Applied Soft Computing 112 (2021) 107776

Contents lists available at ScienceDirect

Applied Soft Computing

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

Bayesian neural networks for virtual flow metering: An empirical study

Bjarne Grimstad [a,b,*], Mathilde Hotvedt [a,b], Anders T. Sandnes [a,c], Odd Kolbjørnsen [c], Lars S. Imsland [b]

[a] Solution Seeker AS, Oslo, Norway
[b] Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway
[c] Department of Mathematics, University of Oslo, Oslo, Norway

ARTICLE INFO

Article history:
Received 30 December 2020
Received in revised form 13 June 2021
Accepted 29 July 2021
Available online 4 August 2021

Keywords:
Neural network
Bayesian inference
Variational inference
Virtual flow metering
Heteroscedasticity

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.

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 [1]. In most production systems, flow rates are measured using well testing. While these measurements are of high accuracy, they are intermittent and infrequent [2]. 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 [3]. 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 [4]. 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 [5]. 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, FlowManager, and Olga [6].

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 [5]. In recent years,

* Corresponding author at: Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway.
E-mail address: bjarne.grimstad@ntnu.no (B. Grimstad).

https://doi.org/10.1016/j.asoc.2021.107776

there has been an increasing number of publications on data-driven VFMs [4]. 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 [7]. Additionally, data-driven VFMs may require less maintenance than a mechanistic VFMs [8]. 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 [9]. 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 [2], 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 [5]. 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 [10].

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 [11]. 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 non-stationary. The pressure in the reservoir decreases as the reservoir is drained and the composition of the produced fluid changes with time [12]. 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 [13]. 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 [14]. 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 [4], 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 [15]. The method has shown promising results in terms of quantifying prediction uncertainty on other problems subject to small datasets and dataset shift [16]. 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 [17–20]. One of the most popular and promising data-driven methods for VFM are neural networks (NN). In [17], 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 timespan of the data covered only 30 h. The three studies, [21–23], 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 [8,10,24,25]. In [24], 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 [8], 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 ($\leq$ 80) in the hidden layer of the network. The study reported an error of 5.9%. In [10,25], 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. [2,17,26–28]. 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 [29]. 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 [30]. 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 [29]. Although probabilistic models and Bayesian inference are well-known in other fields of research, probabilistic VFMs are rare, yet existent [31–34].

The following series of studies, [31–33], 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 [30,35]. 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 [36]. 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 [15]. 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 [15]. 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 [37,38].

3. Flow rate measurements and dataset

A petroleum production well is illustrated in Fig. 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.

The two main devices to measure multiphase flow rates in a well are the multiphase flow meter (MPFM) and test separator, both illustrated in Fig. 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 [39]. 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 [2].

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 [39,40], 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 [41]. 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.

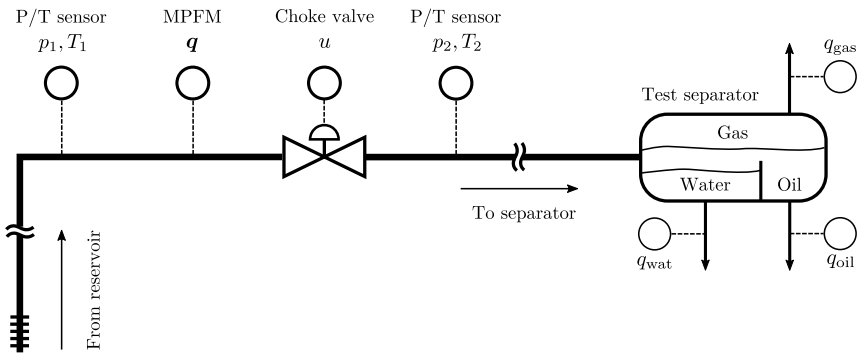
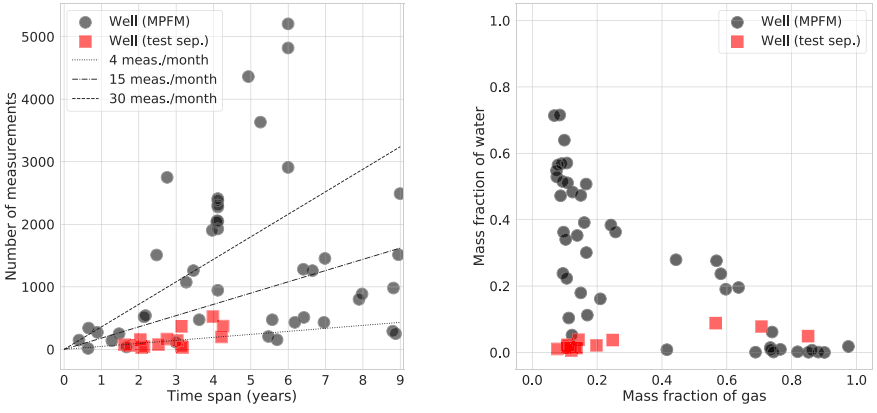


Fig. 1. Sensor placement in a typical production well. A MPFM measures multiphase flow rates in the well. During well testing, single phase flow rates are measured with high accuracy after fluid separation.



