted to replace assumptions or simplifications. For instance, a common approach to estimate the density of gas in a mechanistic model is with the real gas law. Instead, if this relation is described with a data-driven model, a white-to-gray VFM model is obtained. Another example is to introduce a data-driven model to capture the error between the output of the mechanistic model and corresponding measurements, see an example in Timur Bikmukhametov and Johannes Jäschke (2020b). In general, the data-driven models may substitute any factors or terms in the mechanistic model. An example of a black-to-gray VFM model is if first principles are exploited to calculate additional features to be applied as input to a data-driven model. This is commonly referred to as feature engineering. A different approach is a division into natural submodels, for instance individual wells in an asset, describe each with a data-driven model and combine the output using first-principle laws. The two approaches can also be juxtaposed. For instance, both a mechanistic and a data-driven model can be developed to predict the multiphase flow rate and the model outputs combined in an ensemble model. Independent of the gray-box model type, measures should be taken to determine an appropriate degree of influence the mechanistic and data-driven part should have on the model output. In other words, there should exist a pertinent balance between learning from physics and learning from data. For instance, if the available process data are inaccurate, the mechanistic part of the model should influence the gray-box model output the most. If the process exhibits unknown behavior, the data-driven part should have the greatest impact. Desirably, the gray-box model should learn as much as possible from both physics and data.

1.2 Literature review

The literature reports substantial research on mechanistic and data-driven modeling of VFMs (Amin, 2015; Zangl, Hermann, and Schweiger, 2014; AlAjmi, Alarifi, and Mahsoon, 2015; T. A. AL-Qutami, Ibrahim, and Ismail, 2017; T. A. AL-Qutami, Ibrahim, Ismail, and Ishak, 2017a; T. A. AL-Qutami, Ibrahim, Ismail, and Ishak, 2017b; T. AL-Qutami et al., 2018; Omrani et al., 2018; T. Bikmukhametov and J. Jäschke, 2019; Ghorbani et al., 2018). An extensive review is found in Timur Bikmukhametov and Johannes Jäschke (2020a). Some well-known commercial mechanistic VFMs are Olga, LedaFlow, FlowMananger, ValiPerformance, and Prosper. In the study by Amin (2015), it was found that all the above-mentioned commercial mechanistic VFM achieved an error less than 5% and 10% for the prediction of oil and gas flow rates, respectively. The noticeable series of studies on data-driven VFM by T. A. AL-Qutami, Ibrahim, and Ismail (2017), T. A. AL-Qutami, Ibrahim, Ismail, and Ishak (2017a), T. A. AL-Qutami, Ibrahim, Ismail, and Ishak (2017b), and T. AL-Qutami et al. (2018) achieved errors of 1.5%, 4.2%, and 4.7% on the predictions of gas, oil, and water flow rates, respectively.

Despite recent emerging tools for hybrid, gray-box modeling, such as gPROMS (Siemens Process Systems Engineering, 2021), and even a commercially available hybrid VFM: TurbulentFlux (Ruden, 2020), little literature on the performance of gray-box VFMs exist. TurbulentFlux reports an error of 4% on multiphase flow rate predictions over two months for one of the tested wells. However, the robustness in performance for different wells is not reported. Furthermore, as no reference model is tested on the available data it is difficult to conclude whether the hybrid model performs better than alternative approaches. Nevertheless, some examples exist in the literature (Xu et al., 2011; Al-Rawahi et al., 2012; Kanin et al., 2019; Timur Bikmukhametov and Johannes Jäschke, 2020b). Most of these studied different gray-box approaches on synthetic data, either as an experimental set up in a test rig (Xu et al., 2011) or a multiphase flow loop (Al-Rawahi et al., 2012), or using lab data available online (Kanin et al., 2019). The study in Timur Bikmukhametov and Johannes Jäschke (2020b) investigated several hybrid VFM variants on real production data, with a large focus on the black-to-gray modeling approach. However, their results were based on process data from only one subsea well and the modeling approach could benefit from a deeper study of more petroleum wells.

1.3 Contributions

This research contributes to the field of gray-box VFM modeling with an in-depth study of five white-to-gray VFM models of a petroleum production choke valve. A mechanistic and data-driven model is developed for comparison of the performance and scientific consistency. The study is a significant expansion of the work done in Hotvedt, Grimstad, and Imsland (2020) and Hotvedt, Grimstad, and Imsland (2021). The number of tested gray-box models is increased, the complexity of the model components is higher, and data from more wells are included. The VFM models are developed for 10 petroleum wells at Edvard Grieg (Lundin Energy Norway, 2020). Real, historical production data are used in the model development, thus no experimental setup or simulator is required for data acquisition. With data from 10 wells, the robustness of the modeling approaches can be investigated to a certain extent. The results in this research are in respect to the VFM application, and the generalizability to other application areas is not considered.

2 Production choke valve models

A production system is illustrated in Figure 2, from the down-hole, the closest measurement point to the reservoir, to the separator. The volumetric flow rate from several wells are commingled and the total production from the asset is separated into three phases, oil (Q_O), water (Q_W), and gas (Q_G), at the separator. The production choke valve is located in the wellhead of the production system. The choke is a key element in the daily control and optimization of a petroleum production system. By adjusting the choke opening, the multiphase flow rate through the production system can be controlled to maximize production while meeting operational requirements such as production capacity constraints. In this research, only the production choke is modeled. This results in lesser model complexity and avoids the utilization of down-hole sensor measurements. For assets where down-hole measurements are lacking or faulty, this is advantageous. Naturally, for assets with good down-hole measurements, the VFM can be expanded.

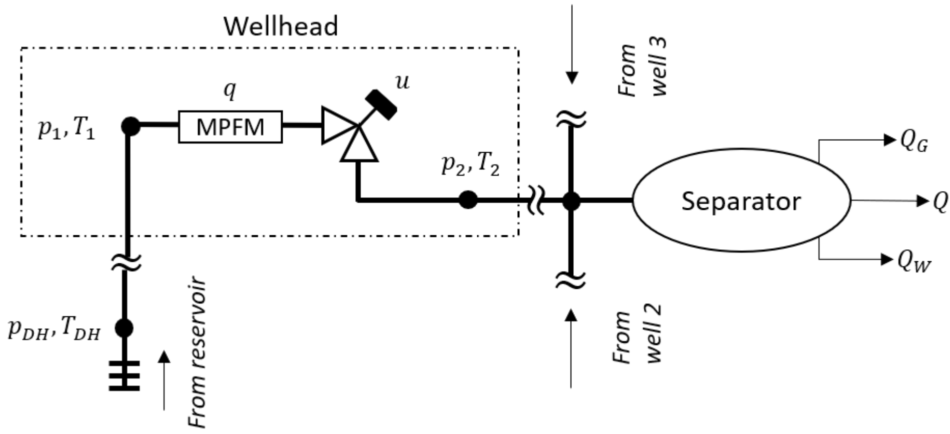


Figure 2: Illustration of the production system, from the down-hole (DH) to the separator. The production choke valve is located in the wellhead. Typically available measurements are indicated.

To model the choke for individual wells, the following measurements are required: the choke opening (u), the pressures (p) and temperatures (T) located upstream (1) and downstream (2) the choke valve, and measurements of the flow rate (q). Measurements of the phasic flow rates in individual wells $\boldsymbol{q} = (q_O, q_G, q_W)$ can be obtained from well tests, for instance using a test separator, or multiphase flow meters (MPFMs) if these are available. Furthermore, the phasic fluid mass fractions are required. Ideally, these should be calculated with a different model for each new sample, for example using a simplified wellbore model as in (Kittilsen, Fjalestad, and Aasheim, 2014). Nevertheless, in this research, the mass fractions are treated as measurements, calculated using the flow rates from the MPFM in the previous measurement sample. Consequently, the utilized mass fraction will lag behind the true mass fractions. However, under the assumption of a slowly time-varying process, the mass fractions should not change significantly between each sample.

2.1 Mechanistic production choke model

Several mechanistic models exist for the production choke, in a varying scale of complexity in space and time. Mechanistic choke models are usually developed assuming steady-state, one dimensional (lumped) flow since increasing the dimensionality of the problem requires a numerical solution of the complex Navier-Stokes equations. These equations are computationally demanding and may not be suitable for use in real-time optimization (Shippen, 2012). There are several well-known choke models in literature and industry (Selmer-Olsen, 1995; Sachdeva et al., 1986; Perkins, 1993; Al-Safran and Kelkar, 2009). In this research, the Sachdeva model is used as the baseline model for hybridization. The Sachdeva model is one of the less complex models as it introduces many assumptions and simplifications. Expectantly, introducing data-driven elements into the mechanistic model should increase the flexibility of the model and possibly supersede some of the simplifications. The exception is distributed effects in space and time as the Sachdeva model is assumed lumped and steady-state.

The Sachdeva model is derived from the combined mass and momentum balance equations (Jansen, 2015, p. 107):

$$\frac{dp}{ds} + \rho v \frac{dv}{ds} = 0, \tag{1}$$

$$\dot{m} = A_1 v_1 \rho_1 = A_2 v_2 \rho_2, \tag{2}$$

in which s is the position along a streamline, ρ is the fluid mixture density, v is the fluid mixture velocity, $\dot{m}$ is the mass flow rate, and A is the area of the choke valve. Positions (1) and (2) indicate the inlet and outlet, respectively. By integrating Equation (1) between location (1) and (2) and introducing several assumptions, for example:

- no-slip: the gas and liquid travels through the choke with equal velocity,
- incompressible liquid: liquid densities are constant along s resulting in the oil and water densities being independent of the process conditions,
- frozen flow: no mass transfers from one phase to another across the choke resulting in constant mass fractions independent of process conditions,
- adiabatic gas expansion across the choke: no mass or heat transfers between the fluid and the surroundings,

- thoroughly and homogeneously mixed fluid,
- neglect of momentum effects at (1) due to $A_2 \ll A_1$, yielding $v_2^2 \gg v_1^2$,

a model for the mass flow rate through the choke valve is obtained (see Sachdeva et al. (1986) for complete derivation):

$$\dot{m} = C_D A_2(u) \times \sqrt{2\rho_2^2 p_1 \left(\frac{\kappa}{\kappa - 1} \eta_G \left(\frac{1}{\rho_{G,1}} - \frac{p_r}{\rho_{G,2}} \right) + \left(\frac{\eta_O}{\rho_O} + \frac{\eta_W}{\rho_W} \right) (1 - p_r) \right)}, \tag{3}$$

$$\rho_{G,1} = \frac{p_1 M_G}{Z R T_1}, \tag{4}$$

$$\rho_{G,2} = \rho_{G,1} p_r^{\frac{1}{\kappa}}, \tag{5}$$

$$\frac{1}{\rho_2} = \frac{\eta_G}{\rho_{G,2}} + \frac{\eta_O}{\rho_O} + \frac{\eta_W}{\rho_W}, \tag{6}$$

$$\eta_G + \eta_O + \eta_W = 1. \tag{7}$$

Here $\rho_i, \eta_i, i \in \{G, O, W\}$ are the phasic densities and mass fractions, respectively, M_G is the molar mass of gas, and p_r is the downstream to upstream pressure ratio. The gas expansion coefficient κ is in this article treated as a constant but is in practice a function of pressure and temperature, $\kappa = \kappa(p_1, p_2, T_1, T_2)$. The gas compressibility factor Z is calculated using the correlation in (Sutton, 1985). The discharge coefficient C_D is commonly introduced to account for modeling errors. The area of the choke is a function of the choke opening $A_2 = A_2(u)$ since the choke is adjustable.

The model differentiates between critical and subcritical flow using

$$p_r = \begin{cases} \frac{p_2}{p_1} & \frac{p_2}{p_1} \geq p_{r,c} \\ p_{r,c} & otherwise \end{cases} \tag{8}$$

In short, critical flow is a phenomenon where the mass flow rate through the choke is not increasing for decreasing downstream pressure p_2 and fixed upstream pressure p_1. A rule of thumb for the critical flow boundary $p_{r,c}$ for multiphase flow with a mixture of gas, oil, and water is $p_{r,c} \approx 0.6$ (Jansen, 2015). The volumetric flow rate may be obtained using the mass flow rate and the mixture density in standard conditions (SC), typically 1 atm and $15°C$ (International Organization for Standardization, 1996). In this research, the model output is the oil volumetric flow rate:

$$q_O = \frac{\eta_O \dot{m}}{\rho_{O,SC}}, \tag{9}$$

Mathematically, the mechanistic model (MM) in (3)-(9) is described with the generic function f that predicts the oil volumetric flow rate for the input measurements $\boldsymbol{x}$ and the set of model parameters $\boldsymbol{\phi}_{MM}$:

$$\hat{y}_{MM} = q_{O,MM} = f(\boldsymbol{x}; \boldsymbol{\phi}_{MM}) \in \mathbb{R}, \tag{10}$$

$$\boldsymbol{x} = (p_1, p_2, T_1, T_2, u, \eta_G, \eta_O) \in \mathbb{R}^7, \tag{11}$$

$$\boldsymbol{\phi}_{MM} = (\rho_O, \rho_W, \kappa, M_G, p_{r,c}, C_D) \in \mathbb{R}^6. \tag{12}$$

The $\boldsymbol{\phi}_{MM}$ are components in the model which are considered constant due to certain assumptions or simplifications. For instance, as described above, the oil and water densities are constant parameters due to the assumption of incompressible liquid.

2.2 Hybridization of the mechanistic model

To hybridize the MM, any of the factors or terms in (3)-(9) can be substituted with a data-driven model (DM). Approaching the hybridization from a physical point of view, some of the mechanistic model assumptions or simplifications can be imprecise, yielding an erroneous physical behavior. For instance, in low temperature and high-pressure conditions, the real gas law relation in (4) may be inaccurate. Instead of using a different, and possibly more complex, mechanistic relation such as van der Waals equation of state, the hybrid model utilizes a DM to substitute the real gas law. Presumably, by learning the gas density relation from patterns in the measurements only, a relation that is suitable for the process and adaptable to the current conditions is obtained. Taking another example, the adiabatic gas expansion equation in (5) assumes that no heat or mass transfer occurs between the system and surroundings, yet, in practice, both exist. If the available measurements reflect these physical phenomena, a DM substituting (5) should, to some extent, be able to implicitly capture the effect of, for instance, heat transfer on the flow rate, even without measurements of the ambient temperature. Similarly, most of the assumptions listed above may be replaced with a data-driven model to account for erroneous physics. Consequently, the model should be more generic and suitable for utilization in a larger range of process conditions. Nevertheless, data-driven models are generally only valid in the domain of the data they have been exposed to. Hence, if the system is exposed to previously unseen process conditions, the hybrid models will likely have to be retrained or recalibrated to adapt to the new data.

There is an abundant number of hybridization options of the mechanistic model. Therefore, only a few of the simplifications and assumptions of the baseline model are investigated. Further, numerous combinations of these simplifications are viable, and for simplicity, only one simplification is considered at the time. Thereby, five hybrid model (HM) variants are developed, each addressing and substituting one of the following simplifications with a DM:

1. The area function, $A_2(u)$
2. The upstream gas density function, replacing (4)
3. The adiabatic gas expansion function, replacing (5).
4. The homogeneous mixture density function, replacing (6).
5. An additive error model to capture structural errors of the MM

Mathematically, the inserted DM is defined by

$$\hat{y}_{DM} = g(\boldsymbol{x}_{DM}; \boldsymbol{\phi}_{DM}) \in \mathbb{R}, \tag{13}$$

where $\boldsymbol{x}_{DM} \subseteq \boldsymbol{x}$ depends on the HM variant, and $\boldsymbol{\phi}_{DM}$ are a set of nonphysical parameters defining the structure of the DM. For the interested reader, if there exist measurements of what the DM represents, for example, density measurements, these may be incorporated into the model development by the means of prior parameter specification.

The HM is defined as a combination of the MM and DM by:

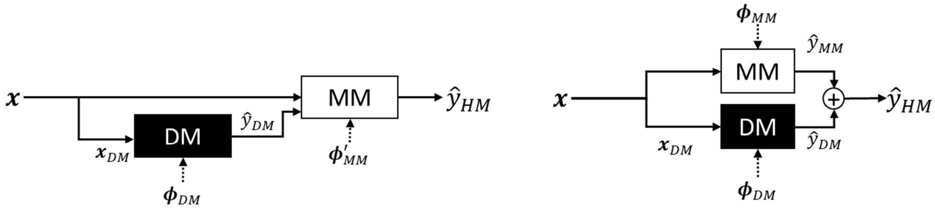
$$\hat{y}_{HM} = q_{O,HM} = h(\boldsymbol{x}_{HM}; \boldsymbol{\phi}_{HM}) \in \mathbb{R}, \tag{14}$$

where $\boldsymbol{x}_{HM} \subseteq \boldsymbol{x}$ and the hybrid model parameters $\boldsymbol{\phi}_{HM}$ is all of $\boldsymbol{\phi}_{DM}$ but not necessarily all of $\boldsymbol{\phi}_{MM}$ since some are redundant when introducing the DM in the MM. For instance, replacing (4) with a DM, the parameter M_G is no longer needed in the equations.

The five HMs may be illustrated with the following figures, variant 1-4 in Figure 3a, here $\boldsymbol{\phi}'_{MM} \subseteq \boldsymbol{\phi}_{MM}$, and HM variant 5 in Figure 3b. It should be noted that the framework used to develop the gray-box models are not restricted to the variants in Figure 3b. For instance, only small changes to the model are necessary to implement black-to-gray VFM models.

The applied data-driven model for all the hybrid model variants is a fully connected, feed-forward neural network. Naturally, other data-driven methods may be applied such as regression trees or support vector machines. Nevertheless, as mentioned in Section 1, neural networks are flexible and can adapt to arbitrarily complex patterns in data. Furthermore, the neural network is easily integrated into a model development framework where the model parameters are found with maximum a posteriori estimation and stochastic gradient-based optimization. This will be introduced in Section 3. In short, a feed-forward neural network is a collection of L layers, represented with the following equations:

$$\begin{aligned} \text{Input} \quad & z_0 = \boldsymbol{x}_{DM} \\ \text{Hidden layer(s)} \quad & \boldsymbol{z}_i = a_i(\boldsymbol{W}_i \boldsymbol{z}_{i-1} + b_i), \quad i \in \{1, .., L-1\} \\ \text{Output layer} \quad & z_L = \boldsymbol{W}_L \boldsymbol{z}_{L-1} + b_L \end{aligned} \tag{15}$$



(a) HM 1-4

(b) HM 5

Figure 3: Illustration of the five hybrid model variants. (a) Hybrid model variant 1-4, (b) Hybrid model variant 5: additive error model.

At each layer, the inputs are transformed with a linearly affine function with weight matrix $\boldsymbol{W}_i$ and bias b_i and sent through an activation function a. The rectified linear unit activation function has been used, which is the elementwise maximum operator $\text{ReLU}(\boldsymbol{z}_i) = \max\{0, \boldsymbol{z}_i\}$. This results in the neural network being a set of piecewise linear equations. The nonphysical parameters of the network are the collection of weights and biases on all layers $\boldsymbol{\phi}_{DM} = \{(\boldsymbol{W}_1, b_1), \dots (\boldsymbol{W}_L, b_L)\}$.

3 Parameter estimation of hybrid models

Regardless of the location of the model on the gray-scale in Figure 1, the uncertain model parameters should be estimated from data. For a fully mechanistic model, good prior values on the parameters often exist and parameter estimation is not a requirement, although usually a necessity, for high accuracy model predictions. For a fully data-driven model, parameters are initialized randomly and parameter estimation is a requirement. Thus, the latter argumentation applies to hybrid models. Parameter estimation is also referred to as model training.

3.1 Maximum a posteriori estimation

Consider a dataset $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$ with n measurements of the process explanatory variables $\boldsymbol{x}_i = (x_{i,1}, \dots x_{i,d}) \in \mathbb{R}^d$, and target variable $y_i \in \mathbb{R}$. Assume the process to be described by the following measurement model

$$y_i = h(\boldsymbol{x}_i; \boldsymbol{\phi}) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_{\epsilon,i}^2) \quad i \in \{1, .., n\}, \tag{16}$$

where $\hat{y}_i = h(\boldsymbol{x}_i; \boldsymbol{\phi})$ are the model predictions of the target variable, with model parameters $\boldsymbol{\phi} \in \mathbb{R}^m$ and normally distributed measurement ϵ_i with zero mean and variance $\sigma^2_{\epsilon,i}$. Observe that this measurement model can incorporate output measurement from different devices, if available, by changing $\sigma^2_{\epsilon,i}$ appropriately for measurement i. Even synthetic data generated with mechanistic simulators may be included in this approach.

In parameter estimation problems, the parameters $\boldsymbol{\phi}$ of the model h will be inferred using the available data $\mathcal{D}$. This can be done using Bayesian inference where the prior parameter distribution $p(\boldsymbol{\phi})$ is updated to a posterior parameter distribution:

$$p(\boldsymbol{\phi}\,|\,\mathcal{D}) = \frac{p(\mathcal{D}\,|\,\boldsymbol{\phi})p(\boldsymbol{\phi})}{p(\mathcal{D})}. \tag{17}$$

Equation (17) includes intractable integrals (Blei, Kucukelbir, and McAuliffe, 2017) and approximation techniques are commonly required for a numerical solution. In this research, maximum a posteriori (MAP) estimation is applied.

In MAP estimation, only the mode of the posterior distribution is considered and the parameters are found with the following optimization problem:

$$\boldsymbol{\phi}^{\star}_{MAP} = \arg\max_{\boldsymbol{\phi}} p(\boldsymbol{\phi}\,|\,\mathcal{D}) = \arg\max_{\boldsymbol{\phi}} \Big[\log p(\mathcal{D}\,|\,\boldsymbol{\phi}) + \log p(\boldsymbol{\phi}) \Big], \tag{18}$$

where $\log p(\mathcal{D}\,|\,\boldsymbol{\phi})$ is called the loglikelihood of the model. By further assuming normally distributed parameter priors $\phi_i \sim \mathcal{N}(\mu_i, \sigma_i^2), i \in \{1, .., m\}$ the following optimization problem may be derived (Bishop, 2006):

$$\boldsymbol{\phi}^{*}_{MAP} = \arg\min_{\boldsymbol{\phi}} \left[\sum_{i=1}^{n} \frac{1}{\sigma^2_{\epsilon,i}} \left(y_i - f(\boldsymbol{x}_i; \boldsymbol{\phi})\right)^2 + \sum_{i=1}^{m} \frac{1}{\sigma_i^2} \left(\phi_i - \mu_i\right)^2 \right]. \tag{19}$$

In short, MAP estimation is a trade-off between minimizing the error between target variable predictions and measurements and minimizing parameter deviation from the prior mean μ. By setting a constant noise level $\sigma^2_\epsilon = const.$, the MAP estimation is equal to maximum likelihood estimation (MLE) with ℓ_2-regularization, a common approach in the data-driven modeling domain (Goodfellow, Bengio, and Courville, 2016). The variance of the parameters and measurement noise determine the degree of regularization. In Hotvedt, Grimstad, and Imsland (2021), it was shown that MAP estimation is necessary for a hybrid model to obtain plausible and physically consistent values of the physical model parameters after estimation. Further, regularization must be used to avoid overfitting of the model and ensure adequate generalization performance (Goodfellow, Bengio, and Courville, 2016).

A different perspective of the MAP estimation problem is that it balances learning from physics and learning from data. With softer regularization, achieved by setting flat, noninformative prior parameter distributions $\sigma_i \to \infty$, the data will have a large influence on the estimation outcome. This is because the regularization terms are down-weighted in optimization. The same effect is achieved with a small noise variance, implying that the measurements are accurate. With harder regularization, the opposite effect is achieved where the physics, in this case, the parameter priors, will have a higher influence on the estimation outcome and the adaption to data down-weighted in the optimization.

For the HMs in Section 2, the MAP objective function is divided into three terms, the MLE and two parameter regularization terms, one each for the physical and nonphysical parameters. In this research, only MPFM measurements are used and thus:

$$\begin{aligned}\boldsymbol{\phi}^*_{MAP} = \arg\min_{\boldsymbol{\phi}} &\sum_{i=1}^{n} \Big(y_i - h(\boldsymbol{x}_{i,HM}; \boldsymbol{\phi}_{HM})\Big)^2 \\ &+ \sigma_\epsilon^2 \left[\sum_{i=1}^{m_1} \left(\frac{\phi_{i,MM} - \mu_{i,MM}}{\sigma_{i,MM}} \right)^2 + \sum_{i=1}^{m_2} \left(\frac{\phi_{i,DM} - \mu_{i,DM}}{\sigma_{i,DM}} \right)^2 \right].\end{aligned} \tag{20}$$

Here m_1 and m_2 is the number of physical and nonphysical parameters, respectively.

3.2 Priors on the physical parameters

For the physical model parameters, good prior values of the mean $\mu_{i,MM}$ often exist. For instance, for freshwater density $\mu_{\rho_w} \approx 1000$ kg/m^3. The parameter variances may be set to reflect the uncertainty in the prior mean value. If the assumption of normally distributed parameters is exploited, the variance may be approximated using the absolute maximum and minimum values of the parameters and calculating the 6σ band of the distribution,

$$\sigma_{i,MM} = \frac{\max(\phi_{i,MM}) - \min(\phi_{i,MM})}{6}, \tag{21}$$

for which the probability of obtaining values outside the band is $\approx 0.03\%$. For harder regularization of a specific parameter, the variance may be decreased, resulting in a sharper distribution, and the opposite for softer regularization.

3.3 Priors on the nonphysical parameters

Finding priors for the nonphysical parameters in the model is not trivial. However, He-initialization is recommended for neural networks with ReLU as activation function (He et al., 2015). With He-initialization, each element in the weight matrix on each layer $\boldsymbol{W}_i, i \in L$ (see Section 2) is initialized from a normal distribution with mean and variance

$$\mu_{DM} = 0, \qquad \sigma^2_{DM} = \left(\sqrt{\frac{2}{m_{l,i}}}\right)^2), \quad i \in 2, ..L, \tag{22}$$

where $m_{l,i}$ are the number of inputs on layer i. On the first layer, no activation function is applied to the inputs and $\sigma^2_{DM} = \left(\sqrt{1/m_{l,1}}\right)^2$).

On the other hand, for the hybrid models where the neural network represents a mechanistic relation, more informative priors for the nonphysical parameters are found by pretraining the network on synthetic data generated with the mechanistic relation in question. The obtained values of the weights and biases of the pretrained network are used as the $\boldsymbol{\mu}_{DM}$ when training the final model. However, the the network is trained on synthetic data only and it assumed that the updated prior parameter means are just as uncertain as before. Therefore, the parameter variances in (22) are utilized. If real measurements of the variable existed, such as density measurements, these could be used in the pretraining.

3.4 Priors on the measurement noise

In an industrial setting, a common measure of the error of a measurement device is the mean absolute percentage error (MAPE), comparing the measured signal to a known reference value y_{ref}. Following the derivation in Grimstad, Hotvedt, et al. (2021), the MAPE may be translated into the variance of the measurement noise with

$$\sigma_\epsilon^2 = \left(\sqrt{\frac{\pi}{2}} \alpha |y_{ref}| \right)^2 , \tag{23}$$

where α is the MAPE, for instance $\alpha = 0.1$ for 10% MAPE. In this study, the reference value is not known and the variance of the measurement noise is approximated by using the available data. Because the MAP estimation in (20) assumes a constant noise level $\sigma_\epsilon^2 = const.$, the mean value of the measured target variable in the training data is used as the reference value, $y_{ref} = 1/n \sum_{i=1}^{n} y_i$. As mentioned in Section 3.1, in practice the σ_ϵ^2 may be adjusted to influence the degree of regularization on the parameters.

4 Case study

The case study develops the five listed white-to-gray VFM models in Section 2 for 10 petroleum wells on Edvard Grieg (Lundin Energy Norway, 2020). Edvard Grieg is an asset on the Norwegian Continental Shelf and consists of under-saturated oil without a gas cap. The asset is relatively new where production commenced in 2015. The wells, hereafter referred to as W01-W10, are well-instrumented with available measurements of the explanatory variables defined in (11). An MPFM located in the wellhead of each well provides measurements of the volumetric flow rate. The models are trained with MAP estimation introduced in Section 3 using real, historical production data from the 10 wells. The number of data samples per well is unequal and spans approximately 1.5-4 years. No additional experimental or synthetic data are considered. For comparison, the Sachdeva model in Section 2, and a fully connected feed-forward neural network, are implemented. Two aspects of the models are investigated. First, the predictive performance in terms of accuracy is analyzed in Section

4.1. Thereafter, the scientific consistency is examined in Section 4.2. Considerations for improvements in future work are discussed in Section 4.3.

The datasets for each well are preprocessed in two steps. First, the processing technology in Grimstad, Gunnerud, et al. (2016), is utilized to generate a compressed dataset of steady-state operating points suitable for steady-state modeling. Secondly, a set of filters are applied to remove data samples that likely originate from erroneous sensor data, such as negative pressures or choke openings. The dataset is split into training and test set according to time to mimic an industrial setting where the developed models are used to predict the future responses of the process. The test set consists of the three latest months of the data samples. The regularization method early stopping (Goodfellow, Bengio, and Courville, 2016) is utilized to train the models. This algorithm monitors the error on a validation dataset during model training to find the appropriate number of loops through the training data, called epochs, to train the model without overfitting. The validation data is 20% of the training data, extracted in randomly chosen chunks, each chunk representing data samples from two chronological weeks. Due to the stochasticity of the training algorithm, the early stopping algorithm is run several times, and the average number of epochs is used to train the final model. The optimizer Adam (Kingma and Ba, 2015) is applied with mini-batches, and the learning rate is $\alpha = 10^{-4}$.

An overview of the seven implemented models is found in Table 1. The table illustrates which mechanistic model parameters $\boldsymbol{\phi}'_{MM} \subseteq \boldsymbol{\phi}_{MM}$, are present in the model, which factor or term is replaced by a neural network g, and which measurements $\boldsymbol{x}_{DM}$ are used as input to the data-driven element. For short, the hybrid models are named $HM(\star)$, where $\star$ is the factor or term the neural network substitutes. The fully mechanistic and the fully data-driven model are referred to as the MM and the DM respectively. For all neural networks, the network depth and width are set to 3 × 100. The size may be excessive for some of the models. Nonetheless, following recommendations from (Bengio, 2012) the size can be set arbitrarily large as long as regularization is employed to prevent overfitting. For the HM(A_2), HM($\rho_{G,1}$), HM($\rho_{G,2}$), and HM(ρ), the neural networks are pretrained with synthetic data before utilized in the final model. For each of the final 70 choke models (for 10 wells and 7 model types), the parameters are initialized using the prior parameter distributions

Table 1: An overview of the developed models of the production choke valve: five hybrid, one fully mechanistic, and one fully data-driven model.

VFM model	$\boldsymbol{\phi}'_{MM}$	$g(\boldsymbol{x}_{DM}; \boldsymbol{\phi}_{DM})$	$\boldsymbol{x}_{DM}$
MM	$\rho_O, \rho_W, \kappa, M_G, p_{r,c}, C_D$	n.a.	n.a.
HM(A_2)	$\rho_O, \rho_W, \kappa, M_G, p_{r,c}$	Area function	u
HM($\rho_{G,1}$)	$\rho_O, \rho_W, \kappa, p_{r,c}, C_D$	Upstream gas density	p_1, T_1
HM($\rho_{G,2}$)	$\rho_O, \rho_W, M_G, p_{r,c}, C_D$	Gas expansion	p_1, p_2, T_1, T_2
HM(ρ)	$\rho_O, \rho_W, \kappa, M_G, p_{r,c}, C_D$	Mixture density	$p_1, p_2, T_1, T_2, \eta_G, \eta_O$
HM(ε)	$\rho_O, \rho_W, \kappa, M_G, p_{r,c}, C_D$	Additive error	$p_1, p_2, T_1, T_2, \eta_G, \eta_O$
DM	n.a.	Oil flow rate	$p_1, p_2, T_1, T_2, u, \eta_G, \eta_O$

described in Section 3.2 and 3.3. The variance of the measurement noise σ_ϵ^2 is calculated assuming a MAPE of 10% and following the procedure in Section 3.4. A trick is utilized to enforce the positivity of the physical model parameters. A temporary parameter S is learned instead of the real parameter ϕ, and the transformation

$$\phi_i = \exp\left(S_{\phi_i} + \zeta\right), \quad \text{for } i = 1, .., m, \tag{24}$$

is used to obtain the real parameter value. Here ζ is a small constant to avoid vanishing gradients in the optimization problem.

4.1 Predictive performance

In Figure 4, the mean absolute percentage error (MAPE) is calculated for each choke model and illustrated in a box plot comparing the different model types. Table 2 shows a detailed view of the MAPEs for the individual choke models. For the interested reader, the predicted volumetric flow rates are illustrated together with the measured flow rate (downscaled) in A, Figure 1.

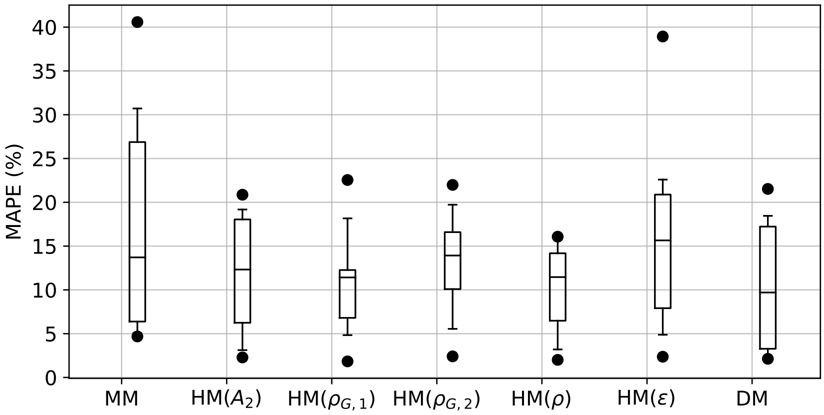


Figure 4: Box plot of the mean absolute percentage error for each model across all wells. The horizontal line in the box is the median performance.

There are several interesting observations to make. Firstly, the median errors are large for all model types and not at the level with the reported errors in literature, see Section 1. Figure 4 shows that the DM is the only model achieving a median MAPE below 10%, though barely with 9.4%. Secondly, the results indicate that moving the model on the gray-scale from white to gray does improve the average performance significantly, see Table 2. The MM achieves an error of 17.2% against

10.3% for the best HM. However, comparing the HMs to the DM with an error of 10.4%, there is only a small improvement. Thirdly, large variations in performance for the different choke models are observed in Table 2. For instance, for W01, all the model types perform excellently and are on the level with the reported errors in the literature (less than 4% MAPE). Yet, for W02, the performance is unsatisfactory for all model types. The large differences in performance may also be observed by looking at the cumulative deviation plots in Appendix A, Figure 2. This plot shows the percentage of test points that fall within a certain percentage deviation from the true value (Corneliussen et al., 2005).

Table 2: Mean absolute percentage error for the individual choke models. The best performing choke model is highlighted in bold.

	MM	HM(A_2)	HM($\rho_{G,1}$)	HM($\rho_{G,2}$)	HM(ρ)	HM(ε)	DM
W01	4.7	3.1	**1.8**	2.4	2.0	4.9	2.4
W02	28.9	16.7	**11.2**	19.7	14.4	18.0	17.7
W03	20.8	**9.3**	18.2	16.7	16.1	22.6	15.7
W04	**5.7**	18.5	11.6	15.5	13.7	13.6	18.5
W05	8.3	11.9	12.5	22.0	16.1	17.4	**3.6**
W06	40.6	20.9	6.5	9.9	9.3	38.9	**3.7**
W07	30.7	2.3	4.8	5.5	5.7	6.0	**2.1**
W08	5.1	5.2	7.7	10.7	3.2	**2.4**	3.2
W09	12.7	12.8	11.6	12.3	**8.8**	13.9	21.5
W10	14.7	19.2	22.5	16.3	**13.6**	21.9	15.8
Across wells	17.2	12.0	10.9	13.1	10.3	16.0	10.4

There are several factors that may cause the observed prediction accuracy of the different models. Three of these will be discussed in the following. Section 4.1.1 will focus on the impact model simplifications may have on the accuracy, Section 4.1.2 will elaborate on the task of balancing learning from physics and learning from data, and Section 4.1.3 discusses the likely influence of available data.

4.1.1 The possible impact of model simplifications

First of all, it must be kept in mind that only the production choke valve is modeled, and any effects of the remaining production system on the multiphase flow, such as the wellbore, are disregarded. It is believed that the average predictive performance would improve by modeling a larger part of the production system. Second of all, several assumptions and simplifications are introduced in the baseline mechanistic choke model. Dependent on process conditions, flow regimes, and fluid composition, these may be appropriate to describe the physical behavior of the flow through the choke in some wells but imprecise in others. For instance, observe how the HM(A_2) for W03 has a much better performance than any of the other model types. This may indicate that the mechanistic area function is poorly calibrated for this well in the other model types. For W01, HM($\rho_{G,1}$) has the best performance and may suggest that the assumption of the real gas law is inadequate. Naturally, these are only indications and the results could benefit from a deeper analysis of the suitability of different hybrid models in different cases.

4.1.2 The nontrivial task of balancing learning from physics and data

With adequate design and training, the HMs were expected to exploit both physics and data to their full extent and thereby perform better than non-hybrid models. Certainly, on a well level, six wells perform better with an HM. However, seen from Table 2, wells W04-W07 perform better with either a mechanistic or a data-driven model. This may cast light upon the nontrivial task of balancing learning from physics and data. The HM may be too simplistic, and consequently, not flexible enough to capture complex physical behavior. Likewise, the data-driven elements may be erroneously influenced by the data. Hence, an appropriate approach to control the influence of the mechanistic and data-driven component is yet to be discovered, at least for the white-to-gray hybrid model types investigated in this research.

4.1.3 The influence of the available data

As neural networks have the power to adapt to arbitrarily complex patterns in the data, the large MAPEs seen for many of the DMs may indicate that the quality of the available data is inadequate. Real, historical production data are used in both model training and testing. It is not uncommon that production data are noisy and biased, which complicates the modeling process and may yield an unfair indication of predictive performance for some models. Naturally, different model types or estimation techniques exist which to a greater extent exploits uncertainty in the model parameters and measurements. On the other hand, such methods require specifications of uncertainty that are not easily available, and the resulting models are usually of higher complexity. Further, it is believed that the large error for several of the choke models is mainly caused by the datasets originating from the underlying, nonstationary process. In time with the reservoir being depleted, the pressure in the down-hole will decrease. If the goal is to maintain a steady production rate, the operators must increase the choke opening. Extracting the test dataset chronologically may therefore result in a set of process conditions that are substantially different from the conditions seen in the training dataset. If so, a steady-state model like the baseline mechanistic model or a standard neural network will not be able to capture the slowly varying, underlying changes.

Figure 5 illustrates this issue. Shown is the upstream pressure p_1 versus the choke opening u for approximately the same oil volumetric flow rate. The coloring indicates time, the lightest colors are the latest time samples. Notice that for some wells (for example W05, W06, W07), the coloring is grouped, indicating that in time, different process conditions are required to maintain the volumetric oil flow rate. Naturally, the flow rate will also depend on other variables such as the mass fractions. Nevertheless,

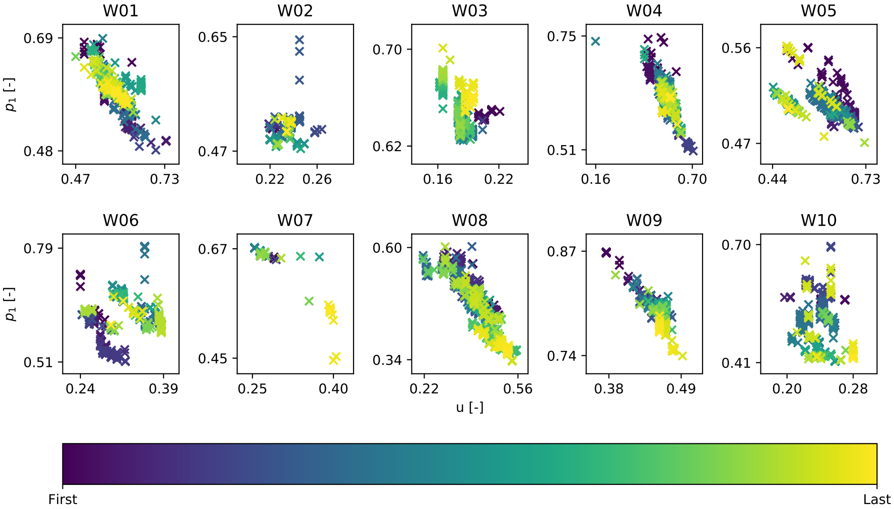


Figure 5: Upstream pressure versus choke opening in time for approximately the same volumetric oil flow rate. Dark colors are the earliest time samples, whereas the light colors are the latest and are included in the test set.

in a nonstationary situation, using three months of test data and assuming the model parameters to be constant and representative for the physical behavior during three months may be inappropriate. It may also discredit the high accuracy prediction potential of the models. Using the developed models to predict the process response only one week ahead greatly increases the accuracy, see the comparison of three months prediction against one-week predictions in Figure 6.

4.2 Scientific consistency

One consideration of a model is the performance in terms of accuracy, another is the scientific consistency. Inconsistent physical behaviors may cast doubt about the trustworthiness of the models and cause the generalization abilities to be poor. First, the outputs from the neural networks in the hybrid models are investigated. Figures 7a and 7b shows the output from the neural network in HM(A_2) and HM($\rho_{G,1}$), respectively, as a function of one of the inputs, for three of the wells. The results are diverse. In some of the choke models, the output of the neural network has a trend coherent with the expected physical behavior, illustrated with the mechanistic relation. This is seen for W01. However, notice that some of the other curves go to zero or explode, illustrating scientific inconsistency. This effect has also been

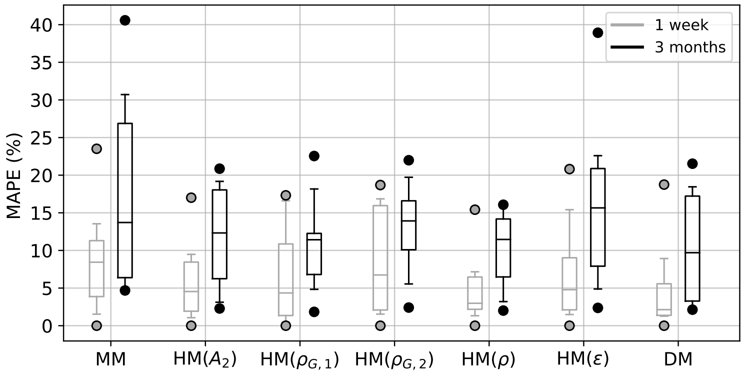
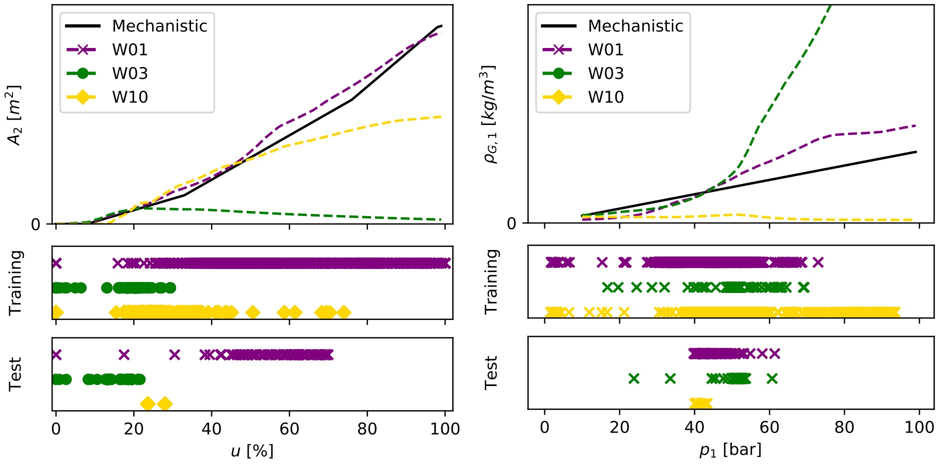


Figure 6: Box plot comparing the mean absolute percentage error for each model across wells using three months of data in black (right box) and one week of data in gray (left box).

observed for the HM($\rho_{G,2}$) and the HM(ρ). There are two likely explanations for the nonphysical behaviors. Firstly, the behavior may be influenced by the lack of data or erroneous data. For instance, for W03, data are lacking for choke openings greater than 40%. Secondly, due to the high capacity of neural networks, the data-driven part of a hybrid model may capture any modeling error and not just the factor or term the network was intended to represent. For instance, even though the HM(A_2) had the best performance for W03 of all models, the area function is not in line with the expected physical behavior. This indicates that the learned neural network area function may have captured other modeling errors than just a poorly calibrated area function.

Additionally, a short sensitivity study is conducted to investigate the scientific consistency of the output of the seven implemented VFM models. The choke models trained on data from W01 are examined for which all models achieved a good performance, see Table 2. Five test points are randomly picked from the test dataset, the choke opening u and the upstream pressure p_1 are individually perturbed and the responses in the oil volumetric flow rate q_O are investigated. Under the assumption of constant process conditions and considering the production choke as an isolated unit without the influence of the rest of the production system, the oil flow rate should be expected to 1) increase with increasing choke opening, and 2) increase with increasing upstream pressure. The sensitivity study is presented in Figure 8.

Most of the models seem to mimic the expected physical behavior except for the DM, for which the oil flow rate decreases with increased pressure above a certain threshold. This effect is caused by the DM being influenced by the available data to a larger degree than the other model types, and that the available data reflects the behavior



(a) HM(A_2)

(b) HM($\rho_{G,1}$)

Figure 7: The learned neural network area function in (a) H(A_2) (b) H($\rho_{G,1}$), and for three of the wells, illustrated together with a typical mechanistic curve (black, solid). Also shown are the training and test data points for each well.

of the complete production system and not only the choke. This can be explained in more detail by looking at the correlation plot of the available measurements in the dataset corresponding to W01, see Figure 9. Observe the negative correlation between the oil flow rate q_O and the upstream pressure p_1. By looking at the choke as an isolated unit this correlation contradicts the expected physical behavior. On the other hand, additionally considering the wellbore, the observed correlation has a scientific explanation: increased pressure in the wellhead may result in a decreased pressure drop in the wellbore and a decreased oil flow rate. Nevertheless, if the goal of the modeling was to develop a choke model, the DM would be considered scientifically inconsistent. These results reflect upon both the positive and negative nature of models with high flexibility. They may adapt to any behavior seen in the available data, thus also erroneous data. On the other side, this sensitivity study is small and only conducted for one well. Conclusions on the scientific consistency of the general gray-box model cannot be made. Nevertheless, the results motivate the use of gray-box VFM models if scientific consistency is of importance to the end-users of the models.

4.3 Suggestions for improvements in future work

From the results presented in Section 4 there are several aspects that can be investigated to improve upon both the prediction accuracy and the scientific consistency of hybrid models in future work.

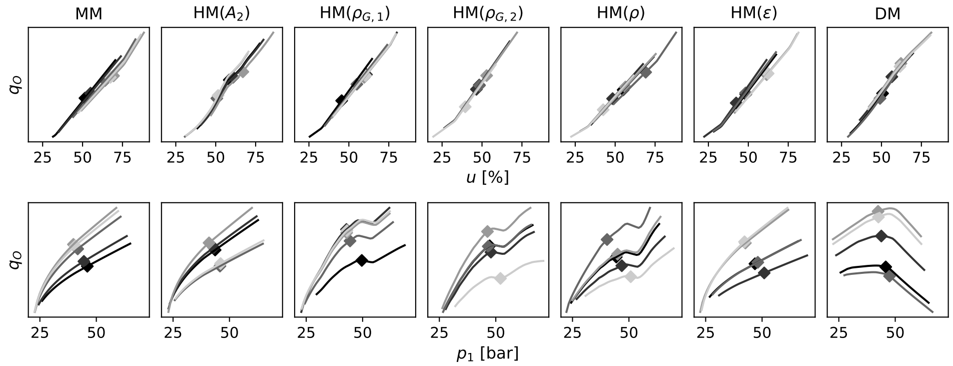


Figure 8: Sensitivity analysis of the different models for W01. Five initial points are picked at random, marked with diamond, and the response of the volumetric oil flow rate when perturbing the choke opening u (upper) and the upstream pressure p_1 (lower) is illustrated.

Firstly, only a few simplifications and assumptions are investigated as hybridization options in Section 2.2 although numerous exist. It is likely that other hybrid model types may be better at balancing the task between learning from physics and learning from data. Further, different types of data-driven models or other mechanistic choke models may yield better performances for these wells. There is also the question raised in Section 4.1.1 on the suitability of different hybrid models in different cases. One approach in this direction is to utilize an advanced simulator to generate synthetic data, in which process conditions and other characteristics can be controlled.

Secondly, Section 4.1.3 discussed the influence of the available data on the prediction accuracy and pointed out noisy and biased measurements, together with nonstationary process conditions as influential factors. A future research path is to experiment with different estimation methods or model types that exploits knowledge regarding the uncertainty in parameters and measurements. Some examples are variational inference as estimation method, state estimation techniques such as the Kalman Filter (Kalman, 1960), or probabilistic models. In case of nonstationary process conditions, time dependent models may be utilized. Yet, such models greatly increase the computational complexity and may not be suitable for real-time applications. Another possibility is online learning, a learning method that may improve upon future predictive performance without adding complexity to the models.

Lastly, in Section 4.2, the scientific consistency of the gray-box models were discussed and several issues raised. Several possible approaches may be investigated to improve upon the scientific consistency. Firstly, a stronger regularization of the priors obtained from the pretrained neural networks could possibly result in the network replicating the mechanistic relation to a higher degree, whilst avoiding capturing other modeling

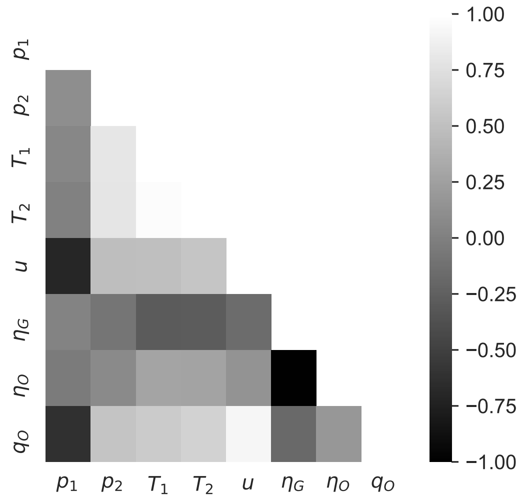


Figure 9: A visualization of the correlation between the explanatory variables and the target variable measurements in the dataset corresponding to W01.

errors. Secondly, the inclusion of additional data-driven elements in a gray-box model, for instance, an error term, could enable the original data-driven element to capture the proposed physics only. Thirdly, the utilization of methods that enables learning from datasets across wells, for instance transfer learning or multitask learning, may positively change the results as more data are exploited.

5 Concluding remarks

This article contributes towards the development of gray-box virtual flow meters in the petroleum industry. The focus has been on white-to-gray box models where a mechanistic model is used as a baseline and data-driven elements inserted to increase model flexibility. The choke valve of 10 petroleum wells has been modeled using real production data spanning at most four years of production.

The results are diverse with a prediction accuracy is in the range of 1.8%-40.6%, and no recommendations towards the suitability of different gray-box models may be drawn. The results cast light upon the nontrivial task of balancing learning from both physics and data. It is believed that the accuracy is strongly influenced by nonstationarity in the available data. Nevertheless, the results indicate that gray-box models may outperform a mechanistic and a data-driven model if an appropriate balance between the model components is identified. In particular, the gray-box modeling approach seems to increase the accuracy compared to mechanistic models and may improve the scientific consistency compared to data-driven models.

While the gray-box modeling approaches are tested on 10 different wells, these wells, while being fairly typical offshore wells, are hardly representative for all wells. Therefore, a direct generalization of the results to other assets is difficult. Assuredly, the results could benefit from a deeper analysis of gray-box modeling on wells with significantly different characteristics. Furthermore, the research has studied the approach with VFM as application, and generalization to other application areas is inadmissible without further experimentation. On the other side, the gray-box modeling approach itself should apply to any process systems where both physical equations and process data exist.