(a) Measurement frequency (b) Average mass fractions

Fig. 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).

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 [42]. 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 Fig. 2(a). 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. Fig. 2(b) illustrates the spread in mass fractions among the wells.

In the following, we model the multiphase flow through the production choke valve, a crucial component in any VFM. We consider ideal conditions, in the sense that all measurements required by a reasonable choke model are available [43]. 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 [17].

4. Probabilistic flow model

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

$$\left.\begin{aligned} y_i &= z_i + \epsilon_i \\ z_i &= f(\boldsymbol{x}_i, \boldsymbol{\phi}) \\ s_i &= g(z_i, \boldsymbol{\psi}) \\ \epsilon_i &\sim \mathcal{N}(0, s_i^2) \end{aligned}\right\} i = 1, \ldots, N, \tag{1}$$

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

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

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 [44]. A commonly used form of a hidden

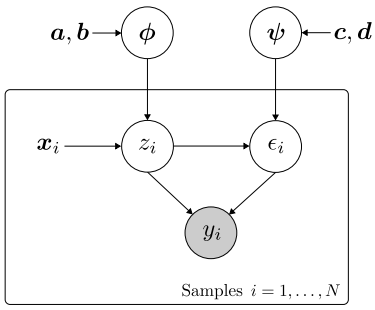


Fig. 3. A probabilistic graphical model for flow rates. Random variables are inscribed by a circle. A gray-filled circle means that the random variable is observed. The dependence $z_i \to \epsilon_i$ indicates that the noise is heteroscedastic, while the dependence $\boldsymbol{\psi} \to \epsilon_i$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

layer l is $f^{(l)}(\boldsymbol{x}) = \text{ReLU}(W^{(l)}\boldsymbol{x} + \boldsymbol{b}^{(l)})$, where the rectified linear unit (ReLU) operator is given as $\text{ReLU}(\boldsymbol{z})_i = \max\{z_i, 0\}$, $W^{(l)}$ 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 $i \in \{1, \dots, K_\phi\}$, 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 Fig. 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

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

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 difficult to capture [45]. 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 $g(z, \boldsymbol{\psi}) > 0$.

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

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

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

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.

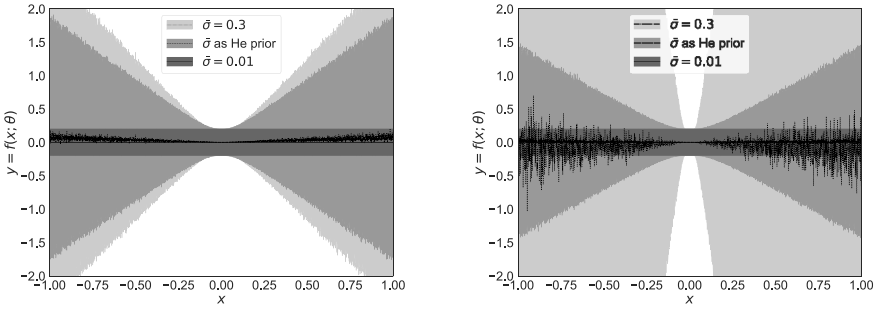
4.2. Prior for the neural network weights, $p(\boldsymbol{\phi})$

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

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

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,

[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.



(a) Two hidden layers, each with 50 nodes (b) Five hidden layers, each with 50 nodes

Fig. 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.

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 [14]. 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 [46,47]. For example, He-initialization [47] 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.

Fig. 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

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

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

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 $p(\boldsymbol{\theta} \mid \mathcal{D})$ after observing data $\mathcal{D}$. The update is performed according to Bayes' rule:

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

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

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

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

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

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

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 [15]. 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 $p(\boldsymbol{\theta} \mid \mathcal{D})$, 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 [14].

5.1. MAP estimation

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

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

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

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

From (15), we see that MAP estimation is equivalent to maximum likelihood estimation with L^2-regularization [30].

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. [15]. Consider a variational posterior density $q(\boldsymbol{\theta} \mid \boldsymbol{\lambda})$, parameterized by a real vector $\boldsymbol{\lambda}$. The objective of the optimization is to find a density $q^\star = q(\boldsymbol{\theta} \mid \boldsymbol{\lambda}^\star)$ that minimizes the Kullback–Leibler (KL) divergence to the exact posterior, i.e.

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

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):

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

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

where the expectation $\mathbf{E}_q[\cdot]$ is taken with respect to $q(\boldsymbol{\theta} \mid \boldsymbol{\lambda})$. 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 $D_{\text{KL}} \left(q(\boldsymbol{\theta} \mid \boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta}) \right)$.