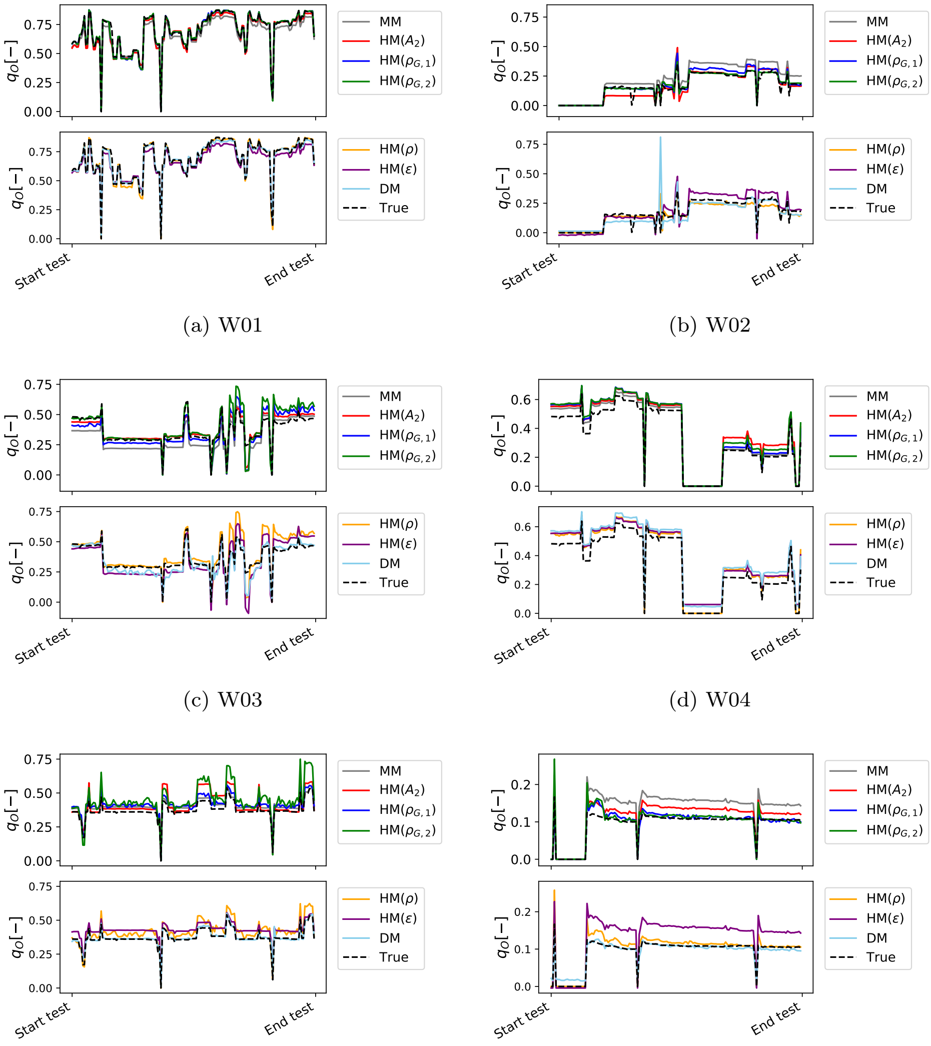
To this end, the results reported in this study are promising, albeit, the true potential of gray-box modeling is yet to be discovered. For example, hybrid modeling could yield great potential in the small data regime, where data-driven models are known to struggle. Several interesting research directions exist for future consideration. Among these are online learning and multi-task learning.

Acknowledgment and funding

This research is a part of BRU21 - NTNU Research and Innovation Program on Digital and Automation Solutions for the Oil and Gas Industry (`www.ntnu.edu/bru21`) and supported by Lundin Energy Norway. Lundin Energy Norway had no part in data collection and analysis, nor in writing of the report. Yet, they approved the paper before submission for publication.

The authors would like to thank Lundin Energy Norway for the opportunity to work with and publish results related to data from the asset Edvard Grieg. We would further like to thank Solution Seeker AS for utilization of their data processing technology, saving valuable time on data curation.

A Case study - results



(a) W01

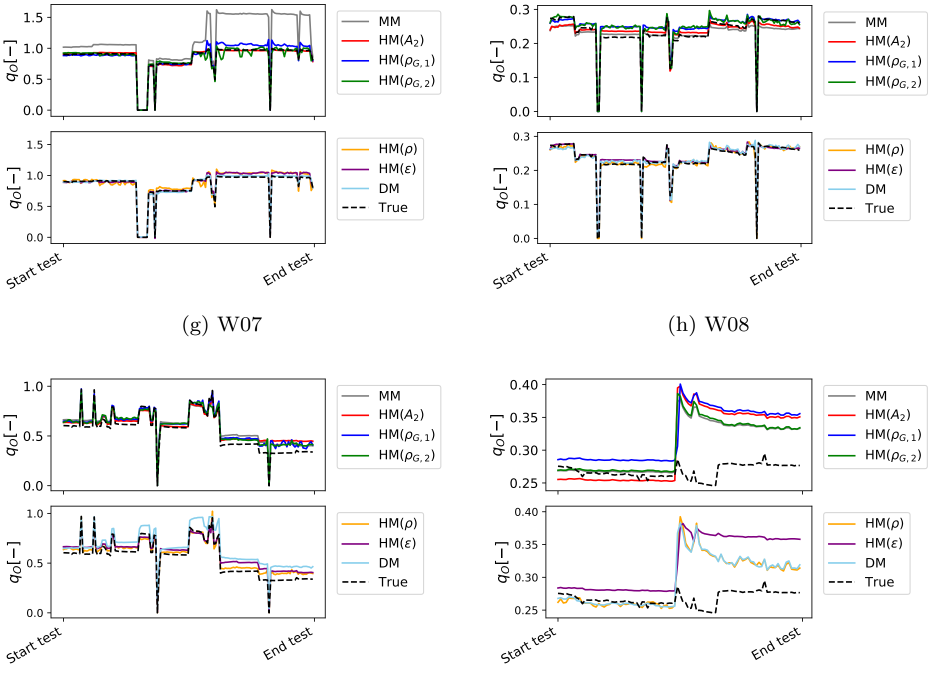
(b) W02

(c) W03

(d) W04

(e) W05

(f) W06



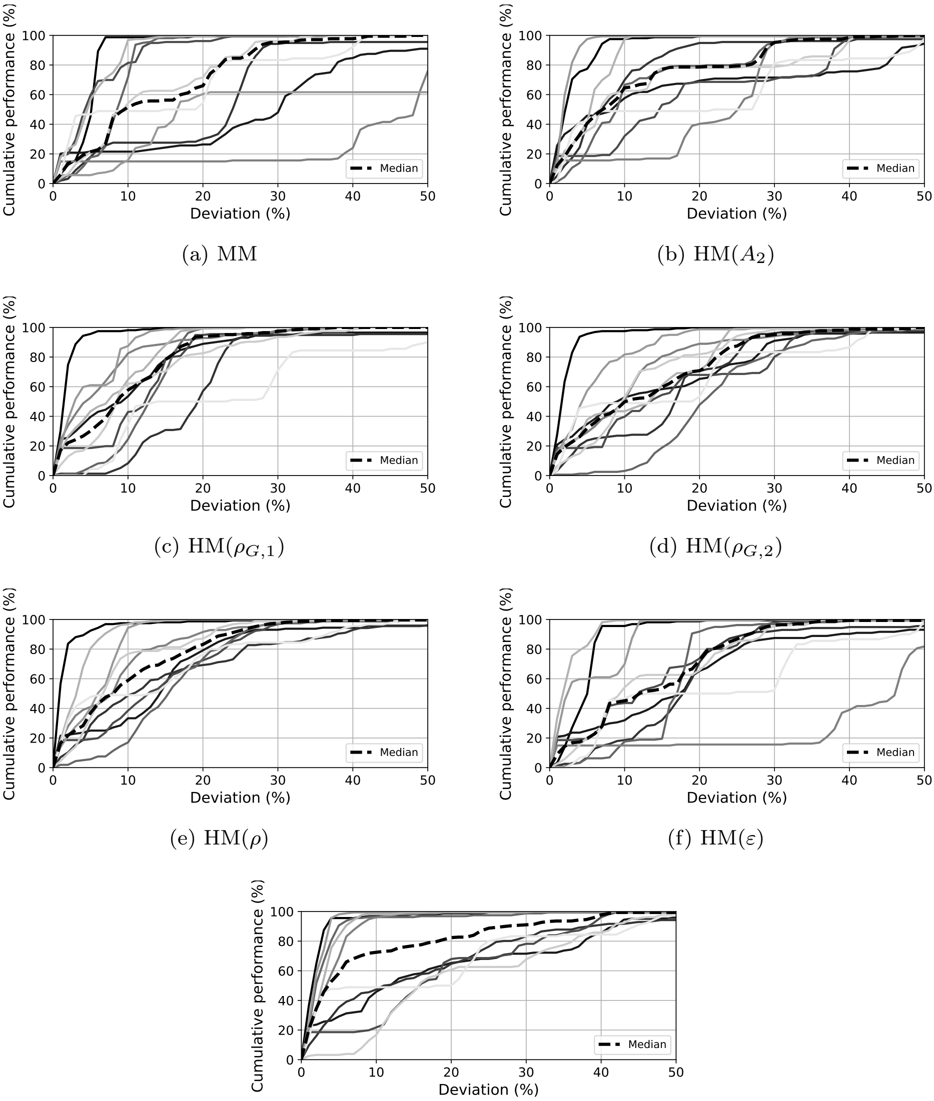
(g) W07

(h) W08

(i) W09

(j) W10

Figure 1: Illustration of the (downscaled) volumetric oil flow rate for each of the well and all models. Shown in dotted black are the measured flow rate from the multiphase flow meter. Notice, for some of the wells all models have adequate prediction accuracy, whilst for other wells, some model predictions are unsatisfactory.



(a) MM

(b) HM(A_2)

(c) HM($\rho_{G,1}$)

(d) HM($\rho_{G,2}$)

(e) HM(ρ)

(f) HM(ε)

(g) DM

Figure 2: Cumulative performance of choke models grouped on the model types. The black dotted line shows the median performance across wells.

References

AlAjmi, M.D., S. A. Alarifi, and A. H. Mahsoon (Jan. 2015). "Improving Multiphase Choke Performance Prediction and Well Production Test Validation Using Artificial Intelligence: A New Milestone". In: *SPE Digital Energy Conference and Exhibition*, pp. 22–29.

Amin, A. (2015). "Evaluation of Commercially Available Virtual Flow Meters (VFMs)". In: *Proceedings of the Annual Offshore Technology Conference*, pp. 1293–1318.

Bengio, Y. (2012). "Practical Recommendations for Gradient-Based Training of Deep Architectures." In: *arXiv:1206.5533v2*, pp. 1–33.

Bikmukhametov, T. and J. Jäschke (2019). "Oil Production Monitoring using Gradient Boosting Machine Learning Algorithm." In: *IFAC-PapersOnLine* 52 (1), pp. 514–519.

Bikmukhametov, Timur and Johannes Jäschke (2020a). "Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models". In: *Computers and Chemical Engineering* 138.

— (2020b). "First Principles and Machine Learning Virtual Flow Metering: A Literature Review". In: *Journal of Petroleum Science and Engineering* 184.

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning.* Springer.

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe (2017). "Variational Inference: A Review for Statisticians". In: *J. Am. Stat. Assoc.* 112.518, pp. 859–877.

Corneliussen, Sidsel et al. (2005). *Handbook of multiphase flow metering.* The Norwegian Society for Oil and Gas Measurements.

Ghorbani, H. et al. (2018). "Prediction of oil flow rate through an orifice flow meter: Artificial intelligence alternatives compared". In: *KeAI, Petroleum.*

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning.* Cambridge, Massachusetts. London, England.: The MIT Press.

Grimstad, B., V. Gunnerud, et al. (Sept. 2016). "A Simple Data-Driven Approach to Production Estimation and Optimization". In: *In: SPE Intelligent Energy International Conference and Exhibition.*

Grimstad, B., M. Hotvedt, et al. (2021). "Bayesian Neural Networks for Virtual Flow Metering: An Empirical Study". In: *Applied Soft Computing* 112.

He, Kaiming et al. (2015). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

Hotvedt, M., B. Grimstad, and L. Imsland (2020). "Developing a Hybrid Data-Driven, Mechanistic Virtual Flow Meter - a Case Study". In: *IFAC-PapersOnLine* 53 (2), pp. 11692–11697.

— (2021). "Identifiability and interpretability of hybrid, gray-box models". In: *IFAC-PapersOnLine* 54 (3), pp. 389–394.

International Organization for Standardization (Dec. 1996). *Natural gas - Standard reference conditions.* Standard. International Organization for Standardization.

Jansen, Jan-Dirk (2015). *Nodal Analysis of Oil and Gas Wells - Theory and Numerical Implementation.* TU Delft, The Netherlands: Delft University of Technology.

Kalman, R.E. (1960). "A New Approach to Linear Filtering and Prediction Problems". In: *Trans. ASME, Journal of Basic Engineering* 82 (Series D), pp. 35–45.

Kanin, E.A. et al. (2019). "A predictive model for steady-state multiphase pipe flow: Machine learning on lab data". In: *Journal of Petroleum Science and Engineering* 180, pp. 727–246.

Kingma, D.P. and J. L. Ba (2015). "Adam: a method for stochastic optimization". In: *International conference on learning representations.*

Kittilsen, P., K. Fjalestad, and R. Aasheim (2014). "Stabilized and Increased Well Production Using Automatic Choke Control". In: *Society of Petroleum Engineers.*

Lundin Energy Norway (2020). *Edvard Grieg.* `https://lundin-energy-norway.com/edvard-grieg/`. Acccessed: 18.01.2020.

Omrani, S. et al. (2018). "Improving the accuracy of virtual flow metering and back-allocation through machine learning". In: *In: Abu Dhabi International Petroleum Exhibition and Conference.*

Perkins, T.K. (1993). "Critical and sub-critical flow of multiphase mixtures through chokes". In: *Society of Petroleum Engineers.*

Prada, C. de et al. (2018). "Developing grey-box dynamic process models". In: *IFAC-PapersOnLine* 51 (2), pp. 523–528.

AL-Qutami, T.A et al. (2018). "Virtual multiphase flow metering using diverse neural network ensemble and adaptive simulated annealing". In: *Expert Syst. Appl.* 93, pp. 72–85.

AL-Qutami, Tareq Aziz, Rosdiazli Ibrahim, and Idris Ismail (2017). "Hybrid neural network and regression tree ensemble pruned by simulated annealing for virtual flow metering application." In: *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 304–309.

AL-Qutami, Tareq Aziz, Rosdiazli Ibrahim, Idris Ismail, and Mohd Azmin Ishak (2017a). "Development of soft sensor to estimate multiphase flow rates using neural networks and early stopping". In: *Int. J. Smart Sens. Intell. Syst.* Vol. 10, pp. 199–222.

— (2017b). "Radial basis function network to predict gas flow rate in multiphase flow." In: *Proceedings of the 9th International Conference on Machine Learning and Computing*, pp. 141–146.

Al-Rawahi, N. et al. (2012). "A neural network algorithm for density measurement of multiphase flow". In: *Multiphase Science and Technology* 24 (2), pp. 89–103.

Roscher, R. et al. (2020). "Explainable Machine Learning for Scientific Insights and Discoveries." In: *arXiv:1905.08883v3*, pp. 1–29.

Ruden, T. (2020). *How we obtain accurate VFM flow rates.* `https://turbulentflux.com/how-we-obtain-accurate-vfm-flow-rates/`. Accessed: 18.01.2020.

Sachdeva, R. et al. (1986). "Two-phase flow through chokes". In: *Society of Petroleum Engineers, Annual Technical Conference and Exhibition.*

Al-Safran, E. M. and M. Kelkar (2009). "Predictions of two-phase critical-flow boundary and mass-flow rate across chokes". In: *Society of Petroleum Engineers* 24 (2).

Selmer-Olsen, S. (1995). “Subsea chokes as multiphase flowmeters:production control at troll olje”. In: *In: Proceedings of the BHR Group 7th Intl.Conference on Multiphase Production. Cannes, France.*

Shippen, M. (2012). “Steady-State Multiphase Flow - Past, Present, and Future, with a Perspective on Flow Assurance”. In: *Energy and Fuels* 26, pp. 4145–4157.

Siemens Process Systems Engineering (2021). *gPROMS Process.* `https://www.psenterprise.com/products/gproms/process`. Acccessed: 18.05.2021.

Solle, D. et al. (2016). “Between the Poles of Data-Driven and Mechanistic Modeling for Process Operation”. In: *Chemie Ingenieur Technik.*

Sutton, R.P. (1985). “Compressibility factors for high-molecular-weight reservoir gases”. In: *SPE Annual Technical Conference and Exhibition.*

Toskey, E.D. (2012). “Improvements to Deepwater Subsea Measurements RPSEA Program: Evaluation of Flow Modeling”. In: *Offshore Technology Conference.*

Willard, J. et al. (2020). “Integrating Physics-Based Modeling With Machine Learning: A Survey”. In: *arXiv:2003.04919v4*, pp. 1–34.

Xu, L. et al. (2011). “Wet gas metering using a revised venturi meter and soft-computing approximation techniques”. In: *IEE transactions on instrumentation and measurement* 60 (3), pp. 947–956.

Zangl, G., R. Hermann, and C. Schweiger (2014). “Comparison of Methods for Stochastic Multiphase Flow Rate Estimation”. In: *Society of Petroleum Engineers.*

E When is gray-box modeling advantageous for virtual flow metering?

Preprint of Hotvedt, Grimstad, Ljungquist, et al. (2022b)

M. Hotvedt, B. Grimstad, D. Ljungquist, and L. Imsland (2022b). “When is gray-box modeling advantageous for virtual flow metering?” In: *Accepted for publication in IFAC-PapersOnLine*

When is gray-box modeling advantageous for virtual flow metering?

M. Hotvedt[1], B. Grimstad[2], L. Imsland[1], and D. Ljungquist[3]

[1]Engineering Cybernetics Department, NTNU, Trondheim, Norway
[2]Solution Seeker
[3]TechnipFMC

Abstract: Integration of physics and machine learning in virtual flow metering applications is known as gray-box modeling. The combination is believed to enhance multiphase flow rate predictions. However, the superiority of gray-box models is yet to be demonstrated in the literature. This article examines scenarios where a gray-box model is expected to outperform physics-based and data-driven models. The experiments are conducted with synthetic data where properties of the underlying data generating process are controlled. The results show that a gray-box model yields increased prediction accuracy over a physics-based model in the presence of process-model mismatch, and improvements over a data-driven model when the amount of available data is small. On the other hand, gray-box and data-driven models are similarly influenced by noisy measurements. Lastly, the results indicate that a gray-box approach may be advantageous in nonstationary process conditions. Unfortunately, model selection prior to training is challenging, and overhead on gray-box model development and testing is unavoidable.

Keywords: gray-box, hybrid model, virtual flow metering, neural networks

1 Introduction

Gray-box modeling is a methodology that integrates physics-based modeling with machine learning techniques in process model development (Willard et al., 2020). The gray-box models are placed on a gray-scale dependent on the degree of integration, ranging from physics-based to data-driven models. A common perception is that physics-based models require little data in development and are more robust to noisy measurement than data-driven models. This perception arguably stems from the high extrapolation capabilities demonstrated by many physics-based models (Oerter, 2006). Nevertheless, complex physical phenomena can be challenging to model in detail using first principles, and simplifications are generally necessary for suitability in real-time control and optimization applications (Roscher et al., 2020). Simplifications reduce the model capacity and thereby the ability to capture complex physical behavior. Therefore, physics-based models often have a bias, or process-model mismatch (Hastie, Tibshirani, and Friedman, 2009).

In contrast, many data-driven models have a large capacity, typically reducing model bias. Furthermore, some data-driven models are computationally cheap to evaluate and are therefore suitable for real-time applications. Moreover, they commonly have lower development and maintenance costs compared to physics-based models (Solle et al., 2016). On the other side, due to the inherent bias-variance trade-off (Hastie, Tibshirani, and Friedman, 2009), a large capacity often results in high variance. High variance causes data-driven models to struggle with extrapolation to future process conditions and to yield low performance in the small data regime (Roscher et al., 2020). Gray-box modeling is expected to leverage the complementary and advantageous properties of physics and data to minimize both bias and variance. In other words, create a model that achieves high performance in the presence of process-model mismatch, little or noisy data, which extrapolates well to previously unseen process conditions and is computationally efficient. Gray-box modeling is similar to introducing strong priors in a data-driven model. In image classification using convolutions neural networks, strong priors in terms of parameter sharing resulted in state-of-the-art performance (Hastie, Tibshirani, and Friedman, 2009).

One application where accurate process models are of high importance is in virtual flow meters (VFMs): a soft-sensor able to predict the multiphase flow rate in real-time at convenient locations in a petroleum asset (Toskey, 2012). The standard practice in the industry today is physics-based models, and several commercial simulators exist (Amin, 2015). In later years, data-driven VFM models have demonstrated high performance (T. A. AL-Qutami, Ibrahim, and Ismail, 2017; T. A. AL-Qutami, Ibrahim, Ismail, and Ishak, 2017a; T. A. AL-Qutami, Ibrahim, Ismail, and Ishak, 2017b; T. AL-Qutami et al., 2018; T. Bikmukhametov and J. Jäschke, 2019; Grimstad et al., 2021). On the other hand, due to the inherently complex multiphase flow rate characteristics and that the available data typically resides in the small data regime (Grimstad et al., 2021), gray-box VFMs have gained increasing attention, see (Timur Bikmukhametov and Johannes Jäschke, 2020; Hotvedt, Grimstad, and Imsland, 2020; Hotvedt, Grimstad, and Imsland, 2021; Hotvedt, Grimstad, and Imsland, 2021; Hotvedt, Grimstad, Ljungquist, et al., 2022) and references therein. However, superior performance over physics-based or data-driven models has yet to be demonstrated. This article contributes in this direction by investigating four scenarios where a gray-box approach is believed to excel over non-gray-box alternatives. These are formulated as four hypotheses:

Hypothesis 1 Under mismatch between a physics-based VFM and the process, a gray-box VFM developed from the physics-based VFM achieves higher performance.

Hypothesis 2 With little available data, a gray-box VFM obtains higher performance than a data-driven VFM.

Hypothesis 3 Increasing the noise level on the data, a gray-box VFM is less influenced than a data-driven VFM.

Hypothesis 4 In nonstationary conditions, a gray-box VFM yields higher performance than a data-driven VFM.

In Hypothesis 1, the increased capacity of the gray-box compared to the physics-based model is believed to be significant. In Hypothesis 1-3, the decreased capacity of the gray-box compared to the data-driven model is believed to be decisive. In real life, available process data can have several uncontrolled characteristics, for instance, faulty sensor measurements. Such characteristics make it challenging to examine and conclude on the hypotheses as it is difficult to deduce whether a poor model performance results from the modeling technique or the available data. This has been experienced in previous work with gray-box VFMs (Hotvedt, Grimstad, Ljungquist, et al., 2022). Therefore, in this work, synthetic data designed to explore the hypotheses are generated by a simulator of a petroleum production choke. In several idealized experiments, the properties of gray-box production choke models are compared to physics-based and data-driven models. Hopefully, the results obtained can act as a guide to when gray-box modeling is likely to be advantageous, also in practical applications.

2 The simulator

The simulator is a physics-based petroleum production choke valve model. A typical production choke along with available measurements is illustrated in Fig. 1. The multiphase mass flow rate (a mixture of oil, gas, and water) $\dot{m}$ through the choke restriction is calculated using an advanced version of the Sachdeva model (Sachdeva et al., 1986), where slip effects, allowing the gas and liquid phases to move with unequal velocity, are included in the model. The slip model is taken from (Al-Safran and Kelkar, 2009). The model requires measurements of the pressure upstream (p_1) and downstream (p_2) of the choke valve, the upstream temperature (T_1), the choke opening (u), and the mass fractions of the phasic fluids $\boldsymbol{\eta} = (\eta_{\text{oil}}, \eta_{\text{gas}}, \eta_{\text{wat}})$. The mass fractions are assumed to sum to one. The volumetric multiphase flow rate $q = q_{\text{oil}} + q_{\text{gas}} + q_{\text{wat}}$ can be obtained from the $\dot{m}$ using the $\boldsymbol{\eta}$ and fluid densities ρ at

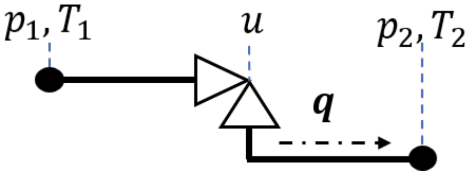


Figure 1: Illustration of the production choke valve and typically available measurements.

standard conditions (SC) (International Organization for Standardization, 1996):

$$q_i = \frac{\eta_i \dot{m}}{\rho_{i,SC}}, \quad i \in \{\text{oil}, \text{gas}, \text{wat}\}. \tag{1}$$

In the simulator, an area function relates the choke opening to the effective flow area through the choke $A(u)$. This function will mimic an equal percentage valve, where an equal increment in u results in an equal percentage changed area. The simulator, or process, is referred to as $\mathcal{P}$ and defined by the notation:

$$y = f(\boldsymbol{x}; \boldsymbol{\phi}) + \varepsilon \in \mathbb{R}, \tag{2}$$

where the model output is $y = q$, f is the first principle equations, the input measurements are $\boldsymbol{x} = (p_1, p_2, T_1, u, \eta_{\text{oil}}, \eta_{\text{wat}}) \in \mathbb{R}^6$, and $\boldsymbol{\phi}$ are constant model parameters. Noise is added to q by sampling ε from a probability distribution, for instance, a Gaussian distribution.

3 Dataset generation

Process $\mathcal{P}$ in Section 2 is used to generate three different datasets $\mathcal{D}_k = \{(\boldsymbol{x}_t, y_t)\}_{t=1}^{N_k}$, $k = \{1, 2, 3\}$. The index t reflects time. The datasets are designed to investigate the hypotheses in Section 1. The sequence of observations in each dataset is sampled from the joint probability distribution of $\mathcal{P}$: $p_t(\boldsymbol{x}, y) = p_t(y \,|\, \boldsymbol{x}) p_t(\boldsymbol{x})$, where $p_t(\boldsymbol{x})$ is the marginal distribution of the inputs and the output y_t follow the conditional distribution $p_t(y \,|\, \boldsymbol{x})$ expressed with (2). Notice, $\mathcal{P}$ is allowed to be nonstationary resulting in $p_{t_1}(\boldsymbol{x}, y) \neq p_{t_2}(\boldsymbol{x}, y)$ for $t_1 \neq t_2$.

Dataset $\mathcal{D}_1$ is generated as a best-case scenario to fairly examine Hypothesis 1-3 in Section 1. Firstly, the process is assumed stationary: $p_{t_1}(\boldsymbol{x}, y) = p_{t_2}(\boldsymbol{x}, y) \forall t$. Secondly, the $\boldsymbol{x}$ are independently drawn. This is idealized as measurements in real data are often strongly correlated (Hotvedt, Grimstad, Ljungquist, et al., 2022). Thirdly, a large range of common process conditions through the lifetime of a petroleum well is covered by sampling the inputs from:

$$\begin{aligned} p_1 &\sim \mathcal{U}(30, 70)\ bar, \\ p_2 &\sim \mathcal{N}(22, 0.5)\ bar, \\ T_1 &\sim \mathcal{N}(50, 2)\ {}^\circ C \\ u &\sim \mathcal{U}(0, 100)\ \%, \\ \eta_{\text{oil}} &\sim \mathcal{U}(0, 80)\ \%, \\ \eta_{\text{wat}} &\sim \mathcal{U}(0, 20)\ \%. \end{aligned} \tag{3}$$

for any t. The p_1, u, η_{oil}, and η_{wat} are sampled from wide uniform distributions as they commonly vary much, whereas p_2 and T_1 vary little, which is mimicked by drawing from narrow normal distributions. To ensure a sufficient dataset size $N_1 = 10000$

observations are sampled. Lastly, only normally distributed noise $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ is considered. The included noise levels are $\sigma_\varepsilon \in \{1, 2, 3, 4, 5, 10\}$, yielding a coefficient of variation of $\sigma_\varepsilon / \mu \in \{0.02, 0.05, 0.07, 0.1, 0.12, 0.24\}$, where μ is the mean of the noise-free flow rate measurements. Normally distributed noise is an idealized case as measurement sensors may comprise different noise types. However, it is interesting to investigate how the models are influenced by increasing level of idealized noise before introducing noise of higher complexity. The dataset is randomly separated into a training and a test dataset with $N_{1,\text{test}} = 2000$. From the training dataset, 20% are randomly extracted as a validation dataset.