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. [48,49].

Suppose that the variational posterior $q(\boldsymbol{\theta} \mid \boldsymbol{\lambda})$ 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:

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

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:

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

An advantage with the SGVB estimator in (20) is that we can utilize the gradient of the model $\nabla_{\boldsymbol{\theta}} \log p(\mathcal{B} \mid \boldsymbol{\theta})$ as computed by back-propagation. When both the variational posterior and prior are mean-field normals, as is the case for our model, $D_{\text{KL}} \left(q(\boldsymbol{\theta} \mid \boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta}) \right)$ can be computed analytically as shown in Appendix 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 [50] 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 α.

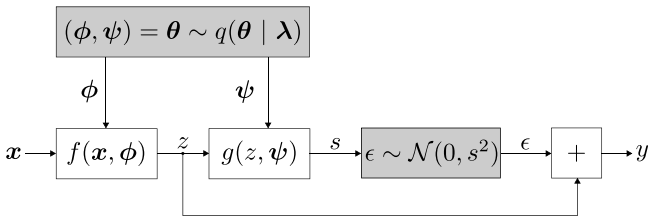
repeat
  Sample mini-batch $\mathcal{B}$ from $\mathcal{D}$
  Sample $\boldsymbol{\zeta} \sim \mathcal{N}(0, I)$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\mu} + \log(1 + \exp(\boldsymbol{\rho})) \circ \boldsymbol{\zeta}$
  Compute $\nabla_{\boldsymbol{\lambda}} \hat{\mathcal{L}}(\boldsymbol{\lambda})$ 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

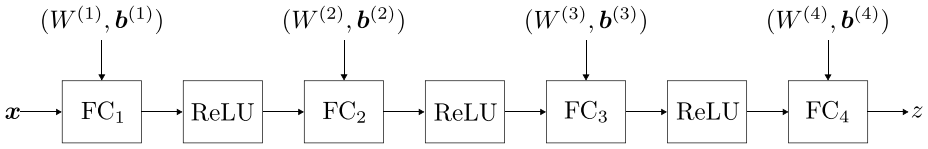
The goal of the case study was to investigate the predictive performance and generalization ability of the proposed VFM. The study was designed to test the predictive performance on historical data and on future data, which reflect the two main applications of a VFM. If the models generalize well, a 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.

A schematic representation of the Bayesian neural network is shown in Fig. 5. The network architecture was fixed to three hidden layers, each with 50 nodes to which we apply the ReLU activation function [51]. Using practical recommendations in [52], the network architecture may be large as long as regularization is used to prevent overfitting. The Adam optimizer [53]



(a) Architecture of BNN



(b) Composition of $f(x, \phi)$

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

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

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

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 [44]. The hyper-parameters were chosen by experimentation and using best practices. The models were implemented and trained using PyTorch [54].

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 Appendix B, Table B.4.

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 Fig. 6. The cumulative performance plot shows the percentage of test points that fall within a certain percent deviation from the actual measurements [39]. 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 Appendix B, Table B.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 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 Fig. 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.

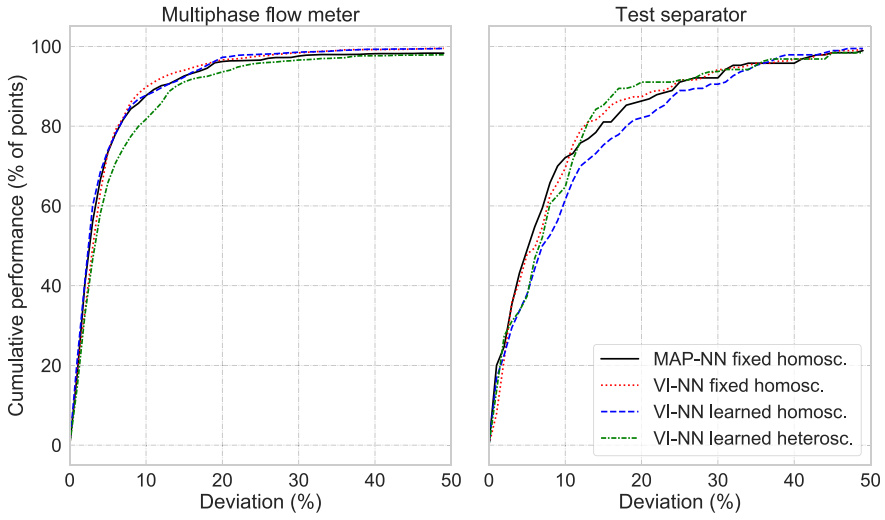


Fig. 6. Cumulative performance of the four models on historical test data. The cumulative performance is shown for wells with (left) MPFM and (right) test separator measurements.

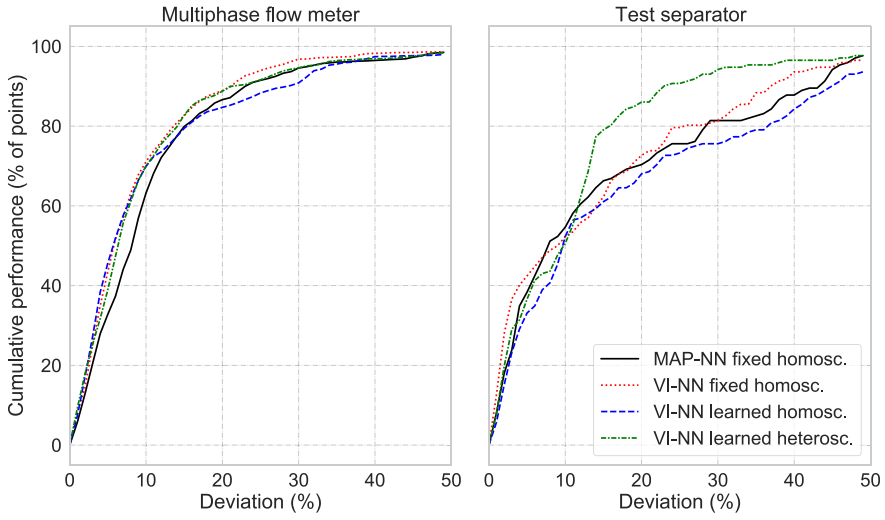


Fig. 7. Cumulative performance of the four models on future test data. The cumulative performance is shown for wells with (left) MPFM and (right) test separator measurements.

6.3. Comparison of performance on historical and future data

A comparison of the MAPEs on historical and future data is illustrated in Fig. 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 Sections 6.1 and 6.2; see Fig. 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).

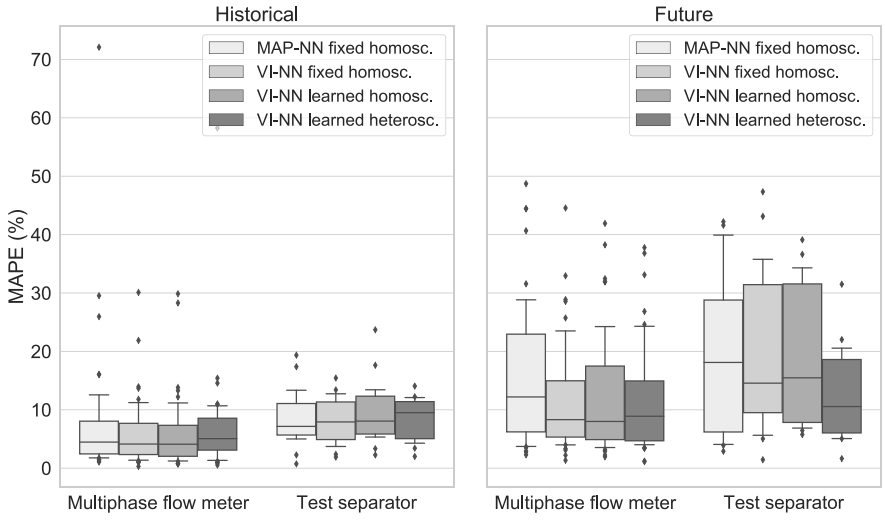
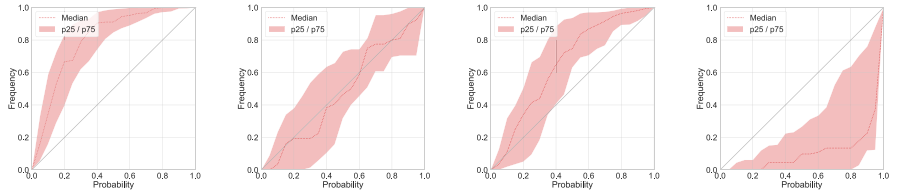
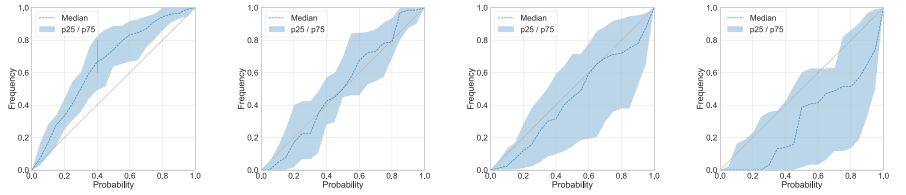


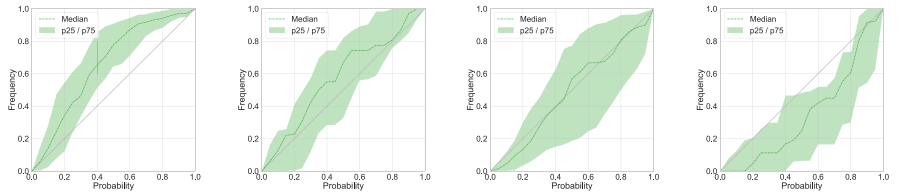
Fig. 8. Comparison of performance on historical and future data for the different models. The box plots differentiate between wells with multiphase flow meter and test separator measurements. The boxes show the P_{25}, P_{50} (median), and P_{75} percentiles. The whiskers show the P_{10} and P_{90} percentiles.