The $\mathcal{D}_2$ and $\mathcal{D}_3$ mimics two typical real case scenarios where the process is nonstationary. In this study, only virtual drift is simulated, meaning that nonstationarity is caused by the marginal distribution $p_t(\boldsymbol{x})$ shifting in time while the conditional distribution $p_t(y \mid \boldsymbol{x})$ stays constant (Ditzler et al., 2015). Virtual drift is commonly seen for a petroleum asset. For instance, in time with the reservoir being depleted, the pressure in the reservoir and the upstream part of the choke decreases. If the petroleum asset is producing on plateau, process engineers increase the choke opening to maintain high production rates (Jahn, Cook, and Graham, 2008). Real drift, which is the opposite of virtual drift, is typically a consequence of substantial mechanical wear of equipment with time. It is believed that real drift is less prominent than virtual drift in a petroleum asset and is the reason why real drift is not simulated in this study. In both datasets, $N_2 = N_3 = 5000$ noise-free observations are sampled. The datasets are split into training and test according to time with $N_{2,\text{test}} = N_{3,\text{test}} = 2000$. Hence, the models will be used to predict future process responses. The validation dataset consists of the 600 latter training observations ordered by time. Dataset $\mathcal{D}_2$ mimics the depleting reservoir as described above. This scenario is illustrated in Figure 2. The p_1 is decreased in time using an exponential function, whereas the choke opening is increased in steps of 2.5%. The remaining variables are kept constant for any t: $p_2 = 22$ bar, $T_1 = 50°C$, $\eta_{\text{oil}} = 85\%$, and $\eta_{\text{wat}} = 2\%$. Dataset $\mathcal{D}_3$ mimics a scenario where the gas-to-oil ratio (GOR) increases with time. This phenomenon typically occurs when the reservoir pressure drops below the bubble point pressure such that the gas dissolved in the oil starts to escape (Jahn, Cook, and Graham, 2008). Fig. 3 illustrates the resulting flow rate q and the mass fractions of oil η_{oil} (green) and gas η_{gas} (orange) when the GOR is linearly increased from 200 to 1000. The p_1 is the same as for $\mathcal{D}_2$ illustrated in Fig. 2. The remaining variables are kept constant for any t: $p_2 = 22$ bar, $T_1 = 50°C$, $u = 100\%$, and $\eta_{\text{wat}} = 2\%$.

4 Models

Five production choke models have been developed: two physics-based, one data-driven, and two gray-box models. The models will be described briefly below. More details can be found in Hotvedt, Grimstad, Ljungquist, et al. (2022). The first physics-based model is the Sachdeva model, referred to as M, and defined by the short notation

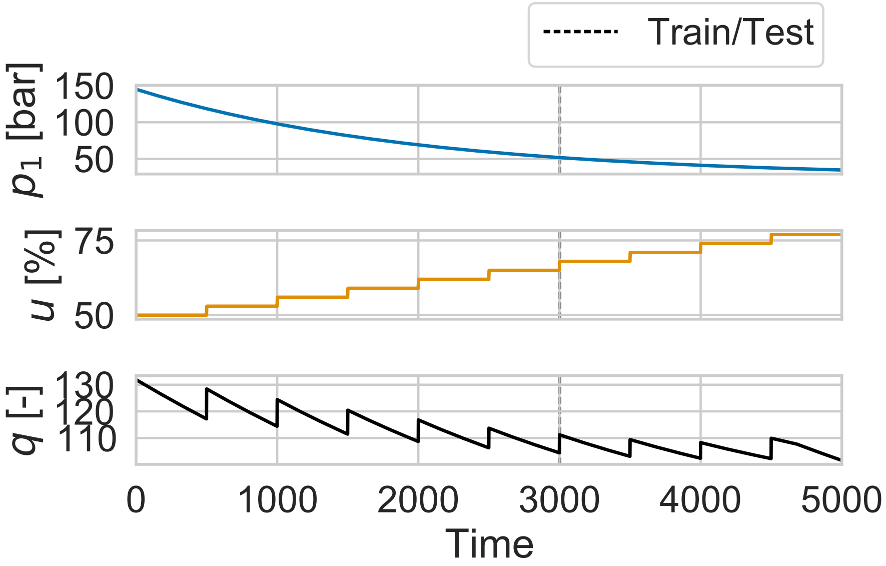


Figure 2: Illustration of the dataset mimicking typical behavior when the reservoir is depleted with time.

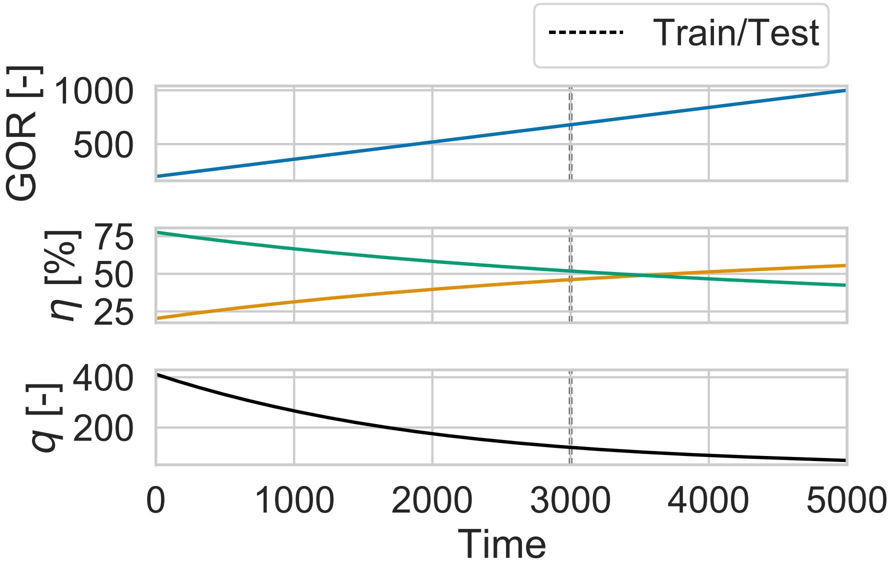


Figure 3: Illustration of the dataset mimicking typical behavior when the gas-to-oil ratio increases. The mass fractions of oil and gas are the green and orange curve, respectively.

$$\hat{y}_{\text{M}} = f_{\text{M}}(\boldsymbol{x}; \boldsymbol{\phi}_{\text{M}}) \in \mathbb{R}, \tag{4}$$

The true area function is kept unknown, and a linear relationship is utilized instead. Among the $\boldsymbol{\phi}_{\text{M}}$ is the discharge coefficient, which is a multiplicative calibration factor used to change the magnitude of the area function. In industrial VFMs, additional calibration factors exist to change the shape of the function. Here, these are excluded to restrict the capacity of M, enforcing a significant mismatch between $\mathcal{P}$ and M.

The second physics-based model is the advanced Sachdeva model used for $\mathcal{P}$, described in Section 2, referred to as M*. That is, the physical equations of the model are equal to the simulator, and the true area function is known. However, the true values of $\boldsymbol{\phi}$ in $\mathcal{P}$ are kept unknown from M* and $\boldsymbol{\phi}_{\text{M}^\star}$ must be estimated from data. M* is defined by

$$\hat{y}_{\text{M}^\star} = f(\boldsymbol{x}; \boldsymbol{\phi}_{\text{M}^\star}) \in \mathbb{R}. \tag{5}$$

Hence, any process-model mismatch will be a consequence of parameter deviation away from the true values and not structural mismatches as for the M.

The data-driven model is a fully connected, feed-forward neural network and is selected due to its large capacity. The model D is defined by

$$\hat{y}_{\text{D}} = f_{\text{D}}(\boldsymbol{x}; \boldsymbol{\phi}_{\text{D}}) \in \mathbb{R}, \tag{6}$$

where $\boldsymbol{\phi}_{\text{D}} = \{(\boldsymbol{W}_1, b_1), \dots (\boldsymbol{W}_L, b_L)\}$ are the weights and biases in the neural network on each layer $l = 1, ..., L$. The rectified linear unit is used as activation function.

The two different gray-box models are based on the M. The first is an error model where a data-driven model attempts to capture additive mismatches between $\mathcal{P}$ and M. This model is referred to as H-E:

$$\begin{aligned} \hat{y}_{\text{H-E}} &= f_{\text{H-E}}(\boldsymbol{x}; \boldsymbol{\phi}_{\text{H-E}}) \\ &= f_{\text{M}}(\boldsymbol{x}; \boldsymbol{\phi}_{\text{M}}) + f_{\text{D}}(\boldsymbol{x}; \boldsymbol{\phi}_{\text{D}}) \in \mathbb{R}. \end{aligned} \tag{7}$$

The second hybrid model addresses the unknown area function of $\mathcal{P}$ by multiplying the initial linear function of the M with a neural network: $A = A_{\text{M}} \times A_{\text{D}}$. Hence, both the magnitude and shape of the area function may be adjusted. This model is referred to as H-A:

$$\begin{aligned} \hat{y}_{\text{H-A}} &= f_{\text{H-A}}(\boldsymbol{x}; \boldsymbol{\phi}_{\text{H-A}}) = f_{\text{M}}(\boldsymbol{x}, A_{\text{D}}; \boldsymbol{\phi}_{\text{M}}) \in \mathbb{R} \\ A_{\text{D}} &= f_{\text{D}}(\boldsymbol{x}; \boldsymbol{\phi}_{\text{D}}) \in \mathbb{R}. \end{aligned} \tag{8}$$

As the neural network in H-A is multiplied with a small value (A_{M}), the capacity of the H-A is likely smaller than the capacity of H-E. This can be argued by acknowledging that large outputs from the network in H-A will be less influential on the flow rate predictions than a large output from the network in H-E.

For all models $i \in \{\mathrm{M}^{\star}, \mathrm{M}, \text{H-A}, \text{H-E}, \mathrm{D}\}$, the parameters are estimated using maximum a posteriori (MAP) estimation:

$$\begin{aligned} \hat{\boldsymbol{\phi}}_i &= \arg\max_{\boldsymbol{\phi}} p(\boldsymbol{\phi}_i \,|\, \mathcal{D}_k) \\ &= \arg\min_{\boldsymbol{\phi}} \Big[\sum_{t=1}^{N_k} \frac{1}{\sigma_{\varepsilon}^2} \left(y_t - \hat{y}_{i,t}\right)^2 \\ &+ \sum_{j=1}^{m} \frac{1}{\sigma_{i,j}^2} \left(\phi_{i,j} - \mu_{i,j} i\right)^2 \Big]. \end{aligned} \tag{9}$$

where m is the number of parameters. The priors on the parameters are assumed normal $\phi_{i,j} \sim \mathcal{N}(\mu_{i,j}, \sigma_{i,j}^2)$. The optimization problem is solved using stochastic, iterative, gradient-based optimization with the optimizer Adam (Kingma and Ba, 2015) and early stopping. Details of the training algorithm are given in Hotvedt, Grimstad, Ljungquist, et al. (2022).

5 Case study

Four experiments (Exp. 1-4) have been conducted to answer the four hypotheses in Section 1. Below, each experiment will be described, and the results visualized. Due to stochasticity, the experiments are run several times, called trials. The results of the trials will be visualized in figures with the median (p_{50}) as a solid line and a shaded area to indicate the lower (p_{25}) and upper (p_{75}) quantiles.

5.1 Exp. 1 - decreasing dataset size

5.1.1 Description

This experiment examines the performance of the models to a decreasing training dataset size. Dataset $\mathcal{D}_1$ is used for this purpose using the noise-free measurements. The considered training data lengths are $N \in \{2, 4, 8, 20, 40, 80, 800, 4000, 8000\}$. The training data is randomly extracted from $\mathcal{D}_1$ in each trial.

5.1.2 Results

The model performance in terms of the mean absolute error (MAE) is visualized as a function of N in Fig. 4.

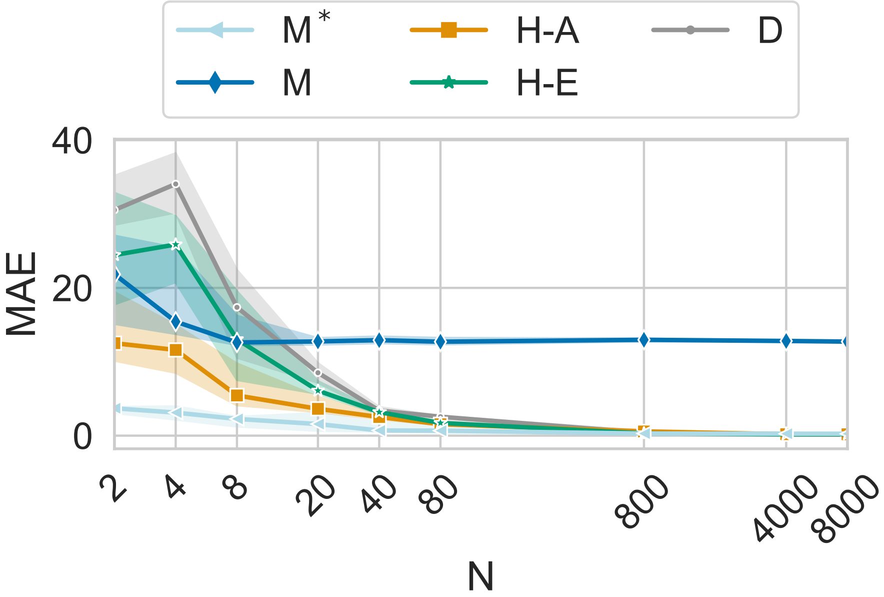


Figure 4: The mean absolute error as a function of the training set size.

5.2 Exp. 2 - increasing noise level

5.2.1 Description

This experiment investigates the robustness of the models to an increasing noise level. The models will be trained using dataset $\mathcal{D}_1$ and the output measurements with the different noise levels σ_ε.

5.2.2 Results

Fig. 5 shows the relative error as a function of the coefficient of variation σ_ε/μ. The relative error is calculated by dividing the MAE obtained at one noise level by the MAE obtained with noise-free measurements. The MAE is calculated using the noise-free q as the true value. A relative error larger than 1.0 means the model performance has decreased.

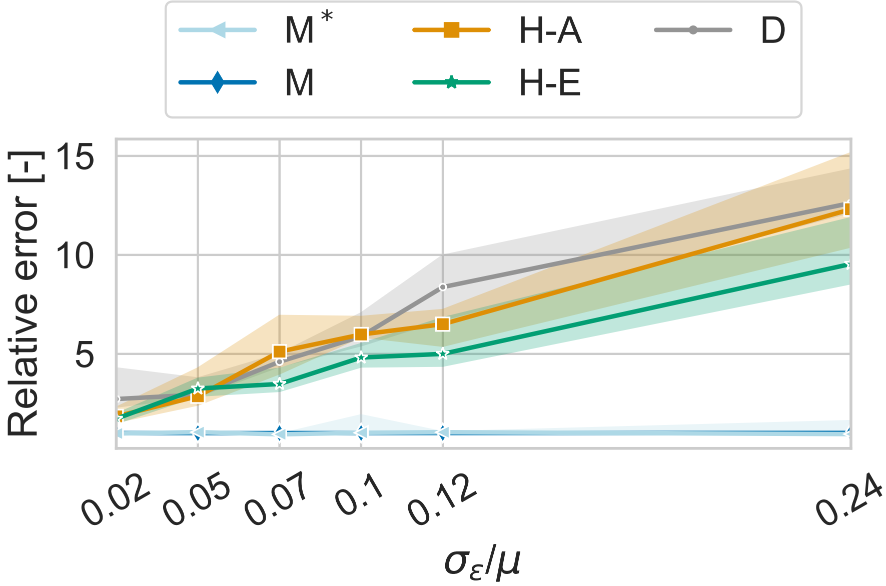


Figure 5: The relative error as a function of the coefficient of variation for the models.

5.3 Exp. 3 - the depleting reservoir

5.3.1 Description

Dataset $\mathcal{D}_2$ is used to analyze the model performances in the nonstationary case of a depleting reservoir.

5.3.2 Results

The absolute value of the prediction error (AE) in time is visualized for the different models in Fig. 6. The black, dotted line separates training and test data. Table 1 gives the validation and test MAE for the models.

Table 1: The validation and test mean absolute error in Exp. 3.

	M*	M	H-A	H-E	D
MAE$_v$	0.1	18.8	2.2	1.3	2.5
MAE$_t$	1.0	24.7	4.3	2.5	2.8

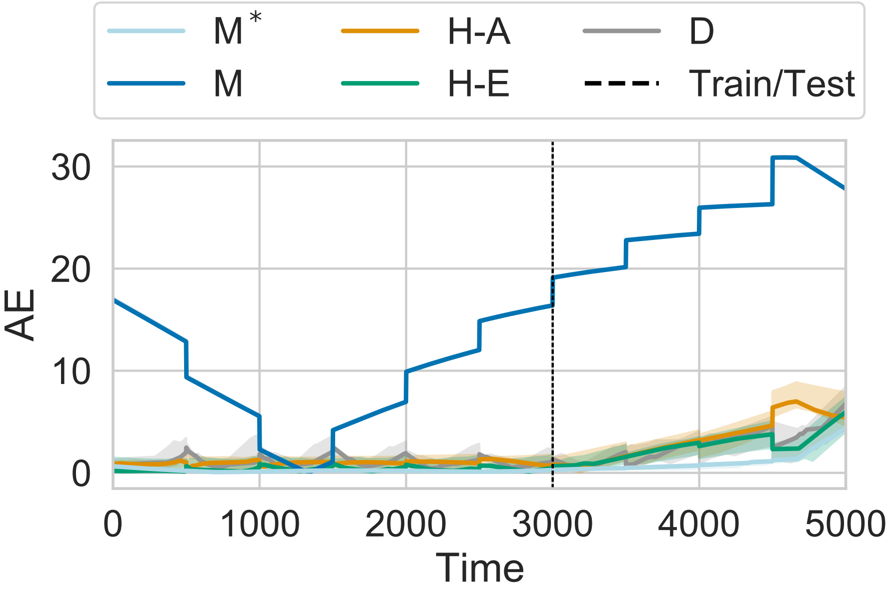


Figure 6: The absolute error of the model predictions as a function of time for Exp. 3.

5.4 Exp. 4 - increasing gas-to-oil ratio

5.4.1 Description

Dataset $\mathcal{D}_3$ is used to analyze the model performance in the nonstationary case of an increasing GOR.

5.4.2 Results

Fig. 7 shows the absolute error in time separated into training and test data. Table 2 gives the validation and test MAE.

Table 2: The validation and test mean absolute error in Exp. 4.

	M*	M	H-A	H-E	D
MAE$_v$	0.2	2.0	3.0	4.9	6.9
MAE$_t$	0.3	1.6	3.4	9.0	12.7

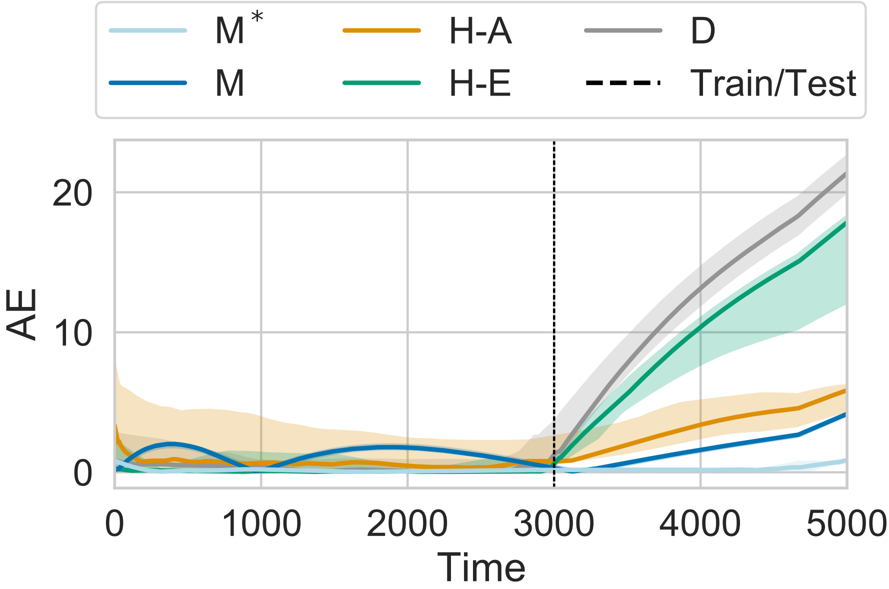


Figure 7: The absolute error of the model predictions as a function of time in Exp. 4.

6 Discussion

Firstly, notice from Fig. 4 that for large dataset sizes, only M yields a high process-model mismatch. For $M^\star$, this was expected as there are no structural mismatches between $M^\star$ and $\mathcal{P}$. For the other models, the negligible MAE indicates a sufficient capacity. Observe, only a few observations ($N > 80$) were required for the D and Hs to obtain negligible MAE, which suggests that the process is simple to learn. With real-life, complex processes, a higher number of observations would likely be required to remove the bias. Secondly, Fig. 4 shows that the error increases the most for the D when the dataset size decreases, followed by H-E and H-A. This implies that the D has the largest variance, followed by H-E and H-A, and adapts the most to the training data, thus, decreasing the generalizability to the unobserved test data. Fig. 5 shows that the M and $M^\star$ are robust against an increasing noise level, whereas the Hs and D are not. This confirms that the Hs and D have a larger variance. On the other hand, Fig. 5 shows that the Hs barely achieve a better performance than the D. Moreover, it seems that the H-E has a lower variance than the H-A, which is conflicting with the results in Fig. 4. However, H-E is designed to capture additive mismatches, which is the only considered noise influence and may explain the slightly better performance.

The results from Exp. 1-2 indicates that gray-box models may yield lower variance than a data-driven model and reduce bias in physics-based models with structural process-model mismatches. Therefore, in nonstationary conditions, the expectation is that the Hs will perform better than the D and the M. Figs. 6-7 and Tables 1-2 do show that at least one H performs better than the D in both experiments and that it is advantageous with an H when the process-model mismatch of the M is large as in Exp. 3. The large mismatch in Exp. 3 is a consequence of the available measurements of u making the assumed linear shape of the area function in M of greater influence than in Exp. 4 where $u = 100\% \; \forall t$. It should be noted, the U-shaped curve of the M on the training data in Fig. 6 is due to the objective function in (9), and the performance on the test data can likely be improved by weighing the recent observations the most. On the other hand, in Exp. 3, the performance of the D is comparable with the Hs. In Exp. 4, the discrepancy in performance between the Hs is large, where the H-A and H-E yield good and poor performance, respectively. Ideally, the best model could be deduced a priori to training by examining known process-model mismatches and the capacity of the models. Nevertheless, this showed nontrivial even for these idealized experiments. For instance, in Exp. 3, the H-A was expected to perform best as it targets the discrepancy between the linear and true area function. Nevertheless, H-E yields the best performance, closely followed by the D. Therefore, model selection must be performed posterior to training using the performance on the validation dataset. Accordingly, the importance of extracting the validation dataset representatively increases, for instance, by time for nonstationary processes. Positively, the results in Tables 1-2 indicate that the errors on the validation data are illustrative for the model performances on the test data as the best model yields the lowest error in both experiments. A disadvantage is the increase of overhead on model development and testing. The observant reader notices that the model performances in Figs. 6-7 decrease with time. This is a typical scenario for steady-state modeling in nonstationary conditions. Utilization of learning methods for frequent model updating would likely improve the long-term performances. Such approaches could also handle the existence of both virtual and real drift.

7 Concluding remarks

Overall, the results in this research show that a gray-box approach to VFM can reduce both model bias and variance compared to a physics-based and data-driven approach, respectively. From the results and discussions, Hypotheses 1 and 2 from Section 1 are confirmed. However, the gray-box and data-driven models have comparable performances for an increasing data noise level and Hypothesis 3 cannot be confirmed. The results from experiments in nonstationary conditions showed that a gray-box model can improve the performance of a data-driven model, hence, confirming Hypothesis 4. Moreover, the gray-box model can significantly improve the performance of a physics-based model under large process-model mismatches. On the other hand, the results also show that it is challenging to determine prior to model training which model yields the best performance in different scenarios, and overhead on model development and testing is unavoidable.

Certainly, the hypotheses were only investigated on synthetic data and generalization to real life is challenging. In real life, there may be other undesired and unknown characteristics of the process complicating model development, for instance, increasingly complex and rare physical phenomena or heteroscedastic measurement noise. Moreover, this work only considers two scenarios of nonstationary process behavior, although possible scenarios are numerous. Additionally, other gray-box model variants may yield different results in different scenarios. Nevertheless, the results from this work indicate that gray-box modeling is advantageous for virtual flow metering in certain scenarios and can hopefully act as a guide in modeling real processes.

Acknowledgement

This research is a part of BRU21 - NTNU Research and Innovation Program on Digital and Automation Solutions for the Oil and Gas Industry (`www.ntnu.edu/bru21`) and supported by Lundin Energy Norway.

References

Amin, A. (2015). "Evaluation of Commercially Available Virtual Flow Meters (VFMs)". In: *Proceedings of the Annual Offshore Technology Conference*, pp. 1293–1318.

Bikmukhametov, T. and J. Jäschke (2019). "Oil Production Monitoring using Gradient Boosting Machine Learning Algorithm." In: *IFAC-PapersOnLine* 52 (1), pp. 514–519.

Bikmukhametov, Timur and Johannes Jäschke (2020). "Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models". In: *Computers and Chemical Engineering* 138.

Ditzler, G. et al. (2015). "Learning in Nonstationary Environments: A survey". In: *IEEE Computational Intelligence Magazine*, pp. 12–25.

Grimstad, B. et al. (2021). "Bayesian Neural Networks for Virtual Flow Metering: An Empirical Study". In: *Applied Soft Computing* 112.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning.* New York, USA: Springer.

Hotvedt, M., B. Grimstad, and L. Imsland (2020). "Developing a Hybrid Data-Driven, Mechanistic Virtual Flow Meter - a Case Study". In: *IFAC-PapersOnLine* 53 (2), pp. 11692–11697.

— (2021). "Identifiability and interpretability of hybrid, gray-box models". In: *IFAC-PapersOnLine* 54 (3), pp. 389–394.

Hotvedt, M., B. Grimstad, D. Ljungquist, et al. (2022). "On gray-box modeling for virtual flow metering". In: *Control Engineering Practice* 118.

International Organization for Standardization (Dec. 1996). *Natural gas - Standard reference conditions.* Standard. International Organization for Standardization.

Jahn, F., M. Cook, and M. Graham (2008). *Hydrocarbon exploration and production, 2nd edition.* Elsevier.

Kingma, Diederik P. and Jimmy Lei Ba (2015). "Adam: A method for stochastic optimization". In: *3rd International Conference on Learning Representations (ICLR)*, pp. 1–15. eprint: `1412.6980`.

Oerter, R. (2006). *The Theory of Almost Everything: The Standard Model, the Unsung Triumph of Modern Physics.* Pi Press.

AL-Qutami, T.A et al. (2018). "Virtual multiphase flow metering using diverse neural network ensemble and adaptive simulated annealing". In: *Expert Syst. Appl.* 93, pp. 72–85.

AL-Qutami, Tareq Aziz, Rosdiazli Ibrahim, and Idris Ismail (2017). "Hybrid neural network and regression tree ensemble pruned by simulated annealing for virtual flow metering application." In: *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 304–309.

AL-Qutami, Tareq Aziz, Rosdiazli Ibrahim, Idris Ismail, and Mohd Azmin Ishak (2017a). "Development of soft sensor to estimate multiphase flow rates using neural networks and early stopping". In: *Int. J. Smart Sens. Intell. Syst.* Vol. 10, pp. 199–222.

— (2017b). “Radial basis function network to predict gas flow rate in multiphase flow.” In: *Proceedings of the 9th International Conference on Machine Learning and Computing*, pp. 141–146.

Roscher, R. et al. (2020). “Explainable Machine Learning for Scientific Insights and Discoveries.” In: *arXiv:1905.08883v3*, pp. 1–29.

Sachdeva, R. et al. (1986). “Two-phase flow through chokes”. In: *Society of Petroleum Engineers, Annual Technical Conference and Exhibition.*

Al-Safran, E. M. and M. Kelkar (2009). “Predictions of two-phase critical-flow boundary and mass-flow rate across chokes”. In: *Society of Petroleum Engineers* 24 (2).

Solle, D. et al. (2016). “Between the Poles of Data-Driven and Mechanistic Modeling for Process Operation”. In: *Chemie Ingenieur Technik.*

Toskey, E.D. (2012). “Improvements to Deepwater Subsea Measurements RPSEA Program: Evaluation of Flow Modeling”. In: *Offshore Technology Conference.*

Willard, J. et al. (2020). “Integrating Physics-Based Modeling With Machine Learning: A Survey”. In: *arXiv:2003.04919v4*, pp. 1–34.

F Passive learning to address nonstationarity in virtual flow metering applications

Preprint of Hotvedt, Grimstad, and Imsland (2022)

M. Hotvedt, B. Grimstad, and L. Imsland (2022). "Passive learning to address nonstationarity in virtual flow metering applications". In: *Submitted to Expert Systems with Application for possible publication*

Passive learning to address nonstationarity in virtual flow metering applications

M. Hotvedt[1], B. Grimstad[1, 2], and L. Imsland[1]

[1]Engineering Cybernetics Department, NTNU, Trondheim, Norway
[2]Solution Seeker

Abstract: Steady-state process models are common in virtual flow meter applications due to low computational complexity, and low model development and maintenance cost. Nevertheless, the prediction performance of steady-state models typically degrades with time due to the inherent nonstationarity of the underlying process being modeled. Few studies have investigated how learning methods can be applied to sustain the prediction accuracy of steady-state virtual flow meters. This paper explores passive learning, where the model is frequently calibrated to new data, as a way to address nonstationarity and improve long-term performance. An advantage with passive learning is that it is compatible with models used in the industry. Two passive learning methods, periodic batch learning and online learning, are applied with varying calibration frequency to train virtual flow meters. Six different model types, ranging from data-driven to first-principles, are trained on historical production data from 10 petroleum wells. The results are two-fold: first, in the presence of frequently arriving measurements, frequent model updating sustains an excellent prediction performance over time; second, in the presence of intermittent and infrequently arriving measurements, frequent updating in addition to the utilization of expert knowledge is essential to increase the performance accuracy. The investigation may be of interest to experts developing soft-sensors for nonstationary processes, such as virtual flow meters.

Keywords: virtual flow metering, nonstationarity, passive learning, online learning, periodic batch learning, neural networks

1 Introduction

Many real-world, physical processes are nonstationary (Sayed-Mouchaweh and Lughofer, 2012). To various degrees, process conditions and properties change with time. Nevertheless, a common assumption in process modeling is time independence, leading to stationary, or steady-state, models (Granero-Belinchón, Roux, and Garnier, 2019). Several arguments militate for the utilization of steady-state models. Firstly, many processes are slowly time-varying making the stationary assumption reasonable for short-term applications. Secondly, steady-state models typically reduce the cost of model development and maintenance (Solle et al., 2016). Thirdly, these models are often less computationally heavy, which can increase the suitability in real-time control and optimization applications (Gravdahl and Egeland, 2002). On the other hand, the performance of steady-state models in nonstationary conditions typically degrade with time and necessitates algorithms that improve the handling of nonstationarity.

Virtual flow metering (VFM) is a soft-sensor technology that utilizes process models for continuous prediction of the multiphase flow rate at key locations in a petroleum asset (Toskey, 2012). In Figure 1, a simplified illustration of the production system for one petroleum well is given along with typically available sensor measurements for well-equipped wells. A multiphase flow meter (MPFM) measures the phasic flow rates, $\boldsymbol{q} = [q_{\text{gas}}, q_{\text{oil}}, q_{\text{water}}]$, through the production choke valve. Under well-testing, the phasic flow rates can be measured using the test separator. The total multiphase flow rate through the production system is $Q = q_{\text{gas}} + q_{\text{oil}} + q_{\text{water}}$. A typical application of VFM is as a back-up to the MPFM in case of failure (Varyan, Haug, and Fonnes, 2015).

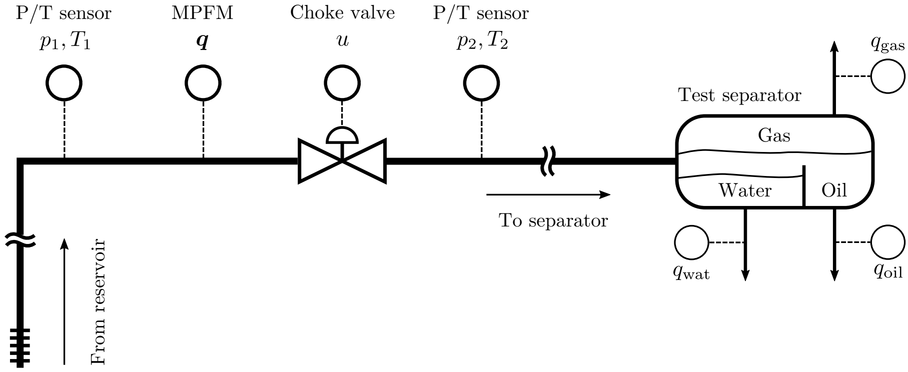


Figure 1: A simplified illustration of the petroleum production system with typical sensor placements. A multiphase flow meter (MPFM) measures the phasic flow rates through the choke valve. Measurements of the phasic flow rates can also be obtained when the well is tested, using, for instance, a test separator.

The underlying process of the VFM comprises the reservoir, wells, pipelines, and processing facility. This process is nonstationary with time-varying process conditions and properties (Guo, Lyons, and Ghalambor, 2007). The multiphase flow rate through the production system has a dynamic nature with both fast and slow transients. Fast transients occur with control changes, which induce pressure waves through the system, such as the opening of the choke valve (Jansen, 2015). These are in the time range of minutes to hours. Slow transients are caused by the reservoir being depleted with time, which in turn results in a pressure declination in the production system and a decreased production flow rate (Foss, Knudsen, and B. Grimstad, 2018). These occur in a time range of months to years, dependent on the size of the reservoir. Furthermore, as the petroleum asset ages, technologies such as artificial lift with gas or water are applied to improve production. Other sporadic changes such as maintenance tasks will also induce transient process behavior. Hence, the natural approach to VFM is nonstationary models. Several commercial VFMs such as Olga and LedaFlow are nonstationary (Amin, 2015), and other examples exist in literature (Holmås and Løvli, 2011; Jordanou et al., 2017). On the other hand, due to the slow dynamics of the reservoir, steady-state reservoir conditions for a certain time interval can often be assumed (Shippen, 2012). Furthermore, considering the inherent complex multiphase flow characteristics, which make it challenging to develop and solve nonstationary VFMs, steady-state VFMs are the most common approach in literature (T. Bikmukhametov and J. Jäschke, 2019), both for physics-based models (Shippen, 2012; Varyan, Haug, and Fonnes, 2015) and machine learning (ML) models (T. A. AL-Qutami, Ibrahim, and Ismail, 2017; T. A. AL-Qutami, Ibrahim, Ismail, and Ishak, 2017a; T. A. AL-Qutami, Ibrahim, Ismail, and Ishak, 2017b; T. AL-Qutami et al., 2018; Timur Bikmukhametov and Johannes Jäschke, 2020; B. Grimstad, Hotvedt, et al., 2021). Nevertheless, studies show that steady-state VFM models should be updated or recalibrated in time to provide adequate long-term prediction accuracy (Sandnes, Bjarne Grimstad, and Kolbjørnsen, 2021; Hotvedt, B. Grimstad, Ljungquist, et al., 2022). Several model learning methods exist that attempt to account for nonstationarity without imposing temporal dependencies in the model. The learning methods can be divided into an active or passive method (Ditzler et al., 2015). In passive learning, the process is assumed to be continuously changing and the model is routinely updated with access to new measurements. In active learning, statistical tests are used to detect significant changes in the process conditions, whereupon model updating is initiated.

For the VFM application, it is not uncommon that new observations arrive infrequently, for example, twice a year or at the most once per month under well-testing (Monteiro et al., 2020). In such an event, active learning is redundant as the process conditions and properties are likely to have changed significantly during the elapsed time, and the model should be updated with each new measurement. For assets with access to continuous flow rate measurements, such as MPFM measurements, the VFM models would likely benefit from updating using these measurements in between well-tests. Nevertheless, in industry, even with frequent access to new measurements, model learning can occur intermittently due to limited resources or manual, non-systematic workflows (Koroteev and Tekic, 2021).

To the authors' knowledge, no studies have investigated the influence of the update frequency on sustaining the prediction accuracy of steady-state VFM models over time, hence, obtaining a high long-term performance. This research contributes in this direction by examining two passive learning methods: periodic batch learning and online learning. Six VFM models are developed for the petroleum production choke valve in 10 petroleum wells on Edvard Grieg, an asset on the Norwegian Continental Shelf (Lundin Energy Norway, 2020). Real production data spanning five years are used in the development. The long-term predictive performance is expected to increase with the frequency of which the models are updated. The best performance is expected from online learning, for which the models are updated with every new measurement. For periodic batch learning, the performance is expected to drop as the frequency is lowered. The rest of the article is structured the following way: section 2 presents relevant theory for steady-state modeling of processes in nonstationary conditions. Thereafter, Section 3 describes the available data and the VFM model types. In Section 4, the numerical study examining the learning methods is described and results visualized and discussed. Lastly, Section 5 gives concluding remarks.

2 Steady-state modeling in nonstationary conditions

Consider a stream of observations $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_t, y_t), \ldots\}$, where $\boldsymbol{x}_t \in \mathbb{R}^d$ represents measured process conditions and $y_t \in \mathbb{R}$ a (dependent) target variable at time t. In general, the set S can be thought of as a realization of a stochastic process $\mathcal{P}$ governed by a generative model (G. Oliveira, Minku, and A. Oliveira, 2021)

$$p_t(\boldsymbol{x}, y) = p_t(y \mid \boldsymbol{x}) p_t(\boldsymbol{x}). \tag{1}$$

In (1), $p_t(\boldsymbol{x})$ is the marginal distribution of the process conditions, and $p_t(y \mid \boldsymbol{x})$ is the conditional distribution of the target, both at time t. The index t indicates that the distributions may be time-variant, and therefore $\mathcal{P}$ may be nonstationary.

In real-time applications of machine learning, like data-driven virtual flow metering, it is natural to develop models on historical data and test the model performance on future data. Collect in $\mathcal{D}_{a:b} = \{(\boldsymbol{x}_t, y_t)\}_{t=a}^{b}$ the sequence of observations with $t \in [a, b]$, and in $\mathcal{D}_a = \{(\boldsymbol{x}_t, y_t)\}_{t=a}$ the single observation at $t = a$. For a model to be developed at time $t = T$, the training dataset is denoted by $\mathcal{D}^{tr} = \mathcal{D}_{1:T}$ and the test dataset by $\mathcal{D}^{tr} = \mathcal{D}_{T+1:\infty}$.

Many machine learning models and algorithms are based upon the assumption that the training and test dataset originate from the same probability distribution; the data points in S are independent and identically distributed (i.i.d.) (Hastie, Tibshirani, and Friedman, 2009). When the stochastic process $\mathcal{P}$ in (1) is nonstationary, the i.i.d. assumption is invalidated, as a dataset shift can occur when moving from the training phase to the test phase. In the following, different types of dataset shifts are explored, and suitable learning methods to alleviate the effect of nonstationarity on predictive performance are discussed.

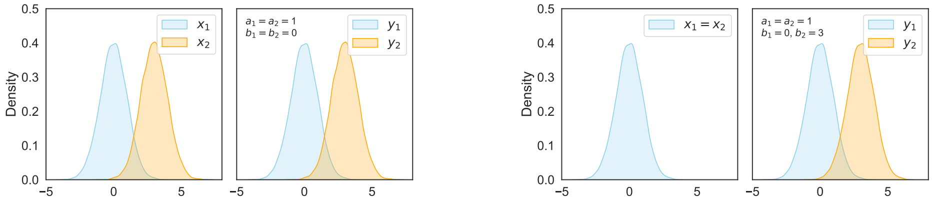
2.1 Dataset shifts

When $\mathcal{P}$ is nonstationary, the joint probability distribution can shift in time resulting in $p_t(\boldsymbol{x}, y) \neq p_{t+\tau}(\boldsymbol{x}, y)$ for an arbitrary lapse $\tau > 0$ in time. Using the model in (1) two types of dataset shifts, also called concept drifts, can occur in time: virtual and real drift[1] (Quiñonero-Candela et al., 2009; Ditzler et al., 2015). With virtual drift, the marginal distribution shifts in time. That is, $p_t(\boldsymbol{x}) \neq p_{t+\tau}(\boldsymbol{x})$ for $\tau > 0$. With real drift, the conditional distribution shifts in time, such that $p_t(y \mid \boldsymbol{x}) \neq p_{t+\tau}(y \mid \boldsymbol{x})$ for $\tau > 0$. Real and virtual drift may happen separately or simultaneously, in any case shifting the joint distribution with time. Notice, in (Quiñonero-Candela et al., 2009), several other specialized forms of dataset shifts are discussed.

As an example, consider a process with a conditional distribution

$$y_t = a_t x_t + b_t, \tag{2}$$

with parameters $\boldsymbol{\theta}_t = \{a_t, b_t\}$. Two subsequent time instances $t = 1$ and $t = 2$ are examined. The input at $t = 1$ is sampled from $p_1(x) \sim \mathcal{N}(0, 1)$. At $t = 2$, the mean changes such that $p_2(x) \sim \mathcal{N}(3, 1)$. If the model parameters remain unchanged, this is virtual drift, and the response in y changes only as a consequence of changes in the marginal distribution. The scenario is illustrated in Figure 2a. In another scenario, consider the input distribution to remain unchanged, but the b parameter of the model to change from $b_1 = 0$ to $b_2 = 3$. The parameter change causes the conditional distribution in (2) to change, thereby causing real drift, illustrated in Figure 2b. Notice, in the two scenarios, virtual and real drift cannot be distinguished by analyzing y only.



(a) Virtual drift. The marginal distribution $p(x)$ changes from one time step to another causing a change $p(y \mid x)$. The parameters remain unchanged.

(b) Real drift. The marginal distribution $p(x)$ does not change but $p(y \mid x)$ changes as a consequence of the changed model parameters.

Figure 2: Dataset shifts illustrated with a) virtual drift and b) real drift.

Virtual drift is commonly seen in the VFM application. For example, in time with the reservoir being depleted the pressure through the production system decreases.

[1] Other naming conventions for virtual drift are virtual concept drift, covariate shift, or input drift. Real drift is also known as real concept drift or output drift.

At the same time, in the early life of a petroleum asset, the production engineers can often increase the choke openings to maintain a constant production rate, also called plateau production (Jansen, 2015). The VFM application can also experience real drift. Substantial mechanical wear of the equipment in the well can occur with time, for instance, due to sand production, and can result in a change in the flow rate even for unchanged process conditions. It is believed that virtual drift is the major cause of observed dataset shifts in VFMs. However, as Figure 2 illustrates, it can be difficult to separate between the two types of drifts.

The next section discusses the impact that dataset shifts can have on steady-state VFM models.

2.2 Parameter estimation of steady-state models

A common approach to steady-state modeling is to use an inductive method to learn an approximation of the conditional distribution $p_t(y \mid \boldsymbol{x})$ in (1) from a fixed set of steady-state observations $\mathcal{D}_{1:T}$. A typical form of the approximation is

$$\hat{y}_t = f_{\boldsymbol{\theta}}(\boldsymbol{x}_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2), \tag{3}$$

where $f_{\boldsymbol{\theta}}$ is a parametric model of the mean, with parameters $\boldsymbol{\theta}$, and ϵ_t is a homoscedastic noise term. The model in (3) is a steady-state model since $\hat{y}_t$ is conditioned on $\boldsymbol{x}_t$, and the parameters $\boldsymbol{\theta}$ and σ_ϵ are time-invariant. The i.i.d. assumption is thus used. Note that, the resulting model is steady-state even though the data used to learn the model originate from a nonstationary process.

Conditional models, like the steady-state model in (3), are commonly trained using maximum a posteriori (MAP) estimation. In MAP estimation, the mode of the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$ is maximized. Here, the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ is given by (3) and $p(\boldsymbol{\theta})$ is a prior on the $\boldsymbol{\theta}$ parameters. For a normal prior, $\theta_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, $i = 1, .., N_\theta$, the optimization problem can be expressed as follows:

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} \quad \log p(\mathcal{D} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \\ &= \arg\min_{\boldsymbol{\theta}} \quad \sum_{i=1}^{N} \frac{1}{\sigma_\epsilon^2} \left(y_i - \hat{y}_i\right)^2 + \sum_{i=1}^{N_\theta} \frac{1}{\sigma_i^2} \left(\theta_i - \mu_i\right)^2 . \end{aligned} \tag{4}$$

where N is the number of data points in the training dataset. From (4), it is seen that MAP estimation is a trade-off between minimizing the squared errors and parameter deviation away from its respective mean value μ_i. By multiplying the objective function by σ_ϵ^2/N, the equivalence of MAP estimation to the familiar minimization of mean squared error with ℓ_2-regularization is obtained (I. Goodfellow, Y. Bengio, and Courville, 2016).

In the machine learning domain, (4) is commonly optimized by first-order gradient descent methods (Bishop, 2006). These methods update the parameters iteratively according to the following scheme:

$$\hat{\boldsymbol{\theta}}^{(k+1)} = \hat{\boldsymbol{\theta}}^{(k)} - \gamma^{(k)} \mathcal{M}(\mathcal{B}, \hat{\boldsymbol{\theta}}^{(k)}),\ k = 1, .., E \tag{5}$$

where E is the number of iterations or steps taken towards the optimal value, γ is the learning rate or step-size, and $\mathcal{M}$ is the set of equations calculating the step direction. The $\mathcal{B}$ is a set of observations extracted from the training dataset and can be in the range of one to all observations. Any parameter that is not included in $\boldsymbol{\theta}$ is called a hyperparameter, for instance, γ, E, and $|\mathcal{B}|$.

The above approach to steady-state modeling is susceptible to dataset shifts since the estimate (optimum) in (4) likely will change with time, resulting in poor test performance. When applied to VFM, for which the data is generated by a nonstationary process, both virtual and real concept drift will negatively influence the long-term predictive performance. A VFM performance that diminishes with time, has been documented in several publications (B. Grimstad, Hotvedt, et al., 2021; Hotvedt, B. Grimstad, Ljungquist, et al., 2022; Sandnes, Bjarne Grimstad, and Kolbjørnsen, 2021). In the following section, passive learning methods are discussed. These methods can be used to account for dataset shifts in steady-state modeling.

2.3 Passive learning for steady-state models

In passive learning, the process $\mathcal{P}$ is assumed to be continuously changing with time, and model updating is routinely initiated regardless of whether or not dataset shifts occur. Two methods of passive learning are examined: online learning (OL) and periodic batch learning (PBL).

At time $t = T$, an initial parameter estimate is obtained from $\mathcal{D}^{\text{tr}} = \mathcal{D}_{1:T}$ using the approach in Section 2.2. The estimated parameters are referred to as $\hat{\boldsymbol{\theta}}_T$, and the resulting steady-state model is given by

$$\hat{y}_t = f_{\hat{\boldsymbol{\theta}}_T}(\boldsymbol{x}_t) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2). \tag{6}$$

From this point in time, the two learning methods can be applied. These are visualized in Figure 3 and are explained in the consecutive sections.

2.3.1 Periodic batch learning

In periodic batch learning, the model in (6) is used to make predictions for $\tau > 0$ time steps $\mathcal{D}^{\text{te}}_{T:\tau}$ before it is retrained at $t = T + \tau$. In retraining, the new parameters $\hat{\boldsymbol{\theta}}_{T+\tau}$ are estimated using all data observed at that time as training data $\mathcal{D}^{\text{tr}}_{1:T+\tau}$ and the approach in Section 2.2. The procedure is repeated with a period of τ, where the posterior parameter distribution can be described with

$$p(\boldsymbol{\theta} \,|\, \mathcal{D}_{1:T+\tau}) \propto p(\mathcal{D}_{1:T+\tau} \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta}), \tag{7}$$

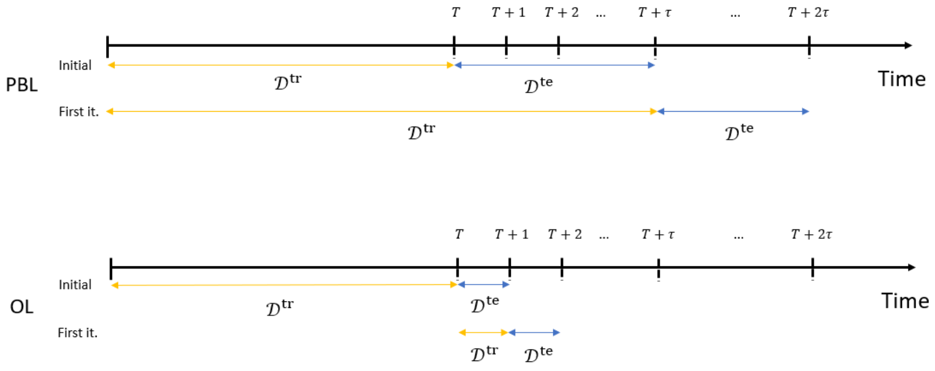

PBL
Initial
First it.
T
$T+1$
$T+2$
...
$T+\tau$
...
$T+2\tau$
Time
$\mathcal{D}^{\mathrm{tr}}$
$\mathcal{D}^{\mathrm{te}}$
$\mathcal{D}^{\mathrm{tr}}$
$\mathcal{D}^{\mathrm{te}}$
OL
Initial
First it.
T
$T+1$
$T+2$
...
$T+\tau$
...
$T+2\tau$
Time
$\mathcal{D}^{\mathrm{tr}}$
$\mathcal{D}^{\mathrm{te}}$
$\mathcal{D}^{\mathrm{tr}}$
$\mathcal{D}^{\mathrm{te}}$

Figure 3: One iteration of the periodic batch learning and online learning update procedure after obtaining the initial parameter estimate. $\mathcal{D}^{\mathrm{tr}}$ are training datasets used in the estimation problem and $\mathcal{D}^{\mathrm{te}}$ are test datasets used to test the predictive capabilities of the model.

An appropriate τ must be determined and can be accomplished by applying a change or shift detection algorithm offline on historical data. There exist much literature on shift detection algorithms, see for example Raza, Prasad, and Li (2015) and references therein. In this research, Hotelling's T-squared test for two multivariate, independent samples is used to investigate a null hypothesis stating that no virtual drift is present in the dataset. The algorithm for determining τ is described in A.

2.3.2 Online learning

In online learning, model updating occur for each new observation that arrives. However, the posterior distribution at the next time step is updated using only the current observation as the training data and the posterior distribution at the previous time step as the prior. For instance, at $t = T+1$:

$$p(\boldsymbol{\theta} \mid \mathcal{D}_{1:T+1}) \propto p(\mathcal{D}_{T+1} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}_{1:T}). \tag{8}$$

Mathematically, (8) can be derived as follows. With the approach in Section 2.2, the posterior parameter distribution at $t = T+1$ is given by

$$p(\boldsymbol{\theta} \mid \mathcal{D}_{1:T+1}) = \frac{p(\mathcal{D}_{1:T+1} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D}_{1:T+1})}, \tag{9}$$

where $p(\mathcal{D}_{1:T+1})$ is the proportionality constant in Bayes' law. Applying the i.i.d. assumption, the likelihood function of the model and the evidence can be written as

$$\begin{aligned} p(\mathcal{D}_{1:T+1} \mid \boldsymbol{\theta}) &= \prod_{t=1}^{T+1} p(\mathcal{D}_t \mid \boldsymbol{\theta}) = p(\mathcal{D}_{1:T} \mid \boldsymbol{\theta}) p(\mathcal{D}_{T+1} \mid \boldsymbol{\theta}) \\ p(\mathcal{D}_{1:T+1}) &= \prod_{t=1}^{T+1} p(\mathcal{D}_t) = p(\mathcal{D}_{1:T}) p(\mathcal{D}_{T+1}), \end{aligned} \tag{10}$$

respectively. Note that, while the i.i.d. assumption is likely false for a nonstationary process, it is already used in steady-state modeling. Inserting (10) in (9), the posterior parameter distribution at $t = T + 1$ can be written as

$$p(\boldsymbol{\theta} \mid \mathcal{D}_{1:T+1}) = \frac{p(\mathcal{D}_{T+1} \mid \boldsymbol{\theta})}{p(\mathcal{D}_{T+1})} \cdot \frac{p(\mathcal{D}_{1:T} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D}_{1:T})} = \frac{p(\mathcal{D}_{T+1} \mid \boldsymbol{\theta})}{p(\mathcal{D}_{T+1})} \cdot p(\boldsymbol{\theta} \mid \mathcal{D}_{1:T}) \tag{11}$$

and (8) is obtained.