(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

Fig. 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 Sections 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 Fig. 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.

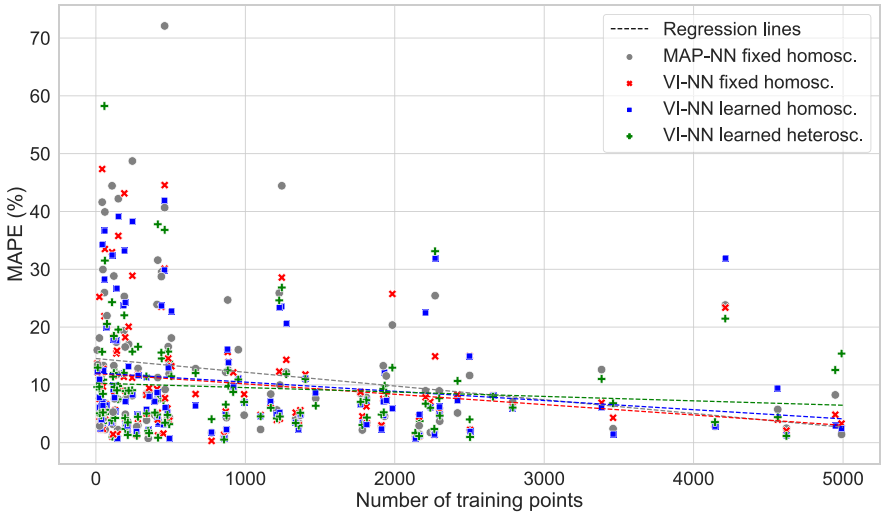


Fig. 10. The plot shows the mean absolute percentage error of the four models on historical and future test data for all wells. A regression line for each model shows the tendency of the error as the number of training points varies.

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

On the other hand, previous studies such as [10], indicate that model performance does not necessarily improve when including data that is several years old. To closer inspect this effect, we compared models developed on successively larger training sets.

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

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

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 Fig. 11.

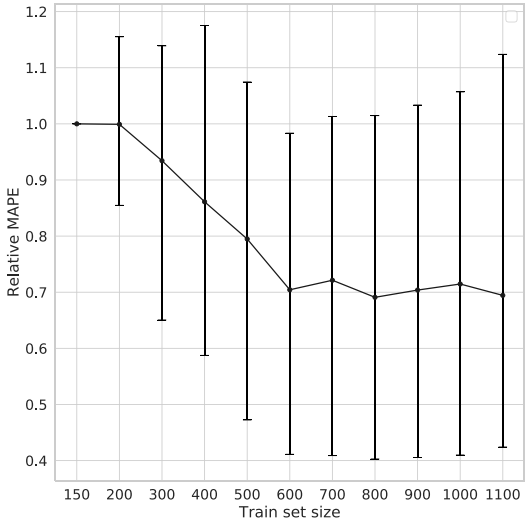


Fig. 11. Relative test errors of the MAP-NN model for increasing training set sizes. Shown are the medians and 50% intervals of 400 trials.

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 Tables 1 and 2 and Fig. 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 Fig. 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 Fig. 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 dependent 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 Fig. 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 Fig. 8, also given in Tables B.4 and B.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 Figs. 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 Fig. 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 [55]. 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.

CRediT authorship contribution statement

Bjarne Grimstad: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original Draft, Visualization, Supervision. **Mathilde Hotvedt:** Formal analysis, Writing – original Draft, Visualization. **Anders T. Sandnes:** Formal analysis, Writing – original draft, Visualization. **Odd Kolbjørnsen:** Validation, Writing – review & editing. **Lars S. Imsland:** Validation, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: The authors Bjarne Grimstad, Mathilde Hotvedt, and Anders T. Sandnes are affiliated with the Norwegian company Solution Seeker AS. The company provides a managed AI service to the petroleum industry. Data-driven VFM is part of this service.

Acknowledgment

This work was supported by Solution Seeker AS.