An issue becomes apparent when deriving the MAP estimate for (8)

$$\hat{\boldsymbol{\theta}}_{T+1} = \arg\max_{\boldsymbol{\theta}} \Big[\log p(\mathcal{D}_{T+1} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta} \mid \mathcal{D}_{1:T}) \Big]. \tag{12}$$

Ideally, the parameter estimation in the previous time step should have provided both the mean and the variance of the updated posterior parameter distribution $p(\boldsymbol{\theta} \mid \mathcal{D}_{1:T}) \sim \mathcal{N}(\boldsymbol{\mu}_T, \boldsymbol{\Sigma}_T)$. However, MAP estimation gives point estimates of the mode only. When the likelihood and prior is normal, an estimate of the mean $\boldsymbol{\mu}_T = \hat{\boldsymbol{\theta}}_T$ is obtained since the mode and mean coincides, but $\boldsymbol{\Sigma}_T$ remains unknown. Therefore, the second term in (12) cannot be calculated if MAP estimation is used in each time step. As discussed in Section 2.2, this term is ℓ_2-regularization of the parameters. According to (I. Goodfellow, Y. Bengio, and Courville, 2016), for some cases, the algorithm early stopping has a similar effect as ℓ_2-regularization. For linear models, the solution obtained with early stopping equals a solution with ℓ_2-regularization where the regularization term is determined by the number of iterations and step-size in early stopping (Santos, 1996). Therefore, for the OL algorithm implemented in this research, the iterative optimization algorithm in (5) uses the posterior parameter estimate from the previous time step as a starting point but iterates only a few steps k towards the optimal value. In such a sense, the approach is similar to an early stopping approach, and will to some degree include parameter regularization.

2.4 Comparison of periodic batch and online learning

There are advantages and disadvantages to both passive learning methods. With OL, the model can quickly adapt to changes in process conditions. Further, as only new observations are used, old data may be discarded yielding low memory requirements. However, it has been shown that some machine learning models such as neural networks are prone to catastrophic forgetting when trained using OL (I.J. Goodfellow et al., 2013; Kemker et al., 2018; Parisi et al., 2019). Catastrophic forgetting is a situation where the model excessively overfits its parameters to new observations resulting in a decreased performance on previously seen observations. This situation occurs due to the stability-plasticity dilemma (Wickliffe and Robins, 2005). The neural network requires adequate plasticity to adapt to new patterns, but too much can cause the network to forget previously learned patterns. The reverse is true for stability. The stability-plasticity of the models is connected to the hyperparameters of the learning algorithms. With time, the optimal hyperparameters can change. This is a problem for OL in real-time applications as a hyperparameter search in each iteration can be infeasible, dependent on the frequency of arrival of new observations. Another potential issue for the OL is the required complex system integration. The method will require fast processing capabilities of new observations to account for erroneous sensor measurements, and model performance monitoring applications are a necessity to analyze model drift and catastrophic forgetting (Ditzler et al., 2015). Furthermore, the learning method must be automated as manual, although systematic, handling of model updating can be impractical in real-time due to limited resources.

PBL addresses catastrophic forgetting as all available observations are used in model updating. Yet, using this method for each new observation can be impractical in real-time applications due to a larger training time caused by larger datasets (Kemker et al., 2018). Therefore, a longer period (τ in (7)) between model retraining can be required and sudden shifts in the data can be missed. On the other hand, if the underlying process is slowly changing, a lower update frequency can be sufficient to capture dominant changes in process conditions. Correspondingly, a manual yet systematic handling of the learning method including measurement preprocessing, conducting a hyperparameter search, and the actual model learning can be more achievable in each iteration. For VFM applications, studies have indicated that the inclusion of too old data may be redundant and not improve the model performance significantly (T. AL-Qutami et al., 2018; B. Grimstad, Hotvedt, et al., 2021). Thus, a windowing strategy can be applied to discard redundant data (Ditzler et al., 2015).

3 Data and models

In this research, six different VFM model types are considered. The data used to develop the VFMs and examine the effect of the learning methods on the long-term prediction performance are real production data from 10 wells, W1-W10, on the Edvard Grieg asset (Lundin Energy Norway, 2020). The available data and the VFM model types are described in the below sections.

3.1 Available data

The available process data consists of observations from the $M = 10$ wells indexed by $j \in \{1, \dots, M\}$. The dataset of well j is $\{(\boldsymbol{x}_{t,j}, y_{t,j})\}_{t=1}^{N_j}$, where N_j is the number of observations, explanatory variables are $\boldsymbol{x}_{t,j} = (u, p_1, p_2, T_1, \eta_{\text{oil}}, \eta_{\text{gas}})_{t,j} \in \mathbb{R}^6$, and target variables are $y_{t,j} = Q_{t,j} \in \mathbb{R}$. The η_{oil} and η_{gas} are the fractions of oil and gas in the fluid mixture. Ideally, the fractions should be estimated using a different model, for instance, a wellbore model as in Kittilsen, Fjalestad, and Aasheim (2014). For simplification, the fractions are approximated using the measured phasic volumetric flows. Measurements of the target variable, the mixture volumetric flow rate, are from both well-tests conducted with a test separator, and from the multiphase flow meter in each well. Commonly, well-test measurements have higher accuracy than MPFM measurements as MPFM are prone to failure and drift over time (Falcone et al., 2013). The data from all wells is denoted by $\mathcal{D}$. Each of the datasets is generated using the processing technology in B. Grimstad, Gunnerud, et al. (2016). This technology compresses the data by removing fast transients. However, slow transients can still be present. Further, the datasets are passed through a set of filters that remove undesired, illogical measurements, for instance, negative pressures or negative flow rate measurements. The wells have an unequal number of observations spanning a different time range, see Figure 4. Some wells have historical observations back to 2016 while others have their first observations in late 2018. Further, there are periods where observations are lacking for some of the wells represented by white holes in the data in Figure 4. Here, the well in question can have been shut down, or the sensors failed. In total, there are 26743 observations from the 10 wells, spanning more than five years of production history. On average, there is less than one day between each measurement. The time between well-tests for a well is varying, with more than one

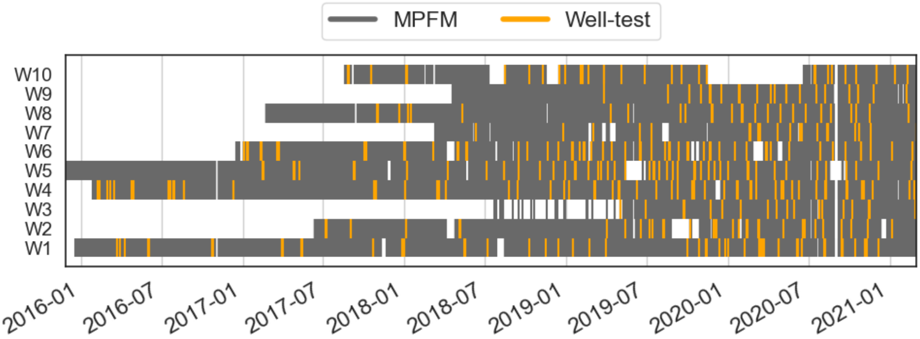


Figure 4: Visualization of the occurrence of observations for each well against time. Some wells have older historical observations than others. Both multiphase flow meter and well-test measurements are available.

year at the longest, and less than a day at the shortest.

3.2 Virtual flow meter models

The six different VFM models considered range from machine learning, or data-driven, to physics-based, or mechanistic, models:

1. A linear regression model (LR)
2. A fully connected feed-forward neural network (NN)
3. A multi-task learning model (MTL)
4. A hybrid, gray-box error model (HEM)
5. A hybrid, gray-box area function model (HAM)
6. A mechanistic model (MM)

There are advantages and disadvantages with all model types (Solle et al., 2016; Hotvedt, B. Grimstad, Ljungquist, et al., 2022). Mechanistic models are built from physical laws and require little process data in development. Yet, simplifications and assumptions are often necessary to make mechanistic models computationally feasible in real-time applications. Hence, model bias or process-model mismatch is typically encountered. Machine learning models are built from available data only and require no prior knowledge about the physics of the process. The capacity of machine learning models vary, where the NN is a typical model with high capacity and the LR a model with low capacity. High capacity models enable adaptation to arbitrarily complex physical relationships as long as these are reflected in the data, commonly reducing model bias. However, due to the inherent bias-variance trade-off of high capacity models, minimizing the bias results in higher variance (Hastie, Tibshirani, and Friedman, 2009). Therefore, high capacity models are often influenced by poor quality data or data located in the small data regime, a situation not uncommon for the VFM application (B. Grimstad, Hotvedt, et al., 2021). Furthermore, higher variance typically decreases the generalization abilities to previously unobserved data, and such models can struggle if used in nonstationary environments where the process experiences dataset shifts. The hybrid models attempt to utilize knowledge from both the mechanistic and the data-driven modeling domain to preserve the advantages but diminish the disadvantages of both methods.

The MTL models are somewhat different from the other model types. This model type enables learning from a plurality of wells, where each well presents a learning task. Instead of separately training a model for each well, which can be considered as single-task learning, the models are simultaneously trained. The advantage of using multi-task learning is two-fold. First, it allows for parameter sharing among models which can drastically improve data efficiency and predictive performance in the small-data regime. This is analogous to an MM whose equations are shared among wells. Second, compared to single-task learning, simultaneous training can lessen the effort and computational cost of developing models when the number of wells becomes large.

In the following sections, a mathematical description of the six VFM models is introduced. In addition to these, a benchmark model used to compare the performance of the models is described.

3.2.1 Benchmark model

A simple benchmark model predicts the flow rate to be the same as the last observed flow rate. Consider chronologically ordered observations $\{y_{1,j}, \dots, y_{N_j,j}\}$ for well j so that $y_{t,j}$ is observed after $y_{t-1,j}$. The prediction from the benchmark model is

$$\hat{y}_{t,j} = y_{t-1,j}, \quad t = 1, .., N_j, \quad j = 1, ..., M. \tag{13}$$

Note that with this model the prediction is independent of the explanatory variables $\boldsymbol{x}_{t,j}$. Further, if the petroleum production is on plateau, resulting in each new observation deviating little from the previous, the benchmark model has the potential of high accuracy.

3.2.2 Linear regression model

The linear regression model fits a multidimensional line to the observed data. The functional form is given by $f_{\boldsymbol{\theta}}^{(\mathrm{LR})} : \mathbb{R}^d \to \mathbb{R}$ and is evaluated for a given $\boldsymbol{x}$ as

$$\hat{y} = \boldsymbol{w}^T \boldsymbol{x} + b. \tag{14}$$

The model parameters consist of a weight vector $\boldsymbol{w} \in \mathbb{R}^d$ and a bias $b \in \mathbb{R}$, $\boldsymbol{\theta} = \{(\boldsymbol{w}, b)\}$.

3.2.3 Feed-forward neural network model

In general, the feed-forward neural network is a set of nonlinear regression lines. It has a functional form $f_{\boldsymbol{\theta}}^{(\mathrm{NN})} : \mathbb{R}^d \to \mathbb{R}$. For a neural network with L hidden layers and one output layer, the parameters are $\boldsymbol{\theta} = \{(W^{(l)}, \boldsymbol{b}^{(l)})\}_{l=1}^{L+1}$, where $W^{(l)}$ and $\boldsymbol{b}^{(l)}$ are the weights and biases of layer l, respectively. The dimensions of $W^{(l)}$ and $\boldsymbol{b}^{(l)}$ determine the width of layer l.

In this work, the rectified linear unit (ReLU) activation function is used as the nonlinearity in the hidden layers (Glorot, Bordes, and Yoshua Bengio, 2011). The ReLU function is denoted by $a : \mathbb{R}^d \to \mathbb{R}^d$, $a(\boldsymbol{z})_i := \max(0, \boldsymbol{z}_i)$, where the max operator is applied element-wise for $i = 1, \dots, d$. This makes the neural network a set of piecewise linear regression lines. The evaluation of model $f_{\boldsymbol{\theta}}^{(\mathrm{NN})}(\boldsymbol{x})$ for a given $\boldsymbol{x}$ is

$$\begin{aligned} \boldsymbol{z}^{(1)} &= \boldsymbol{x} \\ \boldsymbol{z}^{(l+1)} &= a(W^{(l)}\boldsymbol{z}^{(l)} + \boldsymbol{b}^{(l)}), \qquad & l = 1, \dots, L \\ \hat{y} &= W^{(L+1)}\boldsymbol{z}^{(L+1)} + \boldsymbol{b}^{(L+1)}. \end{aligned} \tag{15}$$

3.2.4 Multi-task learning model

A MTL formulation introduces a new semantics of the model parameters compared to the NN in Section 3.2.3. Let $\boldsymbol{\alpha}$ denote parameters that are shared among tasks (here wells), and let $\boldsymbol{\beta}_j \in \mathbb{R}^P$ be P task-specific parameters for wells $j = 1, \dots, M$. The parameters of the MTL model for M wells are collected in $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_M\}$.

When processing a data point $\boldsymbol{x}_{t,j}$ of well j, the model must select the corresponding task-specific parameters, $\boldsymbol{\beta}_j$. The selection can be made by introducing an encoding of tasks. Let $\boldsymbol{e}_j$ be an indicator vector of dimension M, with all zeros, except for a one in position j. By stacking the task-specific parameters in a matrix B with columns $B_{*,j} = \boldsymbol{\beta}_j$, a selection can be made by performing the multiplication $\boldsymbol{\beta}_j = B\boldsymbol{e}_j$.

A simple MTL model is obtained by utilizing the selection mechanism described above. First, $\boldsymbol{\beta}_j$ is selected using the encoding $\boldsymbol{e}_j$. Next, $\boldsymbol{x}_{t,j}$ and $\boldsymbol{\beta}_j$ are fed through a residual neural network with shared parameters $\boldsymbol{\alpha}$. In this work, a residual neural network with pre-activation is used to allow for an identity mapping of the task-specific parameters (He et al., 2016). The resulting model is a simplified version of the MTL choke model introduced in (Sandnes, Bjarne Grimstad, and Kolbjørnsen, 2021).

The functional form of the MTL model is $f_{\boldsymbol{\theta}}^{(\mathrm{MTL})} : \mathbb{R}^d \times \{0,1\}^M \to \mathbb{R}$, where the second argument is the task encoding vector. The evaluation of $f_{\boldsymbol{\theta}}^{(\mathrm{MTL})}(\boldsymbol{x}, \boldsymbol{e}_j)$ for a data point $\boldsymbol{x}$ of well j, is performed as follows:

$$\begin{aligned} \boldsymbol{\beta}_j &= B\boldsymbol{e}_j, \\ \hat{y} &= g_\alpha(\boldsymbol{x}, \boldsymbol{\beta}_j), \end{aligned} \tag{16}$$

where g_α is a residual neural network with L residual blocks given by

$$\begin{aligned} \boldsymbol{z}^{(1)} &= W^{(0,1)}\boldsymbol{x} + W^{(0,2)}\boldsymbol{\beta}_j + \boldsymbol{b}^{(0)}, \\ \boldsymbol{r}^{(l)} &= W^{(l,2)}a(W^{(l,1)}a(\boldsymbol{z}^{(l)}) + \boldsymbol{b}^{(l,1)}) + \boldsymbol{b}^{(l,2)}, \quad & l = 1, \dots, L, \\ \boldsymbol{z}^{(l+1)} &= \boldsymbol{r}^{(l)} + \boldsymbol{z}^{(l)}, & l = 1, \dots, L, \\ \hat{y} &= W^{(L+1)}\boldsymbol{z}^{(L+1)} + \boldsymbol{b}^{(L+1)}. \end{aligned} \tag{17}$$

The weights and biases in (17) are collected in $\boldsymbol{\alpha}$ and are shared among the M wells. These parameters can be learned from all the data in $\mathcal{D}$.

3.2.5 Mechanistic model

The mechanistic choke model is taken from Sachdeva et al. (1986). The equations are developed from the steady-state mass and momentum balance equations for one-dimensional flow along a streamline. In short notation, the mechanistic model is given by $f_{\boldsymbol{\theta}}^{(\mathrm{MM})} : \mathbb{R}^d \to \mathbb{R}$ with parameters $\boldsymbol{\theta} = \{\rho_{\mathrm{oil}}, \rho_{\mathrm{wat}}, \kappa, M_{\mathrm{gas}}, p_{cr}, C_D\}$, and the equation for the volumetric flow rate through the choke is given by:

$$\begin{aligned}\hat{y} = Q &= \frac{\dot{m}}{\rho_{SC}} \\ &= \frac{C_D A_2(u)}{\rho_{SC}} \times \sqrt{2\rho_2^2 p_1 \left(\frac{\kappa}{\kappa - 1}\eta_{\mathrm{gas}} \left(\frac{1}{\rho_{\mathrm{gas},1}} - \frac{p_r}{\rho_{\mathrm{gas},2}}\right) + \left(\frac{\eta_{\mathrm{oil}}}{\rho_{\mathrm{oil}}} + \frac{\eta_{\mathrm{wat}}}{\rho_{\mathrm{wat}}}\right)(1 - p_r)\right)},\end{aligned} \tag{18}$$

Details regarding the model are found in Hotvedt, B. Grimstad, Ljungquist, et al. (2022).

3.2.6 Hybrid error model

This model uses the mechanistic model in Section 3.2.5 as a baseline but inserts a neural network as introduced in Section 3.2.3 to capture the error between the mechanistic model output and measurements, or the process-model mismatch. The functional form of the model is given by $f_{\boldsymbol{\theta}}^{(\mathrm{HEM})} : \mathbb{R}^d \to \mathbb{R}$ with parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\mathrm{MM}}, \boldsymbol{\theta}_{\mathrm{NN}}\}$, where the physical model parameters are the same as given in Section 3.2.5: $\boldsymbol{\theta}_{\mathrm{MM}} = \{\rho_{\mathrm{oil}}, \rho_{\mathrm{wat}}, \kappa, M_{\mathrm{gas}}, p_{cr}, C_D\}$, and the neural network parameters are the weights and biases on each layer of the network as described in Section 3.2.3: $\boldsymbol{\theta}_{\mathrm{NN}} = \{(W^{(l)}, \boldsymbol{b}^{(l)})\}_{l=1}^{L+1}$. The evaluation of HEM for a data point $\boldsymbol{x}$ is described by

$$\hat{y} = f_{\boldsymbol{\theta}}^{(\mathrm{HEM})}(\boldsymbol{x}) = f_{\boldsymbol{\theta}_{\mathrm{MM}}}^{(\mathrm{MM})}(\boldsymbol{x}) + f_{\boldsymbol{\theta}_{\mathrm{NN}}}^{(\mathrm{NN})}(\boldsymbol{x}) \tag{19}$$

3.2.7 Hybrid area function model

This model also uses the mechanistic model in Section 3.2.5 as a baseline. However, the mechanistic relation for the area function $A_2(u)^{(\mathrm{MM})}$ is manipulated by multiplying with a neural network. This may be interpreted as replacing the discharge coefficient C_D from the MM with a neural network. Accordingly, $f_{\boldsymbol{\theta}}^{(\mathrm{HAM})} : \mathbb{R}^d \to \mathbb{R}$ with parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\mathrm{MM}}, \boldsymbol{\theta}_{\mathrm{NN}}\}$, where $\boldsymbol{\theta}_{\mathrm{MM}} = \{\rho_{\mathrm{oil}}, \rho_{\mathrm{wat}}, \kappa, M_{\mathrm{gas}}, p_{cr}\}$ and $\boldsymbol{\theta}_{\mathrm{NN}} = \{(W^{(l)}, \boldsymbol{b}^{(l)})\}_{l=1}^{L+1}$. The evaluation of $f_{\boldsymbol{\theta}}^{(\mathrm{HAM})}$ for data point $\boldsymbol{x}$ is as follows:

$$\begin{aligned} A_2 &= A_2(u)^{(\mathrm{MM})} \times f_{\boldsymbol{\theta}_{\mathrm{NN}}}^{(\mathrm{NN})}(\boldsymbol{x}) \\ \hat{y} &= f_{\boldsymbol{\theta}}^{(\mathrm{HAM})}(\boldsymbol{x}) = f_{\boldsymbol{\theta}_{\mathrm{MM}}}^{(\mathrm{MM})}(\boldsymbol{x}, A_2) \end{aligned} \tag{20}$$

Note, the complete vector of explanatory variables is used as input to the area function network and not just the choke opening u. This is due to the expectation of the effective flow area being dependent on the characteristics of the fluid flowing through the choke, which cannot be captured with just u.

3.3 Prior parameter distribution

All the VFM models except the benchmark model need specification of the prior parameter distributions $\theta_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. For the data-driven model parameters $\boldsymbol{\theta}_{\text{NN}}$, He-initialization is utilized, which is recommended for neural networks with ReLU as activation function (He et al., 2015). For the mechanistic model parameters $\boldsymbol{\theta}_{\text{MM}}$, typical values for the mean μ_i is commonly known. For instance, a typical value for the density of freshwater is $1000kg/m^3$. The variance may be estimated using the known bounds of the parameter in question. Details on prior parameter specification in gray-box models may be found in Hotvedt, B. Grimstad, Ljungquist, et al. (2022).

4 Numerical study

Online learning and periodic batch learning as described in Section 2.3, are used to train the six models in Section 3.2, for the 10 wells, using the data described in Section 3.1. All VFM models except the LR are implemented using the Python framework *PyTorch* (Paszke et al., 2019). The LR is implemented with the Python framework *scikit-learn* (Pedregosa et al., 2011) using the stochastic gradient descent linear regressor to allow for training the model with online learning. As mentioned in Section 2.3.2, the ℓ_2-regularization term cannot be calculated for the online learning method. However, for the hybrid and mechanistic models, an important factor is that the model parameters with a physical interpretation $\boldsymbol{\theta}_{\text{MM}}$ stay within feasible bounds. Therefore, ℓ_2-regularization with the initial priors is applied for these parameters.

The numerical study considers two cases. In Case 1, all available data, both MPFM and well-test measurements are utilized in training. The initial parameter estimate is obtained with historical data before the 1st of January 2019, while the data after this point in time is used to test the learning methods, see Figure 4. This split of data is referred to as the initial split. In Case 2, the models are trained using well-test measurements only. To ensure a sufficient amount of training data, the initial split is applied on the 1st of January 2020, see Figure 4.

Two analyses are conducted before the learning methods can be applied: 1) estimation of the PBL update frequency and 2) a search for optimal hyperparameters in the learning methods. These analyses are given in Section 4.1 and 4.2, respectively, and are applied on the initial training data. From the outcome of the analyses, the models are trained with the learning methods, and the result for the two cases is given in 4.3 and 4.4, respectively.

4.1 Update frequency estimation for periodic batch learning

To estimate a suitable update frequency, Algorithm 1 in A with significance level $\alpha = 0.05$ is used on the initial training data from Case 1. This data is split into two new datasets at time 01.07.2018. The six months of observations leading up to 01.01.2019 are used as the test dataset. From Figure 4, it is seen that W3 does not have observations in the time range suggested. Therefore, the well is excluded from the analysis. In Figure 5, the HT^2 statistic for each observation in the test dataset is illustrated for four of the wells. The coloring indicates whether or not a shift is detected for the observation. W1 and W2 are the two wells of the nine examined with the longest period before a shift is detected, approximately after five months. W8 and W9 are the wells with the shortest period before a shift is detected, approximately after two weeks. Accordingly, different wells can have different optimal update frequencies, and it is likely to change during the lifetime of the petroleum asset. To simplify model learning, all wells are trained using the same update frequency. Therefore, the PBL is tested with a two weeks update frequency. The results are compared to a PBL with an update frequency of 6 months to examine the potential benefit of more frequent updating.

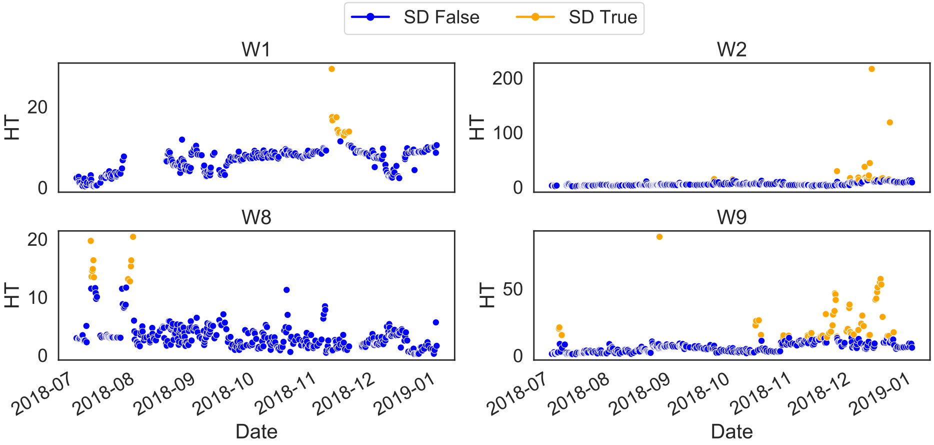


Figure 5: The Hotelling's T-squared statistic with $\alpha = 0.05$ for each of the observation in the test dataset. The coloring indicates if the test observation is detected as a shift (SD: shift detection).

4.2 Hyperparameter search

For the periodic batch learning approach, a hyperparameter grid search for the learning rate is conducted testing $\gamma \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. Early stopping is applied to determine the appropriate number of iterations E. For all VFM models except the LR, the optimizer Adam is applied. This optimizer have shown results in previous research on VFM modeling (Hotvedt, B. Grimstad, and Imsland, 2020; Hotvedt, B. Grimstad, and Imsland, 2021; Hotvedt, B. Grimstad, Ljungquist, et al., 2022; B. Grimstad, Hotvedt, et al., 2021). For the LR, Adam is not an option and the model is trained with SGD, yet, with the learning rate scheduler

$$\gamma^{(k)} = \frac{\gamma^{(0)}}{k^a} \tag{21}$$

where $\gamma^{(0)}$ is the initial learning rate, k is the iteration number, and a is a constant, see (5).

For the online learning approach, the hyperparameter grid search is extended to $\gamma \in \{5 \times 10^{-1}, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-10}\}$. Notice, $\gamma = 10^{-10}$ means close to negligible updating. As online learning processes only one sample at a time, early stopping cannot be applied. Therefore, the hyperparameter search includes experimentation with the number of iterations $E \in \{1, 10, 20\}$. For all models except the LR, the optimizers SGD and Adam are examined. For the LR, the learning rate scheduler (21) along with a constant learning rate is investigated.