Appendix 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

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

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

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

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

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:

$$\begin{aligned}&D_{\text{KL}}\,(q(\boldsymbol{\theta} \mid \boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta})) = \mathbf{E}_q\left[\log q(\boldsymbol{\theta} \mid \boldsymbol{\lambda}) - \log p(\boldsymbol{\theta})\right] \\ &= \mathbf{E}_q\left[\sum_{i=1}^{K} \log q(\theta_i \mid \lambda_i) - \log p(\theta_i)\right] \\ &= \frac{1}{2}\mathbf{E}_q\left[\sum_{i=1}^{K} -\log(2\pi\sigma_i^2) - \left(\frac{\theta_i - \mu_i}{\sigma_i}\right)^2 + \log(2\pi\bar{\sigma}_i^2) + \left(\frac{\theta_i - \bar{\mu}_i}{\bar{\sigma}_i}\right)^2\right] \\ &= \frac{1}{2}\left[\sum_{i=1}^{K} -2\log\frac{\sigma_i}{\bar{\sigma}_i} - \frac{1}{\sigma_i^2}\underbrace{\mathbf{E}_{q_i}\left[(\theta_i - \mu_i)^2\right]}_{=\sigma_i^2} + \frac{1}{\bar{\sigma}_i^2}\mathbf{E}_{q_i}\left[(\theta_i - \bar{\mu}_i)^2\right]\right] \\ &= \frac{1}{2}\sum_{i=1}^{K}\left[-1 - 2\log\frac{\sigma_i}{\bar{\sigma}_i} + \frac{1}{\bar{\sigma}_i^2}\mathbf{E}_{q_i}\left[(\theta_i - \bar{\mu}_i)^2\right]\right] \\ &= \frac{1}{2}\sum_{i=1}^{K}\left[-1 - 2\log\frac{\sigma_i}{\bar{\sigma}_i} + \left(\frac{\mu_i - \bar{\mu}_i}{\bar{\sigma}_i}\right)^2 + \left(\frac{\sigma_i}{\bar{\sigma}_i}\right)^2\right]\end{aligned} \tag{A.3}$$

Appendix B. Results

See Tables B.4 and B.5.

Table B.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 B.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

[1] J.-D. Jansen, Nodal Analysis of Oil and Gas Wells - Theory and Numerical Implementation, Delft University of Technology, TU Delft, The Netherlands, 2015.

[2] D.D. Monteiro, M.M. Duque, G.S. Chaves, V.M.F. Filho, J.S. Baioco, Using data analytics to quantify the impact of production test uncertainty on oil flow rate forecast, IFP Energies Nouvelles 75 (2020) 1–15, http://dx.doi.org/10.2516/ogst/2019065.

[3] E. Toskey, Improvements to deepwater subsea measurements RPSEA program: Evaluation of flow modeling, in: Proceedings of the Annual Offshore Technology Conference, 2012, pp. 1–18, http://dx.doi.org/10.4043/23314-MS.

[4] T. Bikmukhametov, J. Jäschke, First principles and machine learning virtual flow metering: A literature review, J. Pet. Sci. Eng. 184 (2020) http://dx.doi.org/10.1016/j.petrol.2019.106487.

[5] D. Solle, B. Hitzmann, C. Herwig, M. Pereira Remelhe, S. Ulonska, L. Wuerth, A. Prata, T. Steckenreiter, Between the poles of data-driven and mechanistic modeling for process operation, Chem. Ing. Tech. 89 (5) (2017) 542–561, http://dx.doi.org/10.1002/cite.201600175.

[6] A. Amin, Evaluation of commercially available virtual flow meters (VFMs), in: Proceedings of the Annual Offshore Technology Conference, 2015, pp. 1293–1318, http://dx.doi.org/10.4043/25764-MS.

[7] K. Balaji, M. Rabiei, V. Suicmez, C.H. Canbaz, Z. Agharzeyva, S. Tek, U. Bulut, C. Temizel, Status of data-driven methods and their application in oil and gas industry, in: EAGE Conference and Exhibition, Society of Petroleum Engineers, 2018, pp. 1–20, http://dx.doi.org/10.2118/190812-MS.

[8] T.A. AL-Qutami, R. Ibrahim, I. Ismail, M.A. Ishak, Radial basis function network to predict gas flow rate in multiphase flow, in: Proceedings of the 9th International Conference on Machine Learning and Computing, 2017, pp. 141–146, http://dx.doi.org/10.1145/3055635.3056638.

[9] S. Mishra, A. Datta-Gupta, Applied Statistical Modeling and Data Analytics - a Practical Guide for the Petroleum Geosciences, Elsevier, 2018.

[10] T.A. AL-Qutami, R. Ibrahim, I. Ismail, Virtual multiphase flow metering using diverse neural network ensemble and adaptive simulated annealing, Expert Syst. Appl. 93 (2018) 72–85, http://dx.doi.org/10.1016/j.eswa.2017.10.014.

[11] M.M. Câmara, R.M. Soares, T. Feital, T.K. Anzai, F.C. Diehl, P. Thomson, J. Pinto, Numerical aspects of data reconciliation in industrial applications, Processes 5 (2017) http://dx.doi.org/10.3390/pr5040056.