The best combination of hyperparameters is chosen as the set that minimized the mean absolute percentage error (MAPE) across the wells for each model type. The resulting hyperparameters for Case 1 and Case 2 can be seen in Tables B1 and B2, respectively.

4.3 Results of Case 1

In this case, both MPFM and well-test measurements are utilized in training. The box plot in Figure 6 shows the distribution of performances for the wells in terms of the MAPE grouped on the model type and learning method. The reported MAPE for one well is calculated using the predictions on all observations in the initial test set. The models are compared to the benchmark model. Table 1 gives an overview of the average MAPE across the wells for each model, and the last column presents the average MAPE of the learning methods across all wells and models. For the interested reader, Table C1 gives a detailed overview of the MAPEs for each well and model. There are several interesting observations.

Firstly, as expected, the results clearly show that the model error decreases with an increased update frequency. On average, all models achieve a lower prediction error with PBL every second week compared to PBL every six months. With the OL, the average error decreases further with all models achieving an average error

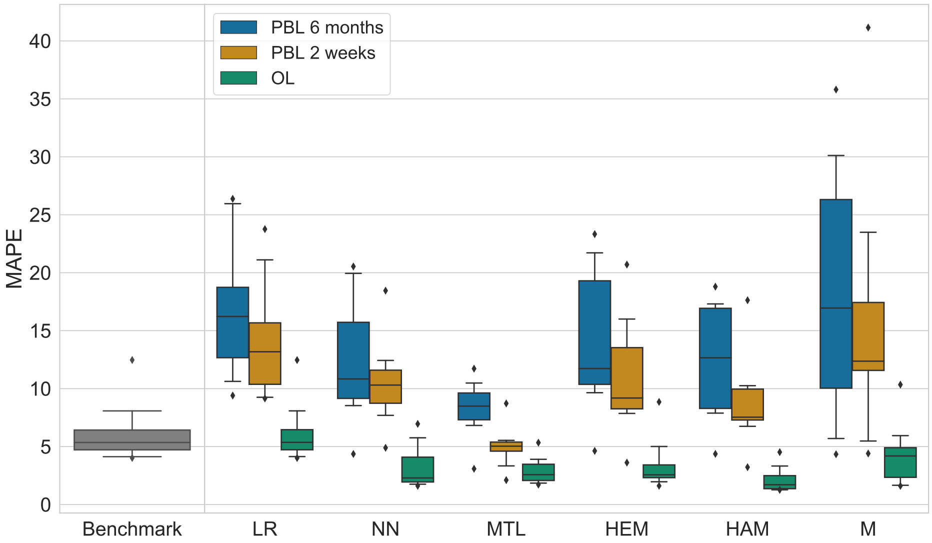


Figure 6: The distribution of average error for each well, grouped for the models and learning methods. The models are trained with all available measurements. Compared to the performance of the benchmark model. The boxes show the P_{25}, P_{50} (median), and P_{75} percentiles. The whiskers show the P_{10} and P_{90} percentiles.

Table 1: Average mean absolute percentage error across the wells for the models and learning methods trained on both MPFM and well-test measurements. The last column is the average MAPE across all wells and models.

Learning method	LR	NN	MTL	HEM	HAM	M	All
PBL 6 months	16.8	12.4	8.3	14.2	12.4	18.1	13.7
PBL 2 weeks	14.2	10.5	5.0	10.9	8.7	15.7	10.8
OL	6.2	3.2	2.9	3.4	2.1	4.2	3.7

of less than 7%. The overall best average performance across wells is achieved with OL on the HAM. The low MAPEs indicate that with access to frequently arriving measurements such as MPFM measurements, and allowed to learn continuously from them, the learning problem is relatively simple and a complex model is not necessary to achieve high VFM accuracy. This is supported by the good performance of the Benchmark which outperforms nearly all models trained with PBL. On the other hand, a disadvantage with the Benchmark is that it cannot be used for sensitivity analyses or in production optimization.

Secondly, from Table C1 it is observed that there are large differences in the error reduction for each well when the update frequency is increased. For instance, for W9 and most models, the error is greatly reduced going from the PBL 6 months to the OL. On the other hand, for W1 the reduction is not as prominent. This is likely related to whether or not the data generating distribution shifts with time. In Figure 7, the Hotelling's T-squared statistic is plotted for W1 and W9 using Algorithm 1 on the initial training and test data. Figure 7 indicates that it is unlikely that W1 experiences dataset shifts. On the other hand, for W9 it can be observed that the data likely shifts with time. Therefore, the results in Table C1 indicate that OL is better at tracking the local optimum of the learning problem when it changes with time.

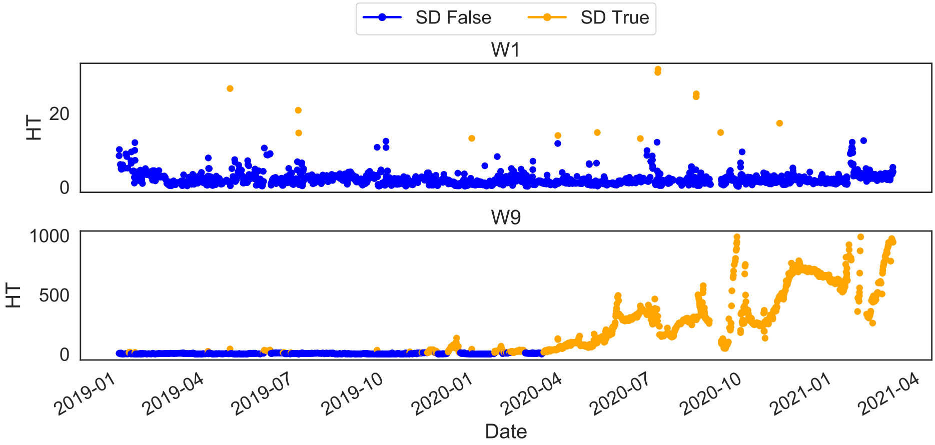


Figure 7: The Hotelling's T-squared statistic for W1 and W9 comparing each observation from 01.01.2019 and forward with time to the training data containing the historical data before 01.01.2019. As seen, for W1 most observations are not detected as shifts. Whereas for W9, all observations towards to end are marked as a shift.

Another figure that illustrates the benefit of updating the model more often is Figure 8, where the prediction error is visualized against time. The error is calculated as a rolling absolute mean error with a window size of 14 days. The shaded regions visualize the 25 and 75 percentiles of the errors across the wells. Notice that the PBL seems to yield a cyclic high and low accuracy. The average error increases with time up until model updating where the average error is reduced, naturally after some delay due to the rolling window. This is best observed for the PBL 6 months, but also to some extent for the PBL 2 weeks.

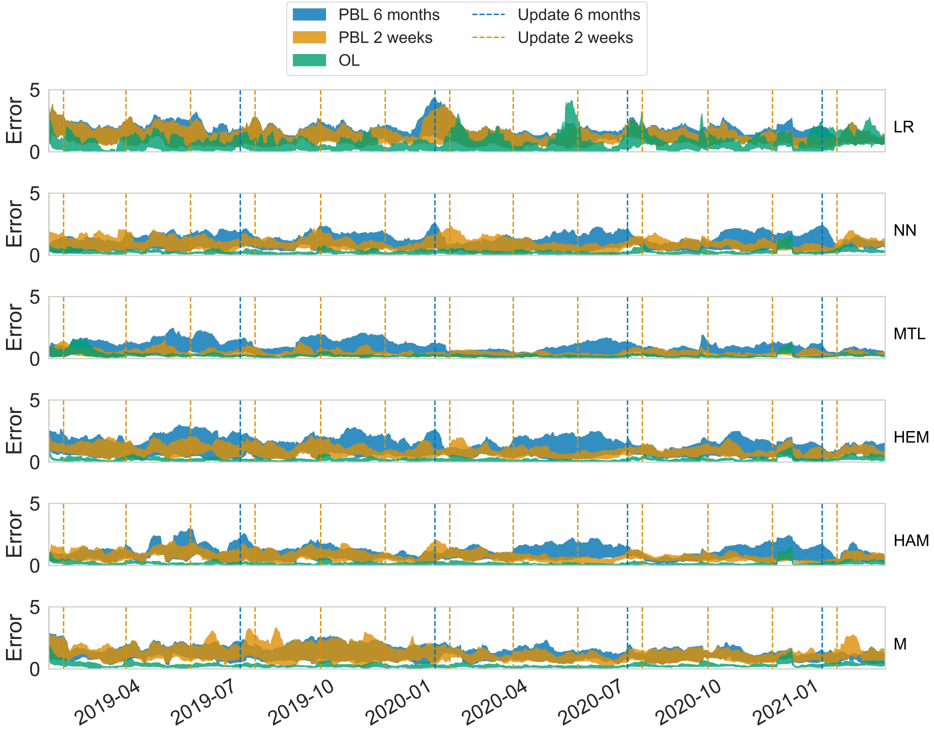


Figure 8: The rolling absolute mean error across the wells against time for the models and learning methods. The window size used to calculate the error is 14 days. The shaded region illustrates the 25 and 75 percentiles of the errors across the wells. The vertical lines illustrate where the models are updated for the PBL 6 months and PBL 2 weeks.

4.4 Results of Case 2

In this case, the models are trained on well-test measurements only, see the observations colored orange in Figure 4. Figure 9 illustrates the distribution of MAPEs for the wells. Table 2 gives an overview of the average MAPE across the wells. Table C2 reports the MAPE for each well, model, and method. First of all, notice the significantly different results obtained for this case compared to Case 1. In Case 1, a trend of decreased error for increased update frequency is observed. Here, the difference in performance is negligible for many models and for other models the error increases going from PBL to OL. The observed results are likely related to the elapsed time between each new well-test, illustrated in Figure 10 by a stacked histogram. Notice that many of the wells have several tests that are more than a month apart. Furthermore,

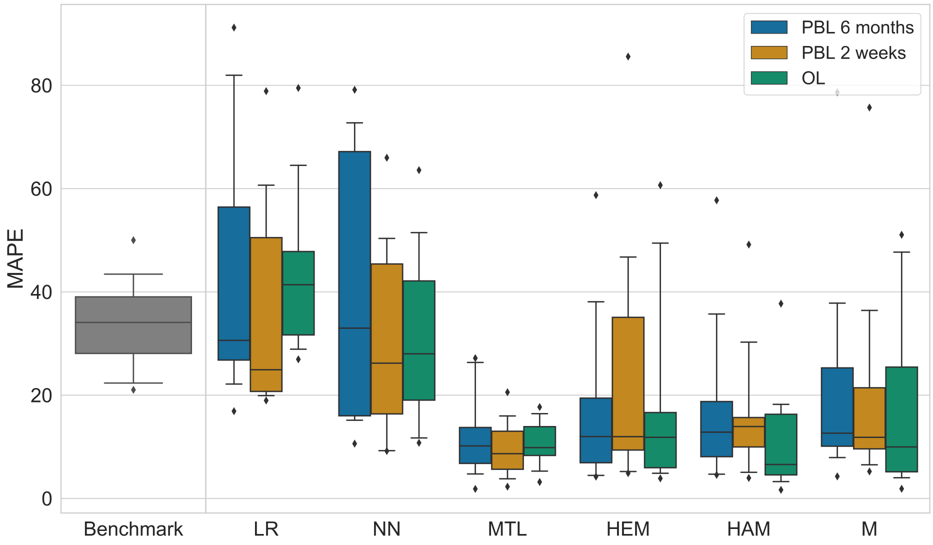


Figure 9: The distribution of mean absolute percentage error (MAPE) for each well, grouped for the models and learning methods. Here the models are trained using well-test measurements only. Compared to the performance of the benchmark model. The boxes show the P_{25}, P_{50} (median), and P_{75} percentiles. The whiskers show the P_{10} and P_{90} percentiles.

Table 2: The average mean absolute percentage error across the wells for the models and learning methods trained on well-test measurements. The last column is the average MAPE across all wells and models.

Learning method	LR	NN	MTL	HEM	HAM	M	All
PBL 6 months	43.2	40.7	12.1	17.9	18.3	21.8	25.7
PBL 2 weeks	37.1	31.1	9.6	24.5	16.7	20.5	23.3
OL	44.3	31.4	10.5	18.7	11.6	17.7	22.4

eight of ten wells have the majority of tests occurring with a frequency lower than 14 days, see Table C3. In such situations, the frequency of model updating is equal for PBL 2 weeks and OL, and the only difference between the two is how the updating is executed. The low frequency of well-tests is also likely the cause of the decreased Benchmark performance compared to Case 1. With a lower frequency, the process conditions can have changed significantly in-between well-tests and two chronological flow rate measurements are likely uncorrelated. The intermittent time between well-tests also makes it challenging to obtain good hyperparameters. If well-tests occur frequently, the model will likely require small parameter updates, and opposite for infrequently arriving well-tests. Non-optimal hyperparameters can explain the overall poorer average performance for all models and methods than for Case 1.

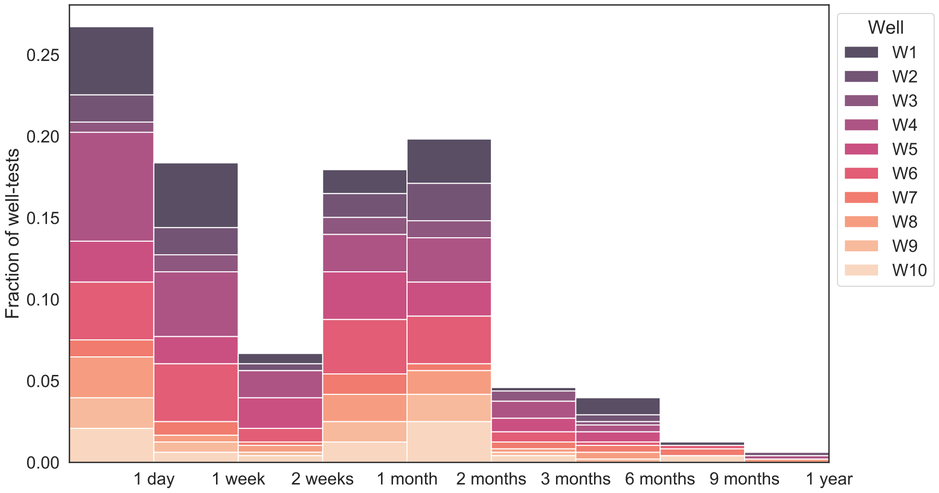


Figure 10: The elapsed time between new well-tests for each well present in the dataset. Keep in mind that the bins in the histogram have different sizes, from one day to three months of elapsed time. Observe that many of the wells have measurements occurring more than two weeks apart.

The large MAPEs in Table 2 show that in the presence of infrequent and intermittent measurements, the learning problem is not trivial and a more complex model than, for instance, the Benchmark, is required to obtain an adequate performance. Nevertheless, the comparable performance of the LR and NN indicate that choosing a data-driven model with higher complexity is not the solution to increased performance in this case. Likely, the amount of data available is too small for high-capacity data-driven models to exploit their capacity. An observation that supports this is that the NN obtained a significant improved performance in Case 1 where the amount of data is higher. On the other hand, having physical considerations in the model structure does seem to be be advantageous for VFM when the amount of data is small. From Figure 9, the MTL, HEM, HAM, and M, all achieve a median MAPE below 20% whereas the NN and LR are well above 20%. Another interesting observation is that learning from several wells as for the MTL seems to yield a more robust approach as the spread in performances is low.

5 Concluding remarks

The results in this research show that a high update frequency of steady-state VFM models is key to sustaining a high performance in nonstationary conditions over time. In particular, if the frequency of measurement arrival is high. Therefore, for petroleum assets with access to multiphase flow meters, steady-state VFMs can yield an excellent performance. Of the two passive learning methods analyzed, online learning achieves the best average performance with an error of 3.7% across all wells and model types. This is an error reduction of 73% compared to the average error of periodic batch learning with an update frequency of 6 months. On the other hand, if the arrival of new measurements is intermittent and with low frequency, which is a common issue on assets with well-testing only, the benefit of frequent model updating is small and less evident. This is likely due to the challenging task of finding good hyperparameters. The average error increased significantly with online learning achieving an average error of 22.4% across all wells and model types. However, an interesting observation is that VFM model types with physical considerations seem to offer the best performance in the presence of little data.

Hence, the results show that online learning seems a promising method to obtain high accuracy steady-state VFM models, in particular, with frequently arriving measurements. However, the method will also require fast measurement processing capabilities, model performance monitoring applications, and automatic handling of the learning process. Therefore, online learning can be challenging to integrate into existing systems. Likely, an appropriate learning method must be chosen as a trade-off between accuracy and available resources. With limited resources, periodic batch learning with frequent model updating, for instance, every second week, can be better suited in real-time applications.

Although the learning methods in this research are investigated for 10 typical subsea wells on the Norwegian continental shelf, these are certainly not representative for all production wells as the multiphase flow characteristics can be very different. Therefore, it is hard to generalize the results and it would benefit the conclusion if more wells from different assets are included. Nevertheless, the overall conclusion of this research is that passive learning with frequent model updating can significantly improve the accuracy of steady-state VFMs in nonstationary environments. The investigation can be of interests to experts developing soft-sensors, like VFMs.

Acknowledgments

The authors would like to thank Lundin Energy Norway for allowing them to work with production data from a real petroleum field. They would further like to thank Solution Seeker AS for the contribution with data collection and pre-processing.

Funding

This research is supported by Lundin Energy Norway and is a part of BRU21 - NTNU Research and Innovation Program on Digital and Automation Solutions for the Oil and Gas Industry (www.ntnu.edu/bru21). Lundin Energy Norway has not taken part in data collection and analysis, nor in writing of the report. However, the article is approved by Lundin before submission for publication.

A Estimation of the update frequency in periodic batch learning

Consider the null hypothesis $\mathcal{H}_0$ to state that there is no virtual drift present in the data such that input distribution $p(\boldsymbol{x})$ does not shift with time. This $\mathcal{H}_0$ is also called the stationary hypothesis. The $\mathcal{H}_1$ hypothesis is the alternative hypothesis that there is a shift in the data. Mathematically:

$$\begin{aligned} \mathcal{H}_0: &\quad p_t(\boldsymbol{x}) = p_{t+\tau}(\boldsymbol{x}) \quad \text{for all } \tau > 0 \\ \mathcal{H}_1: &\quad p_t(\boldsymbol{x}) \neq p_{t+\tau}(\boldsymbol{x}) \quad \text{for any } \tau > 0. \end{aligned} \tag{22}$$

Consider two disjoint datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ with size N_1 and N_2 and inputs observations $\boldsymbol{X}_1 \in \mathbb{R}^{d \times N_1}$ and $\boldsymbol{X}_2 \in \mathbb{R}^{d \times N_2}$, respectively. The Hotelling's T-squared statistic calculates the probability of equal means of the two multivariate input distributions at a significance level α. The statistic is calculated as

$$HT^2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\left(\frac{\boldsymbol{\Sigma}_1}{N_1} + \frac{\boldsymbol{\Sigma}_2}{N_2}\right)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top, \tag{23}$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the sample mean vector and the $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ is the sample covariance matrix of the input. The Hotelling's T-squared statistic follows the F-distribution $F(d, N_1 + N_2 - d - 1)$ (Härdle and Simar, 2012). To estimate an appropriate update frequency in PBL, the two-sample Hotelling's T-squared test can be used on available training data using Algorithm 1

Algorithm 1 Estimation of the update frequency τ in periodic batch learning

```
Require: data $\mathcal{D}_{1:T} = \{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$, significance level $\alpha$
Set $\mathcal{D}_1 = \mathcal{D}_{1:T_1}$ where $1 < T_1 < T$.
for $k = 1, ..., T - T_1$ do
    $\mathcal{D}_{2,k} = \mathcal{D}_{T_1+k}$
    Calculate $HT^2$ using (23) with $\mathcal{D}_1$ and $\mathcal{D}_{2,k}$
    Calculate F-statistic $F_k$ for $HT^2$
    Calculate the critical value $F_{\text{crit}}$ at significance level $\alpha$
    if $F_k < F_{\text{crit}}$ then
        Reject $\mathcal{H}_0$, shift detected
        return $\tau = T_1 + k$
    end if
end for
```

The algorithm is subject to false shift detections, or type II error, for instance, if the observation is faulty or noisy. A workaround is to test additional observations following with time. If shifts are detected on several subsequent observations, virtual drift has likely occurred. If the following observations are not detected as shifts, likely, the detection is falsely reported.

B Hyperparameter search

Table B1: The training algorithm settings as a result of the hyperparameter search for training on both MPFM and well-test measurements. For the batch learning approaches, only the value of the learning rate γ is experimented with. The number of iterations E for PBL is found with early stopping (E.S.). The (s.) and (c.) for the LR refers to using the learning rate scheduler in (21) and constant learning rate, respectively.

Model	PBL 6 months			PBL 2 weeks			OL		
	γ	E	O	γ	E	O	γ	E	O
LR	(s.) 10^{-2}	E.S.	SGD	(s.) 10^{-1}	E.S.	SGD	(c.) 0.5	20	SGD
NN	10^{-4}	E.S.	Adam	10^{-3}	E.S.	Adam	10^{-5}	20	Adam
MTL	10^{-4}	E.S.	Adam	10^{-3}	E.S.	Adam	10^{-6}	20	Adam
HEM	10^{-3}	E.S.	Adam	10^{-3}	E.S.	Adam	10^{-2}	20	SGD
HAM	10^{-3}	E.S.	Adam	10^{-3}	E.S.	Adam	10^{-5}	20	SGD
M	10^{-3}	E.S.	Adam	10^{-3}	E.S.	Adam	10^{-2}	10	Adam

Table B2: The training algorithm settings as a result of the hyperparameter search for training on only well-test measurements. For the batch learning approaches, only the learning rate γ is experimented with. The number of iterations E for PBL is found with early stopping (E.S.). The (s.) and (c.) for the LR refers to using the learning rate scheduler in (21) or constant learning rate, respectively.

Model	PBL 6 months			PBL 2 weeks			OL		
	γ	E	O	γ	E	O	γ	E	O
LR	(s.) 10^{-4}	E.S.	SGD	(s.) 0.5	E.S.	SGD	(s.) 10^{-3}	1	SGD
NN	10^{-4}	E.S.	Adam	10^{-3}	E.S.	Adam	10^{-4}	20	SGD
MTL	10^{-5}	E.S.	Adam	10^{-5}	E.S.	Adam	10^{-5}	20	Adam
HEM	10^{-3}	E.S.	Adam	10^{-4}	E.S.	Adam	10^{-10}	20	SGD
HAM	10^{-5}	E.S.	Adam	10^{-5}	E.S.	Adam	10^{-5}	20	Adam
M	10^{-3}	E.S.	Adam	10^{-3}	E.S.	Adam	10^{-2}	10	Adam

C Additional results from the numerical study

Table C1: Mean absolute percentage errors for all wells and models. The triple of numbers reported is the error for periodic batch learning every 6 months, periodic batch learning every 2 weeks, and online learning, respectively.

Well	LR	NN	MTL	HEM	HAM	M
1	9.4, 9.3, 12.5	4.4 4.9, 1.6	3.1, 2.1, 1.7	4.6, 3.6, 2.0	4.4, 3.2, 1.3	4.3, 4.4, 1.7
2	26.0, 21.1, 5.4	16.4, 18.5, 5.8	10.5, 8.7, 5.3	19.4, 20.7, 8.9	18.8, 17.6, 4.5	35.8, 41.2, 10.3
3	12.6, 9.8, 4.0	10.0, 11.7, 7.0	9.8, 4.9, 3.9	10.9, 8.8, 5.0	14.7, 10.2, 2.7	30.1, 11.7, 5.9
4	12.8, 12.0, 8.1	10.4, 9.8, 1.8	7.2, 5.0, 1.8	10.5, 9.2, 2.4	8.1, 7.3, 1.3	5.7, 5.5, 2.3
5	18.6, 16.0, 6.7	8.9, 8.5, 3.2	7.5, 3.3, 2.1	12.6, 9.2, 1.6	10.6, 7.6, 1.4	13.5, 11.5, 1.6
6	17.8, 14.8, 5.3	13.7, 11.1, 2.4	8.6, 5.4, 3.6	19.0, 16.0, 2.3	17.1, 10.3, 1.7	27.4, 23.5, 4.1
7	26.4, 23.8, 5.7	20.5, 12.4, 4.4	11.7, 5.1, 3.1	23.3, 11.9, 2.9	17.3, 9.1, 3.3	20.4, 16.5, 4.3
8	14.7, 12.6, 4.1	11.2, 10.8, 1.9	6.8, 5.5, 2.5	9.6, 7.9, 2.7	7.9, 6.7, 1.7	11.2, 12.8, 5.0
9	10.6, 9.2, 5.3	8.5, 7.7, 2.2	8.3, 5.3, 2.1	10.3, 8.1, 1.7	8.9, 7.4, 1.9	9.7, 12.0, 2.6
10	18.8, 13.7, 4.5	19.9, 9.4, 2.2	9.2, 4.5, 2.7	21.7, 14.1, 3.6	16.4, 7.3, 1.9	23.0, 17.8, 4.5

Table C2: Mean absolute percentage errors for all wells and models trained on only test separator measurements. The triple of numbers reported is the error for periodic batch learning every 6 months, periodic batch learning every 2 weeks, and online learning, respectively.

Well	LR	NN	MTL	HEM	HAM	M
1	26.5, 19.9, 28.9	10.6, 9.3, 10.8	1.9, 2.3, 3.2	4.2, 5.2, 7.9	4.6, 4.0, 1.7	4.3, 6.5, 1.9
2	91.2, 78.8, 64.5	62.0, 66.0, 30.3	27.2, 16.0, 17.7	58.7, 85.6, 49.4	57.7, 49.2, 37.7	78.7, 75.7, 47.7
3	22.2, 20.0, 46.9	79.1, 25.4, 63.6	11.1, 10.7, 14.7	21.6, 46.8, 17.6	14.1, 14.7, 16.5	28.9, 23.8, 51.1
4	32.9, 24.4, 34.8	16.6, 14.7, 18.5	11.0, 10.1, 8.4	12.8, 10.6, 5.3	10.8, 9.9, 7.0	9.9, 10.5, 5.8
5	28.3, 25.4, 26.9	15.1, 9.2, 11.7	4.7, 3.8, 5.3	4.5, 4.9, 3.9	7.2, 10.2, 4.3	7.9, 5.2, 4.9
6	82.0, 60.7, 48.1	72.7, 47.0, 43.7	14.6, 13.8, 11.4	8.6, 11.8, 13.3	11.6, 13.9, 5.4	14.5, 13.1, 8.8
7	43.5, 50.8, 79.5	68.9, 50.4, 51.5	9.3, 6.4, 11.2	11.7, 18.2, 13.7	19.1, 16.0, 18.2	10.8, 14.2, 11.6
8	16.9, 19.0, 30.6	15.8, 21.3, 20.8	6.3, 5.4, 8.3	6.4, 12.2, 10.4	4.7, 5.1, 6.2	12.8, 9.5, 11.1
9	27.7, 22.9, 36.7	27.1, 27.0, 25.7	8.1, 7.2, 8.5	12.3, 9.0, 4.9	17.6, 14.0, 3.3	12.5, 9.9, 4.0
10	60.7, 49.8, 46.1	38.9, 40.5, 37.8	26.3, 20.6, 16.4	38.1, 40.7, 60.7	35.7, 30.3, 15.6	37.8, 36.4, 30.1

Table C3: The percentage of well-tests for a well where the number of days between two chronological tests resides in the given bins. d=day, w=week, m=month, y=year.