[12] B. Foss, B.R. Knudsen, B. Grimstad, Petroleum production optimization – A static or dynamic problem?, Comput. Chem. Eng. 114 (2018) 245–253, http://dx.doi.org/10.1016/j.compchemeng.2017.10.009.

[13] E. Hüllermeier, W. Waegeman, Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods, Mach. Learn. 110 (3) (2021) 457–506, http://dx.doi.org/10.1007/s10994-021-05946-3.

[14] Y. Gal, Uncertainty in Deep Learning (Ph.D. thesis), University of Cambridge, 2016, p. 174.

[15] D.M. Blei, A. Kucukelbir, J.D. McAuliffe, Variational inference: A review for statisticians, J. Amer. Statist. Assoc. 112 (518) (2017) 859–877, http://dx.doi.org/10.1080/01621459.2017.1285773.

[16] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J.V. Dillon, B. Lakshminarayanan, J. Snoek, Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift, in: 33rd Conference on Neural Information Processing Systems, 2019, arXiv:1906.02530.

[17] G. Zangl, R. Hermann, S. Christian, Comparison of methods for stochastic multiphase flow rate estimation, in: SPE Annual Technical Conference and Exhibition, Vol. 12, 2017, http://dx.doi.org/10.2118/170866-MS.

[18] O. Bello, S. Ade-Jacob, K. Yuan, Development of hybrid intelligent system for virtual flow metering in production wells, in: SPE Intelligent Energy Conference & Exhibition, Society of Petroleum Engineers, 2014, http://dx.doi.org/10.2118/167880-MS.

[19] L. Xu, W. Zhou, X. Li, S. Tang, Wet gas metering using a revised venturi meter and soft-computing approximation techniques, IEEE Trans. Instrum. Meas. 60 (2011) 947–956, http://dx.doi.org/10.1109/TIM.2010.2045934.

[20] T. Bikmukhametov, J. Jäschke, Oil production monitoring using gradient boosting machine learning algorithm, in: 12th IFAC Symposium on Dynamics and Control of Process Systems, Including Biosystems, 52, 2019, pp. 514–519, http://dx.doi.org/10.1016/j.ifacol.2019.06.114, IFAC-PapersOnLine.

[21] S.M. Berneti, M. Shahbazian, An imperialist competitive algorithm - artificial neural network method to predict oil flow rate of the wells, Int. J. Comput. Appl. 26 (2011) 47–50, http://dx.doi.org/10.5120/3137-4326.

[22] M.A. Ahmadi, M. Ebadi, A. Shokrollahi, S.M.J. Majidi, Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir, Appl. Soft Comput. 13 (2013) 1085–1098, http://dx.doi.org/10.1016/j.asoc.2012.10.009.

[23] M.Z. Hasanvand, S. Berneti, Predicting oil flow rate due to multiphase flow meter by using an artificial neural network, Energy Sour. A 37 (2015) 840–845, http://dx.doi.org/10.1080/15567036.2011.590865.

[24] T.A. AL-Qutami, R. Ibrahim, I. Ismail, M.A. Ishak, Development of soft sensor to estimate multiphase flow rates using neural networks and early stopping, Int. J. Smart Sensing Intell. Syst. 10 (2017) 199–222, http://dx.doi.org/10.21307/ijssis-2017-209.

[25] T.A. AL-Qutami, R. Ibrahim, I. Ismail, M.A. Ishak, 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, 2017, pp. 304–309, http://dx.doi.org/10.1109/ICSIPA.2017.8120626.

[26] H.P. Bieker, O. Slupphaug, T.A. Johansen, Well management under uncertain gas or water oil ratios, in: SPE Digital Energy Conference and Exhibition, Society of Petroleum Engineers, 2007, http://dx.doi.org/10.2118/106959-MS.

[27] J.R. Fonseca, M. Gonçalves, L. Azevedo, 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, 2009.

[28] D.D. Monteiro, G.S. Chaves, V.M.F. Filho, J.S. Baioco, Uncertainty analysis for production forecast in oil wells, in: SPE Latin American and Caribbean Petroleum Engineering Conference, 2017, http://dx.doi.org/10.2118/185550-MS.

[29] Z. Ghahramani, Probabilistic machine learning and artificial intelligence, Nature 521 (2015) 452–459, http://dx.doi.org/10.1038/nature14541.

[30] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, New York, USA, 2009, http://dx.doi.org/10.1007/978-0-387-84858-7.

[31] R.J. Lorentzen, A.S. Stordal, G. Nævdal, H.A. Karlsen, H.J. Skaug, Estimation of production rates with transient well-flow modeling and the auxiliary particle filter, Soc. Pet. Eng. 19 (1) (2014) 172–180, http://dx.doi.org/10.2118/165582-PA.

[32] R.J. Lorentzen, A.S. Stordal, X. Luo, G. Nævdal, Estimation of production rates by use of transient well-flow modeling and the auxiliary particle filter: Full-scale applications, SPE Prod. Oper. 31(2) (2016) 163–175, http://dx.doi.org/10.2118/176033-PA.

[33] X. Luo, R.J. Lorentzen, A.S. Stordal, G. Nævdal, Toward an enhanced Bayesian estimation framework for multiphase flow soft-sensing, Inverse Problems 30 (2014) http://dx.doi.org/10.1088/0266-5611/30/11/114012.
[34] N. Bassamzadeh, R. Ghanem, Probabilistic data-driven prediction of well-bore signatures in high-dimensional data using Bayesian networks, Soc. Pet. Eng. (2018) http://dx.doi.org/10.2118/189966-PA.
[35] N.G. Polson, V. Sokolov, Deep learning: A Bayesian perspective, Bayesian Anal. 12 (4) (2017) 1275–1304, http://dx.doi.org/10.1214/17-BA1082.
[36] J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian optimization of machine learning algorithms, in: Proceedings of the 25th International Conference on Neural Information Processing Systems, Vol. 25, 2012, pp. 2951–2959.
[37] Y. Liu, Q. Liu, W. Wang, J. Zhao, H. Leung, Data-driven based model for flow prediction of steam system in steel industry, Inform. Sci. 193 (2012) 104–114, http://dx.doi.org/10.1016/j.ins.2011.12.031.
[38] G.B. Humphrey, M.S. Gibbs, G.C. Dandy, H.R. Maier, A hybrid approach to monthly streamflow forecasting: Integrating hydrological model outputs into a Bayesian artificial neural network, J. Hydrol. 540 (2016) 623–640, http://dx.doi.org/10.1016/j.jhydrol.2016.06.026.
[39] S. Corneliussen, J.-P. Couput, E. Dahl, E. Dykesteen, K.-E. Frøysa, E. Malde, H. Moestue, P.O. Moksnes, L. Scheers, H. Tunheim, Handbook of Multiphase Flow Metering, The Norwegian Society for Oil and Gas Measurements, 2005.
[40] C. Marshall, A. Thomas, Maximising economic recovery - a review of well test procedures in the North Sea, in: Offshore Europe Conference and Exhibition, Society of Petroleum Engineers, 2015.
[41] K. Krejbjerg, N. Lindeloff, H. Berentsen, V.R. Midttveit, Conversion of multiphase meter flowrates, The Norwegian Society for Oil and Gas Measurements (NFOGM), 2019.
[42] B. Grimstad, V. Gunnerud, A. Sandnes, S. Shamlou, I.S. Skrondal, V. Uglane, S. Ursin-Holm, B. Foss, A simple data-driven approach to production estimation and optimization, in: SPE Intelligent Energy International Conference and Exhibition, 2016, http://dx.doi.org/10.2118/181104-MS.
[43] M. Hotvedt, B. Grimstad, L. Imsland, Developing a hybrid data-driven, mechanistic virtual flow meter - a case study, IFAC-PapersOnLine 53 (2) (2020) 11692–11697, http://dx.doi.org/10.1016/j.ifacol.2020.12.663, 21th IFAC World Congress.
[44] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[45] A.M. Woodward, B.K. Alsberg, D.B. Kell, The effect of heteroscedastic noise on the chemometric modelling of frequency domain data, Chemometr. Intell. Lab. Syst. 40 (1998) 101–107, http://dx.doi.org/10.1016/S0169-7439(97)00078-6.
[46] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, 9, 2010, pp. 249–256.
[47] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034, http://dx.doi.org/10.1109/ICCV.2015.123.
[48] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, in: 2nd International Conference on Learning Representations (ICLR), 2014, arXiv:1312.6114.
[49] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural networks, in: 32nd International Conference on Machine Learning, 37, 2015, pp. 1613–1622, arXiv:1505.05424.
[50] D.P. Kingma, T. Salimans, M. Welling, Variational dropout and the local reparameterization trick, in: 28th International Conference on Neural Information Processing Systems, Vol. 2, 2015, pp. 2575–2583.
[51] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Vol. 15, 2011, pp. 315–323.
[52] Y. Bengio, Practical recommendations for gradient-based training of deep architectures, in: Neural Networks: Tricks of the Trade, Springer, 2012, pp. 437–478, http://dx.doi.org/10.1007/978-3-642-35289-8{_}26.
[53] D.P. Kingma, J.L. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations (ICLR), 2015, pp. 1–15, arXiv:1412.6980.
[54] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. 32 (2019) 8026–8037, arXiv:1912.01703.
[55] F. Wenzel, K. Roth, B.S. Veeling, J. Świątkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, S. Nowozin, How good is the Bayes posterior in deep neural networks really? in: Proceedings of the 37th International Conference on Machine Learning, 119, 2020, pp. 10248–10259, arXiv:2002.02405.