Well	< 2w	2w-1m	1m-2m	2m-3m	3m-6m	6m-9m	9m-1y
1	60.9%	10.1%	18.8%	1.5%	7.2%	1.5%	0%
2	46.2%	17.9%	28.2%	0%	5.1%	0%	2.6%
3	36.4%	22.7%	22.7%	13.6%	4.5%	0%	0%
4	64.8%	12.1%	14.3%	5.5%	2.2%	0%	1.1%
5	48.3%	23.3%	16.7%	6.7%	5.0%	0%	0%
6	52.1%	21.9%	19.2%	4.1%	1.4%	1.4%	0%
7	41.7%	25.0%	8.3%	8.3%	8.3%	8.3%	0%
8	45.7%	22.9%	20.0%	2.9%	5.7%	0%	2.9%
9	44.8%	20.7%	27.6%	3.4%	3.4%	0%	0%
10	40.5%	16.2%	32.4%	5.4%	0%	5.4%	0%

References

Amin, A. (2015). “Evaluation of Commercially Available Virtual Flow Meters (VFMs)”. In: *Proceedings of the Annual Offshore Technology Conference*, pp. 1293–1318.

Bikmukhametov, T. and J. Jäschke (2019). “Oil Production Monitoring using Gradient Boosting Machine Learning Algorithm.” In: *IFAC-PapersOnLine* 52 (1), pp. 514–519.

Bikmukhametov, Timur and Johannes Jäschke (2020). “Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models”. In: *Computers and Chemical Engineering* 138.

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning.* Springer.

Ditzler, G. et al. (2015). “Learning in Nonstationary Environments: A survey”. In: *IEEE Computational Intelligence Magazine*, pp. 12–25.

Falcone, G. et al. (2013). “Multiphase flow metering: current trends and future developments”. In: *SPE Annual Technical Conference and Exhibition.*

Foss, B., B. R. Knudsen, and B. Grimstad (2018). “Petroleum production optimization - A static or dynamic problem”. In: *Computers and Chemical Engineering* 114, pp. 245–253.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). “Deep Sparse Rectifier Neural Networks”. In: *14th International Conference on Artificial Intelligence and Statistics.* Vol. 15, pp. 315–323.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning.* Cambridge, Massachusetts. London, England.: The MIT Press.

Goodfellow, I.J. et al. (2013). “An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks”. In: *arXiv:1312.6211.*

Granero-Belinchón, C., S. G. Roux, and N. B. Garnier (2019). “Information Theory for Non-Stationary Processes with Stationary Increments”. In: *Entropy* 21.

Gravdahl, T. and O. Egeland (2002). *Modeling and Simulation for Automatic Control.* Marine Cybernetics AS.

Grimstad, B., V. Gunnerud, et al. (Sept. 2016). “A Simple Data-Driven Approach to Production Estimation and Optimization”. In: *In: SPE Intelligent Energy International Conference and Exhibition.*

Grimstad, B., M. Hotvedt, et al. (2021). “Bayesian Neural Networks for Virtual Flow Metering: An Empirical Study”. In: *Applied Soft Computing* 112.

Guo, B., W.C. Lyons, and A. Ghalambor (2007). *Petroleum Production Engineering - a computer assisted approach.* Elsevier Science and Technology Books.

Härdle, W. and L. Simar (2012). *Applied Multivariate Statistical Analysis, 3rd edition.* Springer.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning.* New York, USA: Springer.

He, Kaiming et al. (2015). “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

He, Kaiming et al. (Mar. 2016). “Identity Mappings in Deep Residual Networks”. In: *Computer Vision - ECCV 2016*, pp. 630–645. eprint: `1603.05027`.

Holmås, K. and A. Løvli (2011). “FlowManager dynamic: A multiphase flow simulator for online surveillance, optimization and prediction of subsea oil and gas production”. In: *BHR Group - 15th International Conference on Multiphase Production Technology*, pp. 241–254.

Hotvedt, M., B. Grimstad, and L. Imsland (2020). “Developing a Hybrid Data-Driven, Mechanistic Virtual Flow Meter - a Case Study”. In: *IFAC-PapersOnLine* 53 (2), pp. 11692–11697.

— (2021). “Identifiability and interpretability of hybrid, gray-box models”. In: *IFAC-PapersOnLine* 54 (3), pp. 389–394.

Hotvedt, M., B. Grimstad, D. Ljungquist, et al. (2022). “On gray-box modeling for virtual flow metering”. In: *Control Engineering Practice* 118.

Jansen, Jan-Dirk (2015). *Nodal Analysis of Oil and Gas Wells - Theory and Numerical Implementation.* TU Delft, The Netherlands: Delft University of Technology.

Jordanou, J. et al. (2017). “Recurrent neural network based control of an oil well”. In: *XIII Simpósio Brasileiro de Automação Inteligente*, pp. 924–931.

Kemker, Ronald et al. (2018). “Measuring Catastrophic Forgetting in Neural Networks Ronald”. In: *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18).* Vol. 1, pp. 3390–3398.

Kittilsen, P., K. Fjalestad, and R. Aasheim (2014). “Stabilized and Increased Well Production Using Automatic Choke Control”. In: *Society of Petroleum Engineers.*

Koroteev, D. and Z. Tekic (2021). “Artificial intelligence in oil and gas upstream : Trends , challenges , and scenarios for the future”. In: *Energy and AI* 3, pp. 1–10.

Lundin Energy Norway (2020). *Edvard Grieg.* `https://lundin-energy-norway.com/edvard-grieg/`. Acccessed: 18.01.2020.

Monteiro, Danielle D. et al. (2020). “Using Data analytics to quantify the impact of production test uncertainty on oil flow rate forecast”. In: *IFP Energies Nouvelles* 75 (7), pp. 1–15.

Oliveira, G., L. Minku, and A. Oliveira (2021). “Tackling Virtual and Real Concept Drifts: An Adaptive Gaussian Mixture Model Approach”. In: *IEEE Transactions on Knowledge and Data Engineering.*

Parisi, German I. et al. (2019). “Continual lifelong learning with neural networks: A review”. In: *Neural Networks* 113 (1), pp. 54–71.

Paszke, Adam et al. (2019). “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32.* Ed. by H. Wallach et al. Curran Associates, Inc., pp. 8024–8035.

Pedregosa, F. et al. (2011). “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Quiñonero-Candela, J. et al. (2009). *Dataset shift in machine learning.* Cambridge, Massachusetts. London, England.: MIT Press.

AL-Qutami, T.A et al. (2018). “Virtual multiphase flow metering using diverse neural network ensemble and adaptive simulated annealing”. In: *Expert Syst. Appl.* 93, pp. 72–85.

AL-Qutami, Tareq Aziz, Rosdiazli Ibrahim, and Idris Ismail (2017). "Hybrid neural network and regression tree ensemble pruned by simulated annealing for virtual flow metering application." In: *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 304–309.

AL-Qutami, Tareq Aziz, Rosdiazli Ibrahim, Idris Ismail, and Mohd Azmin Ishak (2017a). "Development of soft sensor to estimate multiphase flow rates using neural networks and early stopping". In: *Int. J. Smart Sens. Intell. Syst.* Vol. 10, pp. 199–222.

— (2017b). "Radial basis function network to predict gas flow rate in multiphase flow." In: *Proceedings of the 9th International Conference on Machine Learning and Computing*, pp. 141–146.

Raza, H., G. Prasad, and Y. Li (2015). "EWMA model based shift-detection methods for detecting covariate shifts in non-stationary environments". In: *Pattern Recognition* 48 (3), pp. 659–669.

Sachdeva, R. et al. (1986). "Two-phase flow through chokes". In: *Society of Petroleum Engineers, Annual Technical Conference and Exhibition.*

Sandnes, Anders T., Bjarne Grimstad, and Odd Kolbjørnsen (2021). "Multi-task learning for virtual flow metering". In: *Knowledge-Based Systems* 232.

Santos, R. J. (1996). "Equivalence of regularization and truncated iteration for general ill-posed problems". In: *Linear Algebra and its Applications* 236, pp. 25–33.

Sayed-Mouchaweh, M. and E. Lughofer (2012). *Learning in non-stationary environments: methods and applications.* Springer.

Shippen, M. (2012). "Steady-State Multiphase Flow - Past, Present, and Future, with a Perspective on Flow Assurance". In: *Energy and Fuels* 26, pp. 4145–4157.

Solle, D. et al. (2016). "Between the Poles of Data-Driven and Mechanistic Modeling for Process Operation". In: *Chemie Ingenieur Technik.*

Toskey, E.D. (2012). "Improvements to Deepwater Subsea Measurements RPSEA Program: Evaluation of Flow Modeling". In: *Offshore Technology Conference.*

Varyan, R., R.K. Haug, and D.G. Fonnes (2015). "Investigation on the suitability of virtual flow metering system as an alternative to the conventional physical flow meter". In: *SPE, APOGCE*, pp. 2–10.

Wickliffe, C. A. and A. Robins (2005). "Memory retention - the synaptic stability versus plasticity dilemma". In: *Trends in Neurosciences* 28 (2), pp. 73–78.

References

Amin, A. (2015). “Evaluation of Commercially Available Virtual Flow Meters (VFMs)”. In: *Proceedings of the Annual Offshore Technology Conference*, pp. 1293–1318.

Andrade, Gabriel M.P., Diego Q.F. de Menezes, Rafael M. Soares, Tiago S.M. Lemos, Alex F. Teixeira, Leonardo D. Ribeiro, Bruno F. Vieira, and José Carlos Pinto (2022). “Virtual flow metering of production flow rates of individual wells in oil and gas platforms through data reconciliation”. In: *Journal of Petroleum Science and Engineering* 208, p. 109772. DOI: `https://doi.org/10.1016/j.petrol.2021.109772`.

Antonelo, E. A., E. Camponogara, and B. Foss (2009). “Data-driven Soft Sensors in the process industry”. In: *Computers and Chemical Engineering* 33 (4), pp. 795–814.

— (2017). “Echo State Networks for data-driven downhole pressure estimation in gas-lift oil wells”. In: *Neural Networks* 85, pp. 106–117.

Baraldi, P., F. Mangili, G. Gola, B.H. Nystad, and E. Zio (2014). “A hybrid ensemble-based approach for process parameter estimation and degradation assesment in offshore oil platforms”. In: *International Journal of Performability Engineering* 10 (5), pp. 497–509.

Beck, J. V. and K. J. Arnold (1977). *Parameter Estimation in Engineering and Science.* Wiley and Sons.

Bengio, Y. (2012). “Practical Recommendations for Gradient-Based Training of Deep Architectures.” In: *arXiv:1206.5533v2*, pp. 1–33.

Bikmukhametov, Timur and Johannes Jäschke (2020a). “Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models”. In: *Computers and Chemical Engineering* 138.

— (2020b). “First Principles and Machine Learning Virtual Flow Metering: A Literature Review”. In: *Journal of Petroleum Science and Engineering* 184.

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning.* Springer.

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe (2017). “Variational Inference: A Review for Statisticians”. In: *J. Am. Stat. Assoc.* 112.518, pp. 859–877. DOI: 10.1080/01621459.2017.1285773.

Bottou, L., F.E. Curtis, and J. Nocedal (2018). “Optimization Methods for Large-Scale Machine Learning”. In: *Society for Industrial and Applied Mathematics* 60(2), pp. 223–311.

Cherkassky, V. and F. Mulier (2007). *Learning from data.* John Wiley & Sons.

Corneliussen, Sidsel et al. (2005). *Handbook of multiphase flow metering.* The Norwegian Society for Oil and Gas Measurements.

Emerson Automation Solutions (2017). *Control Valve Handbook.* Fisher Controls International LLC.

Falcone, G., G.F Hewitt, C. Alimonti, and B. Harrison (2013). “Multiphase flow metering: current trends and future developments”. In: *SPE Annual Technical Conference and Exhibition.*

Foss, B. A., B. R. Knudsen, and B. Grimstad (2018). “Petroleum production optimization - A static or dynamic problem”. In: *Computers and Chemical Engineering* 114, pp. 245–253.

Ghahramani, Zoubin (2015). “Probabilistic machine learning and artificial intelligence”. In: *Nature* 521 (7553), pp. 452–459. DOI: 10.1038/nature14541.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning.* Cambridge, Massachusetts. London, England.: The MIT Press.

Granero-Belinchón, C., S. G. Roux, and N. B. Garnier (2019). “Information Theory for Non-Stationary Processes with Stationary Increments”. In: *Entropy* 21.

Gravdahl, T. and O. Egeland (2002). *Modeling and Simulation for Automatic Control.* Marine Cybernetics AS.

Grimstad, B., V. Gunnerud, A. Sandnes, S. Shamlou, I. S. Skrondal, V. Uglane, S. Ursin-Holm, and B. Foss (Sept. 2016). “A Simple Data-Driven Approach to Production Estimation and Optimization”. In: *In: SPE Intelligent Energy International Conference and Exhibition.*

Grimstad, B., M. Hotvedt, A.T. Sandnes, O. Kolbjørnsen, and L. Imsland (2021). “Bayesian Neural Networks for Virtual Flow Metering: An Empirical Study”. In: *Applied Soft Computing* 112 (1).

Guillaume, Dubois (2018). *Modeling and Simulation Challenges and Best Practices for Industry.* CRC Press, Taylor & Francis Group.

Guo, B., W.C. Lyons, and A. Ghalambor (2007). *Petroleum Production Engineering - a computer assisted approach.* Elsevier Science and Technology Books.

Hahn, G. J. (1977). “The Hazards of Extrapolation in Regression Analysis”. In: *Journal of Quality Technology* 9 (4), pp. 159–165. URL: https://doi.org/10.1080/00224065.1977.11980791.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning.* New York, USA: Springer.

Hotvedt, M., B. Grimstad, and L. Imsland (2020). “Developing a Hybrid Data-Driven, Mechanistic Virtual Flow Meter - a Case Study”. In: *IFAC-PapersOnLine* 53 (2), pp. 11692–11697.

— (2021). “Identifiability and physical interpretability of hybrid, gray-box models - a case study”. In: *IFAC-PapersOnLine* 54 (3), pp. 389–394.

— (2022). “Passive learning to address nonstationarity in virtual flow metering applications”. In: *Submitted to Expert Systems with Application for possible publication.*

Hotvedt, M., B. Grimstad, D. Ljungquist, and L. Imsland (2022a). “On gray-box modeling for virtual flow metering”. In: *Control Engineering Practice* 118 (1).

— (2022b). “When is gray-box modeling advantageous for virtual flow metering?” In: *Accepted for publication in IFAC-PapersOnLine.*

Hotvedt, M., S. O. Hauger, F. Gjertsen, and L. Imsland (2019). “Dynamic Real-Time Optimisation of a CO2 Capture Facility”. In: *IFAC-PapersOnLine* 52, pp. 856–861.

Hüllermeier, Eyke and Willem Waegeman (Mar. 2021). “Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods”. In: *Machine Learning* 110.3, pp. 457–506. DOI: 10.1007/s10994-021-05946-3.

International Organization for Standardization (Dec. 1996). *Natural gas - Standard reference conditions.* Standard. International Organization for Standardization.

Jansen, Jan-Dirk (2015). *Nodal Analysis of Oil and Gas Wells - Theory and Numerical Implementation.* TU Delft, The Netherlands: Delft University of Technology.

Johansen, T. A. and B. A. Foss (1992). “Representing and learning unmodeled dynamics with neural network memories”. In: *Proceedings of American Control Conference* 184 (1), pp. 3037–3043.

Kalman, R.E. (1960). “A New Approach to Linear Filtering and Prediction Problems”. In: *Trans. ASME, Journal of Basic Engineering* 82 (Series D), pp. 35–45.

Karpatne, A., G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar (2017). “Theory-guided data science: A new paradigm for scientific discovery from data". In: *arXiv:1612.08544v2*, pp. 1–14.

Kingma, Diederik P. and Jimmy Lei Ba (2015). "Adam: A method for stochastic optimization". In: *3rd International Conference on Learning Representations (ICLR)*, pp. 1–15. eprint: `1412.6980`.

Kittilsen, P., K. Fjalestad, and R. Aasheim (2014). "Stabilized and Increased Well Production Using Automatic Choke Control". In: *Society of Petroleum Engineers.*

Kramer, M. A., M. L. Thomsom, and P. M. Bhagat (1992). "Embedding Theoretical models in neural networks". In: *Proceedings of American Control Conference* 1 (1), pp. 475–479.

Linardatos, P., V. Papastefanopoulos, and S. Kotsiantis (2020). "Explainable AI: A review of machine learning interpretability methods". In: *Entropy* 23 (1), pp. 1–45. URL: `https://doi.org/10.3390/e23010018`.

Liu, X. et al. (2019). "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis." In: *Lancet Digital Health* 1 (6), e271–e297.

Luengo, D., L. Martino, M. Bugallo, V. Elvira, and S. Särkkä (2020). "A survey of Monte Carlo methods for parameter estimation". In: *EURASIP Journal on Advances in Signal Processing* 25. URL: `https://doi.org/10.1186/s13634-020-00675-6`.

Lundin Energy Norway (2020). *Edvard Grieg.* `https://lundin-energy-norway.com/edvard-grieg/`. Acccessed: 18.01.2020.

Mikleš, J. and M. Fikar (2007). *Process modelling, identification, and control.* Springer.

Mishra, Srikanta and Akhil Datta-Gupta (2018). *Applied Statistical Modeling and Data Analytics - A Practical Guide for the Petroleum Geosciences.* Elsevier.

Mohammadmoradi, P., H. M. Moradi, and A. Kantzas (Apr. 2018). "Data-Driven Production Forecasting of Unconventional Wells with Apache Spark". In: vol. Day 4 Wed, April 25, 2018. SPE Western Regional Meeting. D041S011R006. DOI: `10.2118/190098-MS`. eprint: `https://onepetro.org/SPEWRM/proceedings-pdf/18WRM/4-18WRM/D041S011R006/1237730/spe-190098-ms.pdf`.

Monteiro, Danielle D., Gabriela S. Chaves, Virgílo M. Ferreira Filho, and Juliana S. Baioco (2017). "Uncertainty analysis for production forecast in oil wells". In: *SPE Latin American And Caribbean Petroleum Engineering Conference.* DOI: `10.2118/185550-MS`.

Monteiro, Danielle D., Maria Machado Duque, Gabriela S. Chaves, Virgílo M. Ferreira Filho, and Juliana S. Baioco (2020). "Using Data analytics to quantify the impact of production test uncertainty on oil flow rate forecast". In: *IFP Energies Nouvelles* 75 (7), pp. 1–15. DOI: `10.2516/ogst/2019065`.

Moran, M. J., H. N. Shapiro, and M. B. Boettner D. D. Baily (2014). *Fundamentals of Engineering Thermodynamics, 8th. edition*. Wiley.

Nocedal, J. and S.J. Wright (2006). *Numerical Optimization*. Springer.

Oliveira, G., L. Minku, and A. Oliveira (2021). "Tackling Virtual and Real Concept Drifts: An Adaptive Gaussian Mixture Model Approach". In: *IEEE Transactions on Knowledge and Data Engineering*. URL: `doi:%2010.1109/TKDE.2021.3099690.`.

Paszke, Adam et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems* 32, pp. 8026–8037. eprint: `1912.01703`.

Perkins, T.K. (1993). "Critical and sub-critical flow of multiphase mixtures through chokes". In: *Society of Petroleum Engineers*.

Prada, C. de, D. Hose, G. Gutierrez, and Pitarch J.L. (2018). "Developing grey-box dynamic process models". In: *IFAC-PapersOnLine* 51 (2), pp. 523–528.

Psichogios, D.C. and L.H. Ungar (1992). "A Hybrid Neural Network-First Principles Approach to Process Modeling". In: *AIChE Journal* 38(10), pp. 1499–1511.

Pukrittayakamee, A., M. Malshe, M. Hagan, L. M. Raff, R. Narulkar, S. Bukkapatnum, and R. Komanduri (2009). "Simultaneous fitting of a potential-energy surface and its corresponding force fields using feedforward neural networks". In: *Journal of chemical physics* 130. URL: `https://doi.org/10.1063/1.3095491`.

Qin, S. and L. Chiang (2019). "Advances and opportunities in machine learning for process data analytics". In: *Computers and Chemical Engineering* 126, pp. 465–473.

Rackauckas, C., Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman (2021). "Universal Differential Equations for Scientific Machine Learning". In: *arxiv:2001.04385v4*, pp. 1–55.

Al-Rawahi, N., M. Meribout, A. Al-Naamany, Ali Al-Bimani, and A. Meribout (2012). "A neural network algorithm for density measurement of multiphase flow". In: *Multiphase Science and Technology* 24 (2), pp. 89–103.

Roscher, R., B. Bohn, M.F. Duarte, and J. Garcke (2020). "Explainable Machine Learning for Scientific Insights and Discoveries." In: *arXiv:1905.08883v3*, pp. 1–29.

Ruden, T. (2020). *How we obtain accurate VFM flow rates.* https://turbulentflux.com/how-we-obtain-accurate-vfm-flow-rates/. Accessed: 18.01.2020.

Sachdeva, R., Z. Schmidt, J. P. Brill, and R.M. Blais (1986). “Two-phase flow through chokes”. In: *Society of Petroleum Engineers, Annual Technical Conference and Exhibition.*

Al-Safran, E. M. and M. Kelkar (2009). “Predictions of two-phase critical-flow boundary and mass-flow rate across chokes”. In: *Society of Petroleum Engineers* 24 (2).

Sansana, J., M. N. Joswiak, I. Castillo, Z. Wang, R. Rendall, L. H. Chiang, and M. S. Reis (2021). “Recent trends on hybrid modeling for Industry 4.0”. In: *Computers and Chemical Engineering* 151. URL: https://doi.org/10.1016/j.compchemeng.2021.107365.

Schichl, Hermann (2004). “Models and History of Modeling”. In: *Modeling Languages in Mathematical Optimization. Applied Optimization.* Ed. by Josef Kallrath. Vol. 88. Boston MA.: Springer. Chap. 2, pp. 25–36.

Schittkowski, K. (2002). *Numerical Data Fitting in Dynamical Systems, A Practical Introduction with Applications and Software.* Springer, Boston, MA.

Selmer-Olsen, S. (1995). “Subsea chokes as multiphase flowmeters:production control at troll olje”. In: *In: Proceedings of the BHR Group 7th Intl.Conference on Multiphase Production. Cannes, France.*

Shippen, M. (2012). “Steady-State Multiphase Flow - Past, Present, and Future, with a Perspective on Flow Assurance”. In: *Energy and Fuels* 26, pp. 4145–4157.

Solle, D., B. Hitzmann, C. Herwig, P. M Remelhe, S. Ulonska, L. Wuerth, A. Prata, and T. Steckenreiter (2016). “Between the Poles of Data-Driven and Mechanistic Modeling for Process Operation”. In: *Chemie Ingenieur Technik.*

Solution Seeker (2022). *NeuralCompass DD VFM.* https://www.solutionseeker.no/products-and-solutions/flow-rates/neuralcompass-dd-vfm/. Accessed: 12.01.2022.

Staff, G. et al. (2020). “Physics guided machine learning significantly improves outcomes for data-based production optimization”. In: *Society of Petroleum Engineers - Abu Dhabi International Petroleum Exhibition and Conference 2020, ADIP 2020*, pp. 1–16.

Su, H.-T., N. Bhat, P.A. Minderman, and T.J. McAvoy (1992). “Integrating neural networks with first principle models for dynamic modeling”. In: *IFAC Proceedings Volumes* 25 (5), pp. 327–332.

Sutton, R.S. and A.G. Barto (2018). *Reinforcement Learning, an introduction.* Cambridge, Massachusetts, London, England: The MIT Press.

Tan, H. H. and K.H. Lim (2019). "Review of second-order optimization techniques in artificial neural networks backpropagation". In: *IOP Conference Series: Materials Science and Engineering.* Vol. 495, pp. 1–7. URL: `https://doi.org/10.1088/1757-899X/495/1/012003`.

Thorn, R., G. A. Johansen, and B. T. Hjertaker (2013). "Three-phase flow measurement in the petroleum industry". In: *Measurement Science and Technology* 24, pp. 1–17.

Toskey, E.D. (2012). "Improvements to Deepwater Subsea Measurements RPSEA Program: Evaluation of Flow Modeling". In: *Offshore Technology Conference.*

Varyan, R., R.K. Haug, and D.G. Fonnes (2015). "Investigation on the suitability of virtual flow metering system as an alternative to the conventional physical flow meter". In: *SPE, APOGCE*, pp. 2–10.

Willard, J., X. Jia, S. Xu, M. Steinbach, and V. Kumar (2020). "Integrating Physics-Based Modeling With Machine Learning: A Survey". In: *arXiv:2003.04919v4*, pp. 1–34.

Woodward, Andrew M., Bjørn K. Alsberg, and Douglas B. Kell (1998). "The effect of heteroscedastic noise on the chemometric modelling of frequency domain data". In: *Chemometrics and Intelligent Laboratory Systems* 40, pp. 101–107.

Xu, L., W. Zhou, X. Li, and S. Tang (2011). "Wet gas metering using a revised venturi meter and soft-computing approximation techniques". In: *IEE transactions on instrumentation and measurement* 60 (3), pp. 947–956.

Yin, Y., Y. Alqahtani, J.H. Feng, J. Chakraborty, and M. P. McGuire (2021). "Classification of Eye Tracking Data in Visual Information Processing Tasks Using Convolutional Neural Networks and Feature Engineering." In: *SN Computer Science* 2. URL: `https://doi.org/10.1007/s42979-020-00444-0`.

Zendehboudi, S., N. Rezaei, and A. Lohi (2018). "Application of hybrid models in chemical, petroleum, and energy systems: A systematic review". In: *Applied Energy* 228, pp. 2539–2566.

ISBN 978-82-326-6719-2 (printed ver.)
ISBN 978-82-326-5668-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology