

A Benchmark Data Set for Two-Phase Coriolis Metering

Olga L. Ibryaeva, Vyacheslav V. Barabanov

South Ural State University, Russia

Manus P. Henry, Michael Tombs, Feibiao Zhou

University of Oxford, UK

Abstract

For more than a decade there has been growing interest in the use of Coriolis mass flow metering applied to two-phase (gas/liquid) and multiphase (oil/water/gas) conditions. It is well-established that the mass flow and density measurements generated from multiphase flows are subject to large errors, and a variety of physical models and correction techniques have been proposed to explain and/or to compensate for these errors. One difficulty is the absence of a common basis for comparing correction techniques, because different flowtube designs and configurations, as well as liquid and gas properties, may result in quite different error curves. Furthermore, some researchers with interests in the modelling aspects of the field may not have suitable multiphase laboratory facilities to generate their own data sets. This paper offers a small data set that may be used by researchers as a benchmark i.e. a common data set for comparing correction techniques. The data set was collected at the UK National Flow Laboratory TUV-NEL, using air and a viscous oil, and provides experimental points over a wide flow range (8:1 turndown) and with Gas Volume Fraction (GVF) values up to 60%. As a first investigation using the benchmark data set, we consider how data sparsity (i.e. the flow rate and GVF spacing in the experimental grid) affects the accuracy of a correction model. A range of neural network models are evaluated, based on different subsets of the benchmark data set. The data set and some exemplary code are provided with the paper. Additional data sets are available on a web site created to support this initiative.

Keywords

Coriolis mass flow metering, two-phase flow, multiphase flow, benchmark data set, neural networks, data driven model.

1. Introduction

For more than a decade there has been growing interest in the use of Coriolis mass flow metering applied to two-phase (gas/liquid) and multiphase (oil/water/gas) conditions [1]. It is well-established that the mass flow and density measurements generated under multiphase flow conditions are subject to large errors. Arising from this general observation are a number of more specific research topics. These include: developing more robust signal tracking algorithms to operate during two-phase conditions (e.g. [2, 3]), understanding the physical causes of the mass flow and density errors (e.g. [4]), and developing methods to correct these errors (e.g. [5], [6]).

As techniques have improved, the use of Coriolis metering for two-phase and multiphase flows has become increasingly ambitious. In most applications, the respective densities of the liquid and gas in the mixture are such that the mass flow of the gas is effectively negligible. In early work, the intention was therefore simply to correct the mass flow of the liquid component to compensate for the errors induced by the presence of the gas [10]. Subsequently, it was noted that the GVF, if estimated, could be used to provide a volumetric gas flow measurement, so that with local pressure and temperature information, it would be possible to calculate the gas mass flow rate too. Further sophistication was introduced with the development of three-phase flow metering [9], where the mass flows of oil, water and gas are all calculated, by employing additional instrumentation. Coriolis metering has also been extended to wet gas applications (e.g. [30]), where $GVF > 95\%$ so that the initial premise, that the gas mass flow rate is negligible compared to that of the liquid, no longer holds. However, irrespective of the sophistication of the final calculation, a common requirement is to provide corrections for the mass flow and density values generated by the Coriolis meter when presented with a gas/liquid mixture, and this basic correction problem is the focus of the current paper.

The prior literature (surveyed in [4]) has demonstrated that there are a variety of physical causes for the mass flow and density errors, and that these interact in complex ways. In addition, the Coriolis flowtube geometry has a strong influence on the errors (as illustrated in [7]). However, flowtube geometry varies significantly between meter designs, and creating a universal Coriolis meter two-phase error model which accounting for geometric features would appear to be a formidable challenge. Two consequences arise from these observations. Firstly, there is no immediate prospect of a purely physical model being developed which is capable of providing mass flow and density error corrections over a comprehensive range of flowtube designs and process conditions. A corollary of this point is that so-called ‘soft’ computing techniques ([5]), based primarily on empirical data rather than physical models, are likely to remain the primary means of providing Coriolis meter multiphase flow corrections for the foreseeable future. The

second consequence is that the variety of measurement error behaviours, arising from differences in flowtube geometries, fluid properties etc., make direct comparisons between soft computing techniques difficult, as the experimental data sets produced by different researchers often have limited common features. A further constraint on future development is the fact that some researchers with interests in soft computing techniques may not have suitable multiphase laboratory facilities to generate their own data sets.

Accordingly, this paper provides a Coriolis meter two-phase flow data set that may be used by other researchers as a benchmark i.e. a common data set for comparing correction techniques. The specific data set chosen has a number of useful features:

- The data set was collected at the UK National Flow Laboratory TUV-NEL, so the results include traceable reference measurements.
- The specific combination of flowtube geometry, flowtube orientation, and choice of fluids (an oil with moderately high viscosity, and nitrogen gas) generates repeatable mass flow and density errors over a wide range of liquid flow rates and GVFs. The flow range covers an 8:1 turndown while GVF values range up to 60%. Note that the turndown ratio, i.e. the range of accurate operation for a Coriolis meter working with a pure liquid, is typically 100:1 or higher. However, with two-phase flows, non-repeatability become a significant problem at low flows (caused by the build-up of random pockets of gas and/or liquid inside the meter), and so the working range for two-phase flow performance may be only 4:1. Hence the 8:1 turndown ratio provided here is a relatively wide range for two-phase flow data.
- The mass flow and density errors are a relatively ‘easy’ data set to model – there are no discontinuities or sudden changes of gradient in the data. Implicit in this remark is an important reminder that this data set only represents the behavior of the selected flowtube, fluids etc. A two-phase flow correction *model* which delivers good performance over this data set will not be generally applicable to other meter designs and application conditions. However, it is to be hoped that *modelling techniques* generating effective correction models for this data set may also prove useful when applied to other data sets representing other flowtubes and application conditions.

A website has been created to make additional data sets available [31]. At the time of writing, two further two-phase data sets are provided, both from experimental work carried out at NEL. One contains data for (salty) water and nitrogen on a 12 mm Foxboro flowtube, while the second data set for a conventional (i.e. low viscosity) oil and nitrogen using the 50mm Foxboro flowtube design. It is planned that further data sets will be made available on the website, including from

other manufacturers' flowtubes. The interested reader is referred to the website for additional information on current and future data sets, which are not discussed further in this paper. Future papers may offer analysis of more challenging benchmark data sets, including data for three-phase (oil/water/gas) flow.

Also in this paper, the potential utility of the data is illustrated by the presentation of a modelling investigation which considers the trade-off between *data sparsity* (i.e. the spacing between experimental data points) and the resulting neural net (NN) based model accuracy. In the investigation, we consider various subsets of the benchmark data set, fit a range of NN models to each subset, and evaluate the resulting model accuracy. The potential application of this study is to assist experimental design (whether in the lab or the field) when considering the GVF and mass flow steps that should be taken in order to achieve a desired level of accuracy. The study further provides quantitative results for various NN model fitting techniques applied to this data set; other researchers are invited to apply alternative soft computing techniques to this data set and compare the resulting modelling accuracies.

Section 2 describes the benchmark data set. Section 3 provides an introduction to the exploration of data sparsity and model accuracy. Section 4 describes the NN modelling techniques employed. Section 5 provides the results obtained.

2. The Benchmark Data Set

The benchmark data set is a subset of the data used in [8]; this describes a trial at TUV-NEL in which two-phase flow data is collected using nitrogen gas and a synthetic oil over a range of oil viscosities, where the viscosity is adjusted by varying the liquid temperature. The Coriolis meter used is a 50 mm diameter Foxboro CFS-10 flowtube and CFT-51 transmitter, incorporated into the Net Oil and Gas skid arrangement. The skid design is described in [9]. The flowtube orientation, skid geometry, and a flow straightener are all arranged to improve the repeatability of the skid measurement for two-phase and multiphase flow.

Reference measurements for the liquid flow rate were traceable with uncertainty of 0.05 % ($k = 2$). A flow control loop adjusted pump speed to maintain the desired liquid flowrate despite changes in GVF (and associated variation in back pressure). Reference measurements for the gas flow rate were traceable with an uncertainty of 0.5% ($k = 2$). The second experimental value to be controlled is the GVF. Accordingly, temperature and pressure measurements at the flowtube entrance were used to calculate the local GVF, and then the gas mass flow rate was adjusted to maintain the GVF at the desired level. Allowing for uncertainty in the temperature and pressure measurements and hence the local nitrogen gas density, the *relative* uncertainty in the GVF

estimate is 1% of reading. Hence, if the estimated GVF is 2% then the true GVF is expected to lie within the range $2 (\pm 0.02) \%$ with 95% confidence, while if the estimated GVF is 50% then the GVF is expected to lie within the range $50 (\pm 0.5) \%$ at the same confidence level. Given the magnitude of the raw mass flow and density errors, and indeed the likely range of residual errors after correction, the uncertainty in the reference measurements is small and for our own analyses has been considered negligible.

The benchmark data is taken from results collected at $15 \pm 1.5 \text{ }^{\circ}\text{C}$, with corresponding oil viscosity $450 (\pm 30) \text{ cP}$, the line pressure $345 (\pm 5) \text{ kPa}$ and the oil density $881 (\pm 1) \text{ kg/m}^3$. While these parameters are provided as background information (and further details will be provided by the authors upon request) only the oil density value (and its small range over the data set) are of direct relevance for the purposes of developing soft computing models.

In most cases, data is based on average values collected over 120 s, where measurement updates were provided at an interval of 1s. In a small number of experiments, 300 s of data was collected and averaged. This longer collection period led to no observed change in behavior, and so these longer experiments are not identified within the data set.

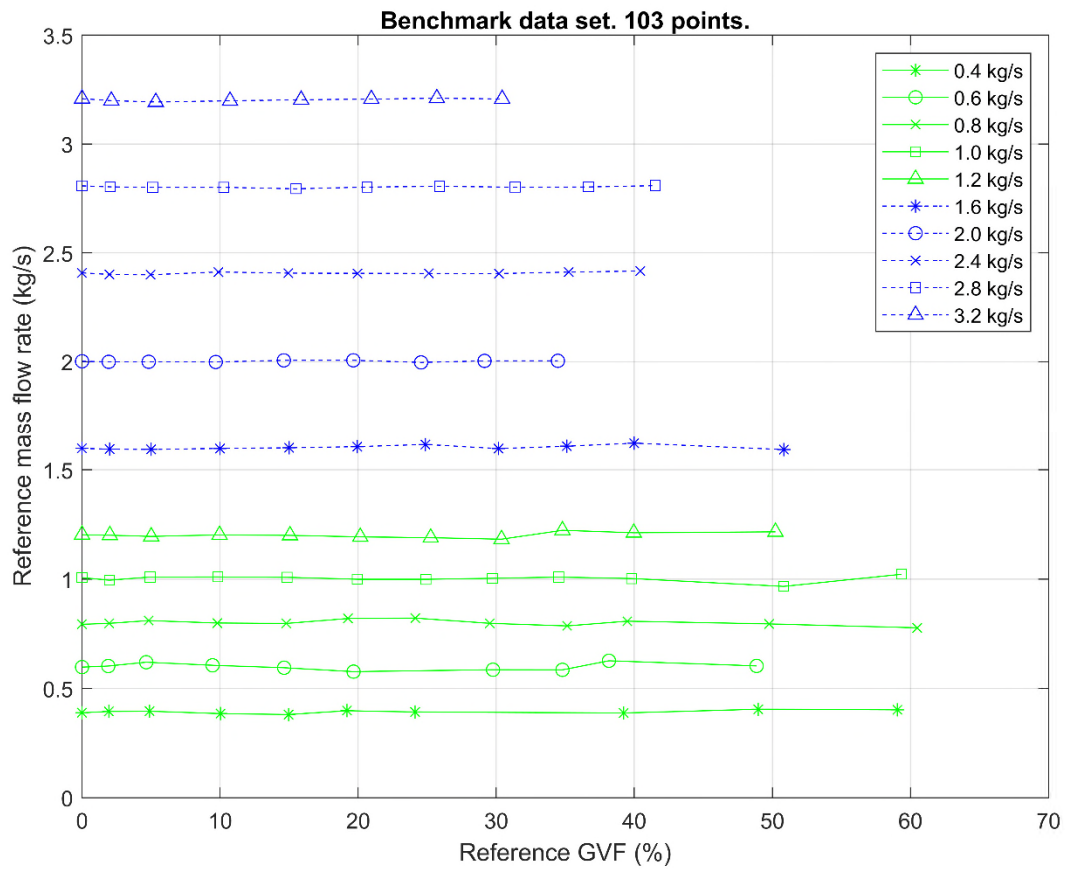


Figure 1: Reference mass flow rates and GVF values for benchmark data set

Figure 1 shows the flow rates and GVF values for the 103 data points in the benchmark set. The data set consists of ten flow lines, at approximately 0.4 kg/s, 0.6 kg/s, 0.8 kg/s, 1.0 kg/s, 1.2 kg/s, 1.6 kg/s, 2.0 kg/s, 2.4 kg/s, 2.8 kg/s, and 3.2 kg/s, covering an 8:1 turndown ratio. Note that these flow rates are relatively low for the 50 mm diameter flowtube, which has a nominal maximum flow rate of 13 kg/s. This is explained by consideration of pressure drop: the 13 kg/s nominal capacity is based on a limit of 100 kPa pressure drop across the flowtube, and assumes the liquid has the density and viscosity of water. In these experimental results, liquid viscosity is significantly higher, and the introduction of gas volume adds to the already high pressure drops. In our multiphase flow work we accept a maximum pressure drop across the flowtube of 150 kPa, and this is reflected in the range of flow rate and GVF points in the data set. In particular, at the higher flow rates the highest GVF value is limited to avoid exceeding 150 kPa pressure drop across the flowtube, and the maximum flowrate is limited to 3.2 kg/s.

The steps between flow lines are smaller (0.2 kg/s steps) at the low flow rates and larger (0.4 kg/s steps) at the higher flowrates. From experience, there is usually greater variation in mass flow and/or density errors at lower flow rates, and so it is appropriate to collect more experimental data in the low flow regions. This greater variation in errors at low flow rates is indeed present in this data set, as described below.

The reference GVF values for each flow line are (approximately) as follows: 0%, 2%, 5%, 10%, 15%, 20%, ... and then in steps of 5% up to a maximum of 60%, or until the pressure drop exceeds 150 kPa. Some experimental points are excluded (e.g. at 0.4 kg/s and 30% GVF) where the data quality was not considered adequate.

We now introduce the density and mass flow error data for the benchmark set. Since the earliest investigations into two-phase flow Coriolis metering (e.g. [10]), density drop (dd) – as defined below - has been used as a parameter for characterizing the meter response. However, its use implies knowledge of the true (gas free) liquid density. In practice, for simple ‘two-phase’ applications involving a single liquid and a gas, this is usually available, and can be provided by the user as a configuration parameter (possibly including a temperature coefficient). Constraining the true liquid density to a known value is required in order to be able to interpret the observed density as a function of the gas content: if both the liquid density and GVF are unknown, then there are many possible combinations of these two variables that could generate a given density value from the flow meter.

The density drop dd is given by the percentage difference between the liquid-only mixture density ρ_l and the observed density ρ_{obs} :

$$dd = \frac{\rho_l - \rho_{obs}}{\rho_l} \cdot 100\%. \quad (1)$$

Note that the density drop calculation is arranged so that its value is always zero or positive, even though the observed density drops in value as dd increases. As shown below, this convention is convenient when plotting data where dd is used as the x-axis variable.

With gas present in the liquid, a density drop is (almost) always observed, and this drop in value can be broadly partitioned into two components: the ‘true’ density drop attributable to the true densities of the gas and liquid, together with their relative volumetric proportions present in the flowtube; and the density drop ‘error’ arising from the two-phase flow errors introduced into the density measurement. If it is assumed that the gas density is negligible, that there is no slip (i.e. the gas and liquid move at the same velocity), and that the true gas/liquid mixture density is known, then the GVF is approximately equal to the ‘true’ density drop:

$$GVF \sim dd_{true} = \frac{\rho_l - \rho_{true}}{\rho_l} \cdot 100\%. \quad (2)$$

where ρ_{true} is the true density of the liquid/gas mixture.

Accordingly, while outside the laboratory the true GVF is unknown and to be estimated via modelling, the density drop is always available via the observed density measurement. It therefore serves as a useful dimensionless parameter for characterizing two-phase flow, and is widely used in Coriolis meter correction models.

The mass flow error \dot{m}_{err} and the density error ρ_{err} are calculated as follows:

$$\dot{m}_{err} = \frac{\dot{m}_{obs} - \dot{m}_{true}}{\dot{m}_{true}} \cdot 100\%. \quad (3)$$

$$\rho_{err} = \frac{\rho_{obs} - \rho_{true}}{\rho_{true}} \cdot 100\%. \quad (4)$$

where \dot{m}_{obs} and ρ_{obs} are the observed mass flow and density measurements from the Coriolis meter, and \dot{m}_{true} is the true (reference) mass flow rate of the gas/liquid mixture. Figures 2 and 3 show the \dot{m}_{err} and ρ_{err} against (observed) dd for the ten flow lines in the benchmark data set.

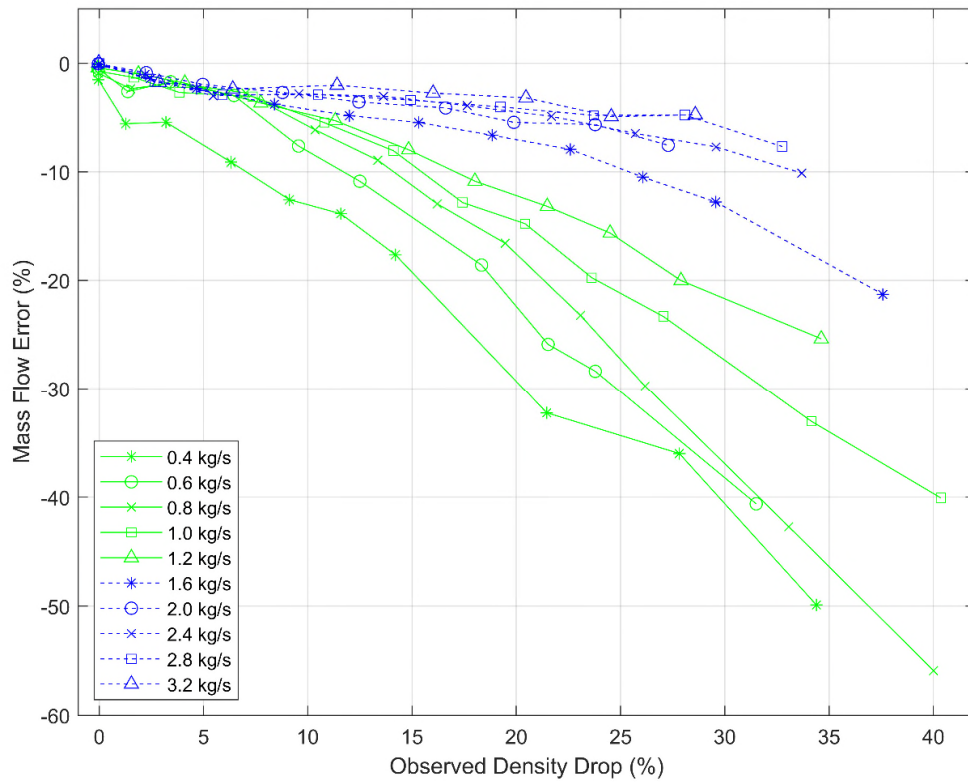


Figure 2: Mass flow error against density drop

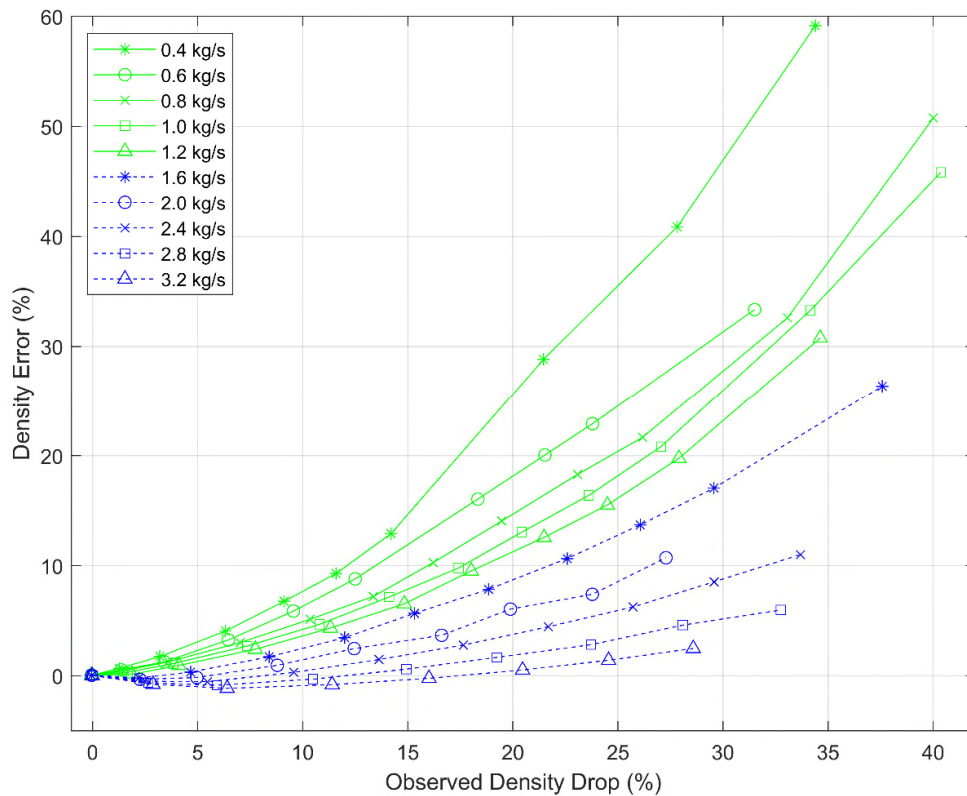


Figure 3: Density error against density drop

Both the mass flow and the density errors are a function of the density drop as well as the true mass flow rate. At low flow rates, both the mass flow and density errors have both higher

magnitude and higher rates of change. This is commonly observed with Coriolis meter two-phase data, and hence the practice of using smaller steps between flow lines points at lower flowrates. The mass flow errors are also fairly typical of a ‘bent’ flowtube design, reaching approximately -60% for low flowrate and high GVF.

The density errors presented here are less typical: these are positive, whereas more often for a flowtube of this type they would be negative ([7], [9]). The positive density errors in this case are attributable to two factors. Firstly, the high viscosity of the oil reduces the magnitude of the negative errors due to the ‘bubble effect’ [4]. Secondly, slip occurs between the liquid and gas phases occur, with the gas travelling at a faster velocity through the flowtube than the liquid, so that the effective gas volume fraction is lower than the nominal value assuming no slip. For clarity, in all cases with gas present the reported density is lower than that of the pure oil, but is higher than the reference mixture density for the given proportions of oil and nitrogen, assuming no slip between the phases. For low flow and high GVF, these errors reach approximately +60%. Note also the interaction between the density errors and the scaling of the density drop. While the true GVF (Figure 1) reaches 60%, the observed density drop (Figures 2 & 3) only reaches 40%, due to the positive density errors (Figure 3).

A useful change of variables that assists the numerical properties of a density correction algorithm is to calculate the ‘density drop error’. As stated above, if it is assumed that the gas density is negligible and there is no slip, then the true density drop (dd_{true} in equation (2)) is equal to the GVF. Accordingly, we may define the density drop error $dd_e = dd - dd_{true}$ as being the difference between the true and observed density drops. Figure 4 shows the density drop error against density drop for the benchmark data set. Note here that the density drop error is the *absolute error* (in percent) and not a *relative error* scaled by the true density drop – the latter calculation would be subject to high levels of noise where the true density drop is small.

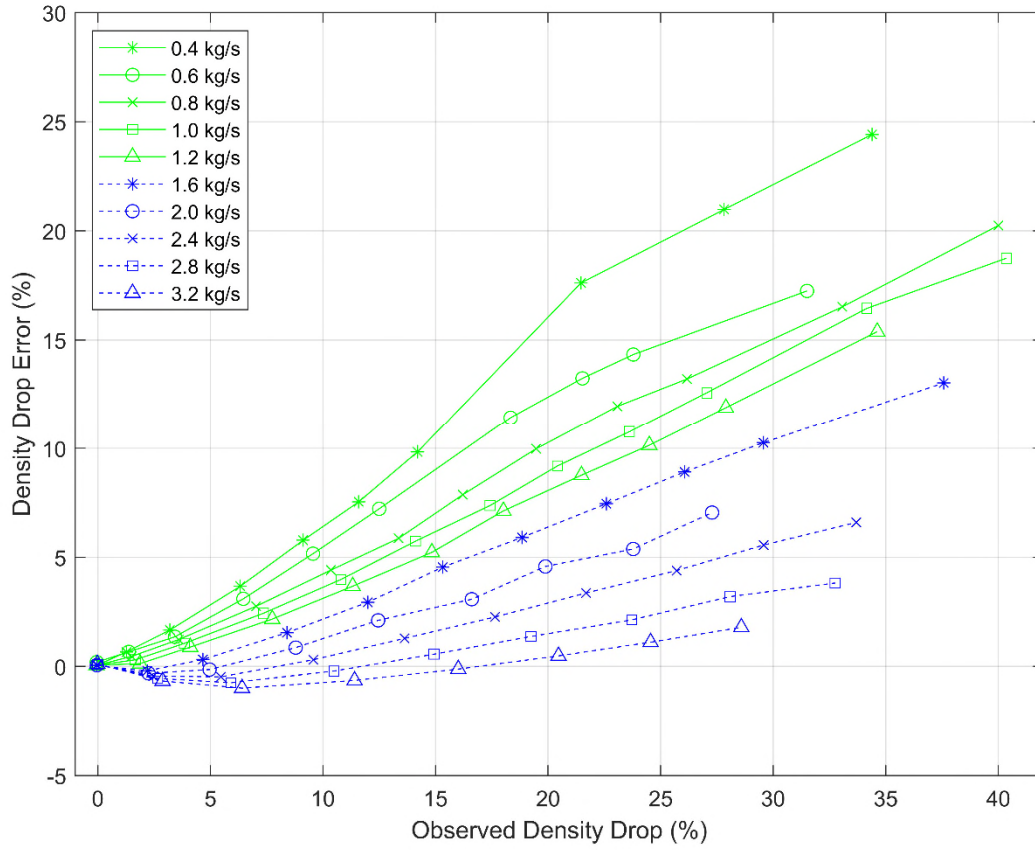


Figure 4: Density drop error against density drop

The density drop error has lower magnitude than the density error, typically resulting in lower residual errors after modelling. It is straightforward to calculate a corrected density drop, and hence GVF and true mixture density, if the density drop error is modelled in this way.

The full data set (see Appendix 1) is available in spreadsheet form, consisting of the following columns of data: nominal mass flow rate; reference mass flow rate, mixture density and GVF; Coriolis meter measurements for mass flow rate and density, and the corresponding density drop; and finally the Coriolis meter errors for mass flow, density, and density drop.

A typical two-phase flow correction scheme would entail developing models of the mass flow error and of the density error (or equivalently the density drop error), based on the Coriolis meter mass flow measurement and density drop. This is the basis for the modelling investigation described in the remainder of the paper, which considers the trade-off between data sparsity and the resulting neural net (NN) based model accuracy.

3. Benchmark investigation

Artificial neural networks have been successfully applied to provide corrected mass flow and density measurements in Coriolis flowmeters for two-phase or multiphase flow applications [7, 9,

10]. Support Vector Machine (SVM) and genetic programming [6], as well as hybrid models [5], have also been applied. For any of these machine learning methods, a data set is required; for a given flowtube design, fluid properties etc there will be a corresponding operating range over which the data set is to be collected. As a general principle, reduced spacing between data points should yield a better model. However, collecting modelling data is expensive: multiphase calibration facilities charge high daily rates, while obtaining reference measurements of the true liquid and gas flowrates in the field (for example by using a portable well test system) is likely to be slow and costly. A number of questions naturally arise: what is the tradeoff between data sparsity and model accuracy? Do certain modelling techniques perform advantageously with lower data densities than others, resulting in good solutions at lower cost? The benchmark data set has been used to perform a simple investigation of these questions. A variety of neural net modelling approaches has been applied to subsets of the benchmark data and the resulting modelling accuracy compared.

It is important to emphasize once again that any conclusions drawn are at least in part dependent upon the mathematical properties of the benchmark error curves, which in turn depend upon the choice of flowtube, fluids etc. Nevertheless, the benchmark provides a common data set for exploring the impact of data sparsity upon model accuracy, and we invite other researchers to report on alternative modelling approaches applied to the same benchmark dataset, including consideration of different data subsets.

Four data sets have been modelled, as follows:

1. The full benchmark data set, with 103 data points (Figure 1)
2. All flow lines but excluding every other GVF point, yielding 53 data points (Figure 5)
3. Excluding every other flow line, yielding 51 data points (Figure 6)
4. Excluding every other flow line and every other GVF point for the remaining flow lines, yielding 27 points (Figure 7).

A range of different neural network design techniques has been applied to each of these data sets and the resulting models having been tested against the full data set to evaluate model performance. It is to be expected that modelling results will be poorer with smaller data sets; of interest is quantification of the reduction in model accuracy, the relative impact of reductions in flow lines vs GVF points, and the relative performance of different neural net techniques over the various data sets.

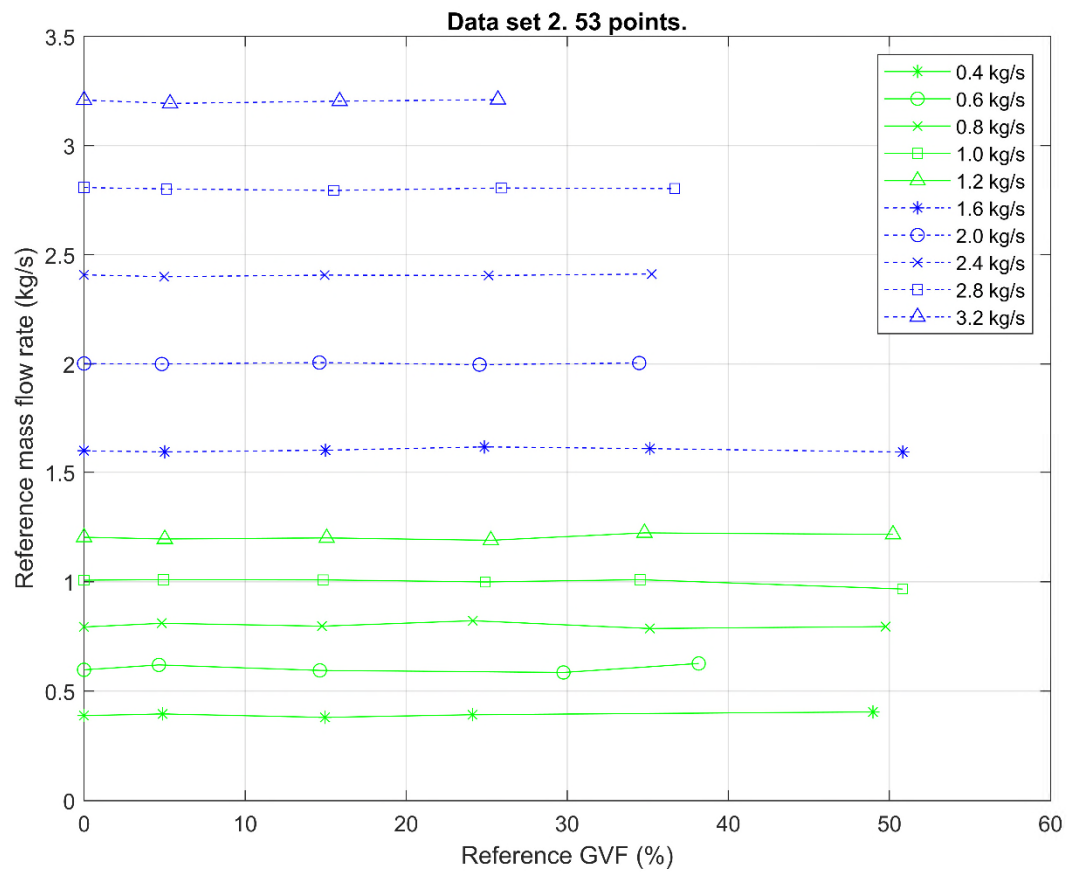


Figure 5: Data set 2, excluding every other GVF point.

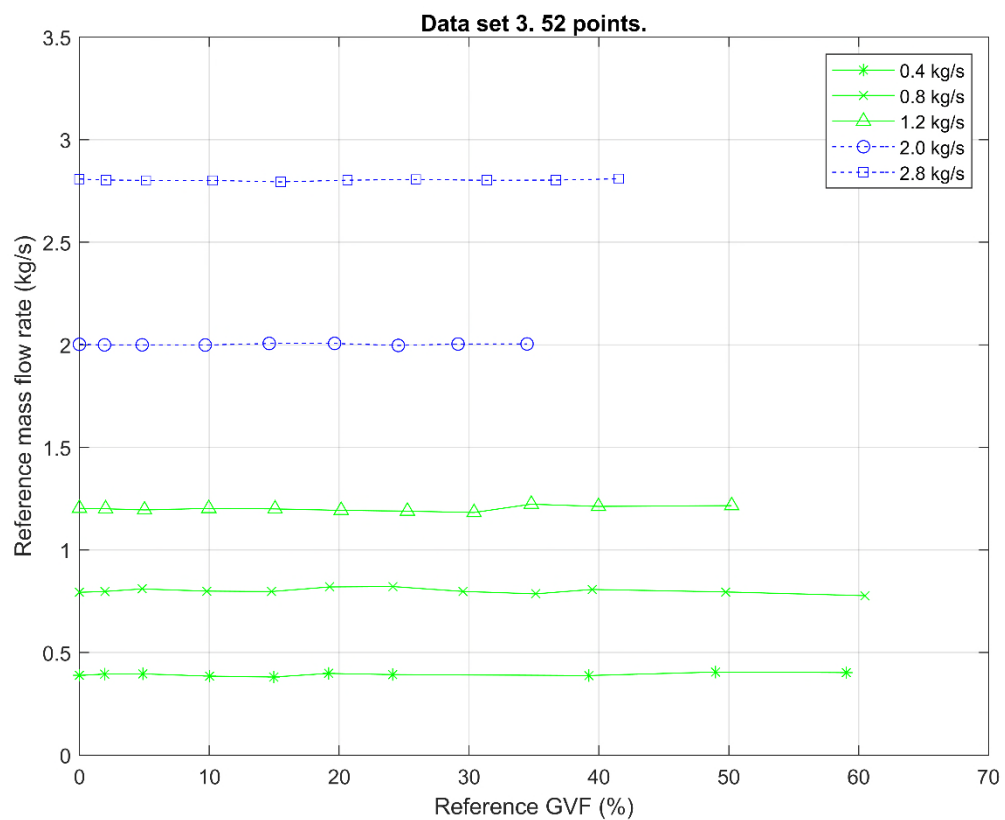


Figure 6: Data set 3, excluding every other flow line.

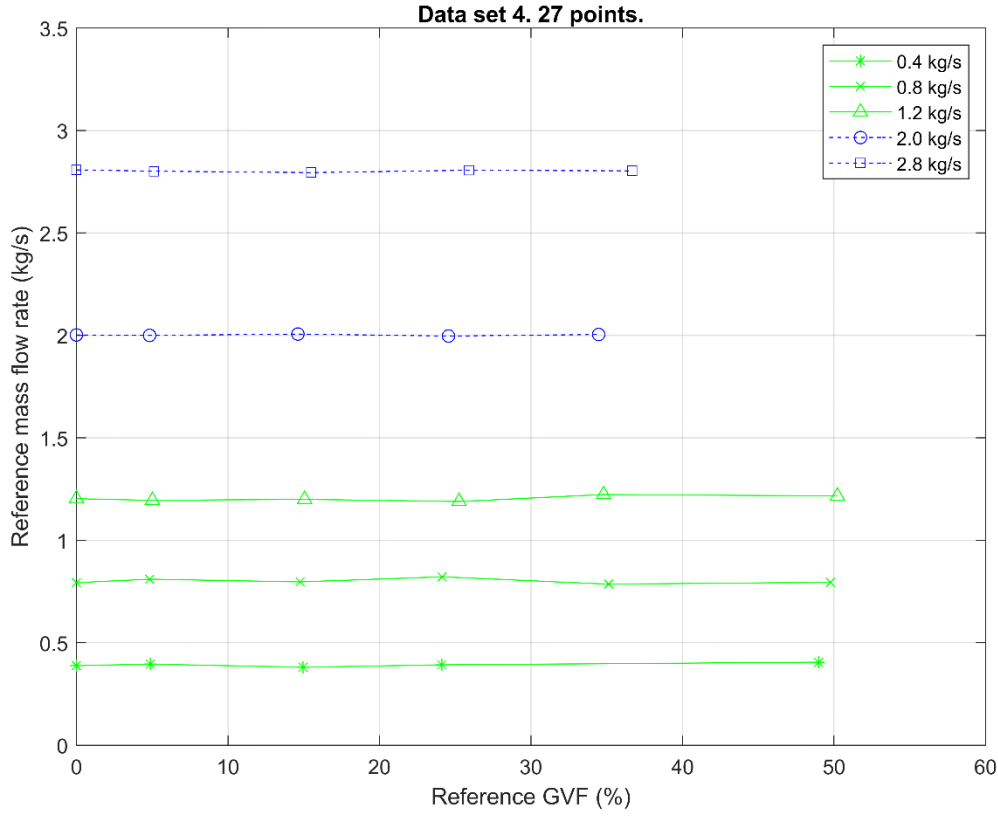


Figure 7: Data set 4, excluding every flow line and every other GVF point.

4. Neural networks modelling techniques

For our experimental analysis we used an open source Python library Keras [11] (with TensorFlow backend) which supports the construction of various neural network structures in a straightforward manner. Keras implements a number of activation functions, and the most popular optimizers, thus making it easy to experiment with a range of options without requiring a high degree of coding effort.

4.1 The loss function and the metric

During model optimization, the loss (cost, error) function is used to quantify the desirability of the current model so that the weights can be updated to improve the loss function on the next step. Whenever a machine learning model is trained, the goal is to find a solution that minimizes the loss function. For a regression problem, the most commonly used loss function is the mean squared error (MSE) which is the mean sum of squared distances between the target variables y_i and their predicted values \hat{y}_i :

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2. \quad (5)$$

One possible drawback of using an MSE loss function is that a single, very poor prediction may have a disproportionate influence due to the square weighting.

The Mean Absolute Error (MAE) is another loss function often used for regression models. MAE is the mean sum of absolute differences between target y_i and predicted values \hat{y}_i :

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|. \quad (6)$$

This metric does not weigh large errors as heavily as MSE, and is less sensitive to outliers than MSE. On the other hand MAE, unlike MSE, is not differentiable. In the modelling examples described here, MSE was used as the loss function to be minimized during network training, while MAE was used as the metric to evaluate final network performance on the full benchmark data set.

4.2 Optimizers – SGD and Adam

During the training process, the parameters (weights) of our model are adjusted to minimize the loss function. Gradient descent is the most widely used technique to update neural network weightings, but every state-of-the-art Deep Learning library includes a number of different algorithms to optimize gradient descent.

Stochastic Gradient Descent (SGD) is one such variant whereby, instead of performing computations on the whole dataset, which is redundant and inefficient, SGD operates over a small subset of the data. This delivers the same performance as regular gradient descent when the learning rate is low.

Recently, new optimizers have been proposed to tackle complex training scenarios where conventional gradient descent methods behave poorly. One of the most widely used and practical of these optimizers is Adam [12]. It combines the advantages of two SGD extensions – Root Mean Square Propagation (RMSProp) and Adaptive Gradient Algorithm (AdaGrad) – and computes individual adaptive learning rates for different parameters.

Despite the widespread popularity of Adam [13, 14], recent research papers have noted that it can fail to converge to an optimal solution under specific circumstances. References [15] and [16] demonstrate that adaptive optimization techniques such as Adam generalize poorly compared to SGD.

In this analysis both SGD and Adam are applied in order to determine which optimizer is best suited to the benchmark data set. Other adaptive learning-rate based optimizers (RMSprop, AdaDelta, etc.) are not considered here because they are similar to Adam.

4.3. Activation functions

The main purpose of the neural net activation function is to introduce non-linearity into the network so that it is capable of reproducing complex behaviours. The activation and loss functions are often selected together. For example, the softmax activation function and the cross-entropy loss function have been found to work well together in a multi-dimensional classification problem [17].

It is widely recognized that a linear activation function is most effective for the output neuron layer in a regression problem [18]. There remains the choice of activation function for the hidden layer.

Sigmoid, one of the classic activation functions that dominated neural network practice for several decades, was eventually judged to be unsuited to learning tasks because of its small derivative, which may result in problems with vanishing gradients. The hyperbolic tangent function (tanh), defined as:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (7)$$

gives better training performance than sigmoid [19, 20], but the tanh function cannot avoid the vanishing gradient issue either.

From this perspective, the ReLU [21] function has proven to be more effective. ReLU is defined as

$$f(x) = \max(0, x) \quad (8)$$

where x is the input to the neuron. Its derivative is unity in the positive region. Currently, ReLU is the most widely used activation function, and its effectiveness has been verified in numerous works, e.g. see [22]. Its success has led to various extensions proposed, one of which is SELU.

The Scaled Exponential Linear Unit (or SELU) was recently introduced in [23] for the development of so-called self-normalizing neural networks. Here a normalization factor λ is included in the neuron's activation via the SELU function, which is defined as:

$$f(\alpha, x) = \lambda \begin{cases} \alpha(e^x - 1), & \text{for } x < 0 \\ x, & \text{for } x \geq 0. \end{cases} \quad (9)$$

Some researchers report performance improvement for SELU over ReLU as the activation function [24, 25], but currently it is not as widely applied [26].

The softplus function (Smooth ReLU) was proposed by Dugas et al. [27] and is defined as:

$$f(x) = \log(1 + e^x). \quad (10)$$

It can be considered as a smooth version of ReLU. The smoothness property makes the softplus function potentially more stable than ReLU, which has a discontinuous gradient at zero. Also, while ReLU has no gradient in the negative region, the softplus function provides output discrimination across all real inputs. The softplus function performed favourably against the ReLU and Sigmoid functions in a study [28], showing better model fitting while requiring fewer epochs to convergence during training.

In summary, given the diversity of experiences reported in the literature, here we apply a variety of hidden layer activation functions: tanh, ReLU, SELU and softplus.

4.4. Neural Net Structure

A final design decision concerns the structure of the neural nets. Note that in each case there are two network inputs (the observed density drop and the observed mass flow rate) with a single output (either the mass flow error, or the density drop error). This suggests a basic neural network structure of 2- N -1, where N , the number of neurons in the hidden layer, is to be determined. The tradeoff here is well-understood: a lower number of neurons may lead to a sub-optimal fit, while too many neurons may lead to over-fitting. In the study undertaken, values of $N = 6, 10, 14, 18$ were used.

5. Experimental results

As described in the previous section, model building experiments have been carried out across a number of parameters, as follows:

- Data sets 1 – 4
- Hidden layer width 6, 10, 14, 18 neurons
- Activation functions: softplus, ReLU, SELU, or tanh
- Optimizers: Adam or SGD.

All datasets and program code, along with one of the resulting architectures, are available on the website [31]. For each combination of parameters, two separate networks were generated, predicting the mass flow error and the density drop error respectively. As the weights and biases

are initialized randomly, a different network is obtained for each training, resulting in different performance. In order to minimize the effect of random initialization, the average MAE (equation (6)) across the full benchmark dataset over 20 networks with the same structure was calculated. When training networks, we used Keras callbacks EarlyStopping and ModelCheckpoint [29], preserving the best model and forcibly stopping training if for 500 epochs there was no improvement in the current model. The results are presented in Tables 1 and 2.

Table 1. Mass flow error model performance (best result for each data set highlighted in bold)

Activation function-Optimizer	2-6-1				2-10-1			
Data set	1	2	3	4	1	2	3	4
softplus – Adam	1.31	4.40	3.42	5.48	0.94	1.17	1.01	5.61
relu – Adam	2.60	3.62	4.56	5.70	1.19	3.73	3.54	5.30
selu – Adam	2.84	3.34	3.06	5.36	1.19	1.88	1.40	4.71
tanh – Adam	2.67	2.62	1.80	2.06	1.07	1.58	1.47	1.89
softplus – SGD	1.72	3.38	2.49	4.61	1.61	2.38	1.41	4.01
relu – SGD	2.27	4.14	3.23	3.98	2.12	3.15	2.56	4.18
selu – SGD	2.41	3.89	2.67	4.35	1.60	2.63	1.67	1.86
tanh – SGD	2.29	2.50	2.38	2.70	2.25	2.60	2.46	2.43
	2-14-1				2-18-1			
Data set	1	2	3	4	1	2	3	4
softplus – Adam	0.97	1.22	0.98	5.32	0.98	0.91	0.97	5.27
relu – Adam	0.90	1.57	1.58	4.50	0.87	0.95	1.85	5.01
selu – Adam	0.93	1.56	1.32	3.94	0.93	1.05	1.07	1.74
tanh – Adam	1.74	1.62	1.33	1.72	1.11	1.01	0.90	1.92
softplus – SGD	1.73	3.63	2.20	5.32	1.18	1.63	1.10	2.06
relu - SGD	1.71	4.09	1.83	5.23	1.34	2.64	1.69	3.25
selu - SGD	1.72	1.64	2.19	3.73	1.28	2.81	2.44	4.35
tanh - SGD	1.56	2.32	2.00	2.46	1.81	2.81	2.64	2.29

Tables 3 and 4 show, for each model size, the technique which generates the lowest error, while Table 5 and 6 show the smallest modelling MAE achieved in each case. A number of observations can be drawn from these results, as discussed below.

5.1. Using a variety of activation functions and optimizers delivers better models

One clear observation from the data in tables 1 and 2 is the wider range of results obtained for the same data set and NN architecture, when using different optimizers and activation functions. For example, in Table 2, for data set 1 and the smallest architecture 2-6-1, the best result is an MAE of 0.46%, while the worst result is 1.86%, around 4 times larger. Over all the examples, the ratio between the best and worst MAE is typically between 1.5 times and 4 times. Given also that,

as discussed below, there is no overall best combination of activation function and optimizer that consistently delivers the best modelling results, we conclude the following: Assuming that the cost of obtaining the experimental data is significantly higher than that of carrying out NN modelling, it is worthwhile exploring a range of activation functions and optimizers in order to find a best solution for a particular dataset. This could result in an MAE reduction by a factor of two or more compared with a strategy of only using a single activation and optimizer.

Table 2. Density drop error model performance (best result for each data set highlighted in bold)

Activation function-Optimizer	2-6-1				2-10-1			
Data set	1	2	3	4	1	2	3	4
softplus - Adam	0.69	0.69	3.13	3.29	0.40	0.51	1.28	2.56
relu - Adam	1.00	2.24	3.27	3.38	0.85	1.07	2.50	3.20
selu - Adam	0.97	1.35	3.13	3.55	0.54	0.90	1.46	2.67
tanh - Adam	0.46	0.77	2.79	4.17	0.45	0.98	1.25	3.33
softplus - SGD	1.86	2.28	1.59	2.92	2.18	2.40	2.22	2.86
relu - SGD	1.42	1.84	1.69	2.76	1.17	2.38	2.04	3.07
selu - SGD	1.57	1.94	2.20	3.02	1.77	2.17	2.45	3.17
tanh - SGD	1.34	2.60	1.26	2.15	1.44	2.75	1.81	3.20
	2-14-1				2-18-1			
Data set	1	2	3	4	1	2	3	4
softplus - Adam	0.40	0.52	1.31	2.57	0.47	0.53	1.11	1.80
relu - Adam	0.44	0.51	1.63	2.77	0.35	0.66	1.26	2.62
selu - Adam	0.46	0.52	1.15	1.81	0.42	0.51	1.26	1.85
tanh - Adam	0.54	0.72	0.74	2.33	0.50	0.65	0.71	2.77
softplus - SGD	1.29	2.87	2.62	2.85	1.00	1.07	1.18	1.23
relu - SGD	1.19	3.07	2.61	2.85	1.09	1.36	1.36	2.28
selu - SGD	0.82	2.96	2.77	3.03	0.87	2.24	1.94	2.88
tanh - SGD	3.29	3.81	3.35	4.00	1.78	4.43	3.96	2.53

Table 3. Best (smallest error) models 2-N-1 for MFR error correction

N	Data_set 1	Data_set 2	Data_set 3	Data_set 4
6	softplus – Adam	tanh – SGD	tanh – Adam	tanh – Adam
10	softplus – Adam	softplus – Adam	softplus – Adam	tanh – Adam
14	relu – Adam	softplus – Adam	softplus – Adam	tanh – Adam
18	relu – Adam	softplus – Adam	tanh – Adam	selu – Adam

Table 4. Best (smallest error) models 2-N-1 for density drop error correction

N	Data_set 1	Data_set 2	Data_set 3	Data_set 4
6	tanh - Adam	softplus - Adam	tanh - SGD	tanh - SGD
10	softplus - Adam	softplus - Adam	tanh - Adam	softplus - Adam
14	softplus - Adam	relu - Adam	tanh - Adam	selu - Adam
18	relu - Adam	selu - Adam	tanh - Adam	softplus - SGD

Table 5. Error values for best models 2-N-1 for MFR error correction

N	Data_set 1	Data_set 2	Data_set 3	Data_set 4
6	1.31	2.50	1.80	2.06
10	0.94	1.17	1.01	1.89
14	0.90	1.22	0.98	1.72
18	0.87	0.91	0.90	1.74

Table 6. Error values for best models 2-N-1 for density drop error correction

N	Data_set 1	Data_set 2	Data_set 3	Data_set 4
6	0.46	0.69	1.26	2.15
10	0.40	0.51	1.25	2.56
14	0.40	0.51	0.74	1.81
18	0.35	0.51	0.71	1.23

5.2. Evaluation of optimizers and evaluation functions

There is significant diversity observed in the results achieved using different evaluation functions. Softplus and tanh, particularly in combination with Adam, most frequently deliver the best model. Tanh seems to be more appropriate for the smaller amount of data.

Adam usually delivered the best model compared with SGD: it was best in 15 out of the 16 results in Table 3 and 13 out of 16 results in Table 4. Nevertheless, as stated above, where computational resources allow building additional models, SGD techniques will occasionally

deliver a better results, especially for a small amount of data and/or for a small number N . Two combinations, relu+SCD, selu+SGD, failed in this analysis to deliver any best models.

5.3. Impact of hidden layer size

This issue is most clearly illuminated through the ‘best’ results shown in Tables 5 and 6. For data set 1, where the model is fitted to the complete data set which is also used for the MAE evaluation, little model improvement is observed beyond $N = 10$. In the case of data set 4, adding hidden units consistently improves the model, but the errors remain twice as large as for data set 1 with mass flow, and over three times larger for density. As discussed below, data sets 2 and 3 have complimentary behavior for mass flow and density models. For mass flow, data set 3 is consistently better than data set 2, and, as with data set 1, there is limited improvement beyond $N = 10$. For density, data set 2 is consistently better than data set 3, and there is no improvement beyond $N = 10$. For mass flow and $N = 18$, both data sets 2 and 3 approach the low errors obtained with data set 1. With data set 4, little mass flow model improvement is observed for $N > 10$.

In reality, having obtained expensive experimental data, it will of course be sensible to use all the data in the subsequent models. The results for data set 1 however illustrate the notion of an optimal hidden layer size – here observed at around $N = 10$ for both mass flow and density models – where increasing the number of units further does not lead to significant model improvement. The can even be observed in data set 4 for mass flow.

5.4. Impact of data sparsity on model performance

Recalling the structures of the four data sets: data set 1 is complete, data set 2 has a reduced number of GVF points, data set 3 has a reduced number of flow points, and data set 4 has reductions in both, with approximately one quarter of the points of data set 1.

From Tables 5 and 6, the four-fold increase in data points between data set 4 and data set 1 leads to a reduction in mass flow error of a factor of 2 (from 1.74% down to 0.87%), and a reduction in density error by a factor of 3 (from 1.23% down to 0.35%).

As discussed above in section 5.3, data sets 2 and 3 have different behavior for mass flow and density: data set 2 outperforms data set 3 for density and vice-versa for mass flow. These results may reflect the relative sensitivities of the errors to changes in true mass flow rate and GVF. As shown in Figure 2, the mass flow error is relatively insensitive to true mass flow rate, particularly for the upper half of the flow range – hence the wider spacing of flow lines at high flow in the original experimental design. In Figure 4, the variations in density drop error with density drop

(and hence GVF) are reasonably linear, so that a reduced number of data points along the density drop axis has limited impact.

In practice, both mass flow and density models are required, and there appears from this data set no clear benefit in providing say more GVF points than flow points in an experimental design. Accordingly, it is recommended to provide approximately equal number of flow lines and GVF points in an experimental programme, but with the flow line distribution weighted towards low flow.

6. Conclusions

This paper has presented a benchmark data set of two-phase (oil/nitrogen) mass flow and density errors obtained from a 50 mm Coriolis mass flow meter. The data set has a large turndown ratio (8:1), GVF values up to 60%, and has positive density errors arising from high oil viscosity and gas/liquid slip.

An investigation has explored the influence of hidden layer width, activation function, optimizer, and data sparsity on neural net model accuracy. Assuming that the cost of data collection is significantly higher than that of analysis and modelling, it is recommended that a range of activation functions and optimizers are applied to find a best solution for a particular dataset. In the example presented here, this led to reduction in modelling error by a factor of two or more compared with using any single activation and optimizer function. No consistent benefit was observed in having more flow points than GVF points (or vice versa) in a two-phase data set. It is recommended to have approximately equal numbers of each, while having shorter flow line spacing at lower flow rates.

We welcome further analysis and modelling based on this data set, and we hope to be able to provide larger data sets, including three-phase (oil/water/gas) data, in the future.

Acknowledgement

We acknowledge the support of Schneider-Electric in funding the original experimental work from which the benchmark data set is derived.

Bibliography

1. Wang T., Baker R., Coriolis flowmeters: A review of developments over the past 20 years, and an assessment of the state of the art and likely future directions, *Flow Meas. Instrum.*, 40 (2014), 99–123.

2. M. Li and M. Henry, "Complex signal processing for Coriolis mass flow metering in two-phase flow," *Flow Meas. Instrum.*, vol. 64, pp. 104–115, Dec. 2018.
3. Henry, MP, Ibryaeva, O, Semenov, A. "Measurement Validation for ICPS: Matrix Pencil Method for Coriolis Metering with Liquid/Gas Flow", 1st IEEE International Conference on Industrial Cyber-Physical Systems (ICPS-2018). 15 May 2018.
4. N. T. Basse, "A review of the theory of Coriolis flowmeter measurement errors due to entrained particles," *Flow Meas. Instrum.*, vol. 37, pp. 107–118, Jun. 2014.
5. Y. Yan, L. Wang, T. Wang, X. Wang, Y. Hu, Q. Duan, Application of soft computing techniques to multiphase flow measurement: A review, *Flow Meas. Instrum.* 60 (2018) 30–43. doi: 10.1016/j.flowmeasinst.2018.02.017
6. L. Wang., J. Liu, X. Wang, T. Wang, Gas–Liquid Two-Phase Flow Measurement Using Coriolis Flowmeters Incorporating Artificial Neural Network, Support Vector Machine, and Genetic Programming Algorithms, *IEEE Trans. Instrum. Meas.* 66(2017) 852 - 868. doi: 10.1109/TIM.2016.2634630
7. M. Henry, M. Tombs, M. Duta, F. Zhou, R. Mercado, F. Kenyery, J. Shen, M. Morles, C. Garcia, and R. Langansan, "Two-phase flow metering of heavy oil using a Coriolis mass flow meter: A case study," *Flow Meas. Instrum.*, vol. 17, no. 6, pp. 399–413, Dec. 2006.
8. Tombs, MS, Zhou, FB, Henry, MP. "Two-Phase Coriolis Mass Flow Metering with High Viscosity Oil", *Flow Measurement and Instrumentation*, Nov 2017.
<https://doi.org/10.1016/j.flowmeasinst.2017.11.009>
9. M. Henry, M. Tombs, M. Zamora, F. Zhou, Coriolis mass flow metering for three phase flow: a case study, *Flow Meas. Instrum.* 30(2013) 112–122. doi: 10.1016/j.flowmeasinst.2013.01.003
10. R. Liu, M. Fuent, M. Henry, M. Duta, "A neural network to correct mass flow errors caused by two-phase flow in a digital coriolis mass flowmeter, *Flow Meas. Instrum.* 12 (2001) 53–63. doi: 10.1016/S0955-5986(00)00045-5.
11. F. Chollet, *Deep learning with Python*, Manning Publications Co, 353 p., 2018.
12. D. Kingma, J. Ba, Adam: a method for stochastic optimization, in: *Proceedings of International Conference for Learning Representations*, 2015.
13. K. Rajput, G. Chetty, R. Davey, Deep neural models for chronic disease status detection in free text clinical records, in: *IEEE International conference on Data Mining Workshops*, 2018. doi: 10.1109/ICDMW.2018.00127
14. S. Jacobson, D. Reichman, J. Bjomstad, L. Collins, J. Malof, Reliable training of convolutional neural networks for GRP-based buried threat detection using the Adam

- optimizer and batch normalization, in: Proceedings Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIV, 2019.
15. N. Keskar, R. Socher, Improving generalization performance by switching from Adam to SGD, <https://arxiv.org/abs/1712.07628>
 16. A. Wilson, R. Roelofs, M. Stern, N. Srebro, B. Recht, The marginal value of adaptive methods in machine learning, <https://arxiv.org/abs/1705.08292>
 17. M. Nielsen, Neural network and Deep Learning, Determination Press, 2015.
<http://neuralnetworksanddeeplearning.com>
 18. Charu C. Aggarwal, Neural Networks and Deep Learning: A Textbook, Springer, 2018, 497 p.
 19. B. Karlik, A. Vehbi, Performance analysis of various activation functions in generalized MLP architectures of neural networks, International Journal of Artificial Intelligence and Expert Systems, 1 (2011). 111–122.
 20. R. Neal, Connectionist learning of belief networks, Artificial Intelligence, 56 (1992), 71–113. doi:10.1016/0004-3702(92)90065-6
 21. V. Nair, G. Hinton, Rectified linear units improve restricted Boltzmann machines, in: Proceedings of the 27th international conference on machine learning. 2010.
 22. X. Glorot, A. Bordes, Y. Bengio. Deep sparse rectifier neural networks, in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, p 315–323.
 23. G. Klambauer, T. Unterthiner, A. Mayr, Self-Normalizing Neural Networks, in: 31th Conference on Neural Information Processing Systems, 2017.
 24. F. Sakketou, N. Ampazis, On the Invariance of the SELU Activation Function on Algorithm and Hyperparameter Selection in Neural Network Recommenders, in: book Artificial Intelligence Applications and Innovations, 2019.
 25. Ł. Kidziński et al., Learning to Run Challenge Solutions: Adapting reinforcement learning methods for neuromusculoskeletal environments, in: The NIPS '17 Competition: Building Intelligent Systems. The Springer Series on Challenges in Machine Learning. Springer, Cham.
 26. D. Li, H. Yang, C. Yuan, X. Qin, Leaf Classification Utilizing Densely Connected Convolutional Networks with a Self-gated Activation Function, in book: D. Huang, M. Gromiha, K.Han, A. Hussain, Intelligent Computing Methodologies. Springer, 2018.
https://doi.org/10.1007/978-3-319-95957-3_40

27. C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, R. Garcia, Incorporating Second-Order Functional Knowledge for Better Option Pricing, in: Advances in Neural Information Processing Systems 2001.
28. H. Zheng, Z. Yang, W. Liu, J. Liang, Y. Li, Improving deep neural networks using softplus units, in: International Joint Conference on Neural Networks, 2015.
29. <https://keras.io/callbacks/>
30. YF Geng, H Yeung, Y Cao, LC Xing, H Zhu, W Drahm, Using Coriolis Mass Flowmeter for Wet Gas Metering, Measurement and Control, Vol 43, No 6, 2010.
31. <https://cmfdata.susu.ru>. The site manager is Olga Ibryaeva: ibriaevaol@susu.ru

Appendix 1: Two-phase flow data set

Nominal	Reference Measurements			Coriolis Meter Measurements			Coriolis Meter Errors		
Mass Flow	Mass Flow	Density	GVF	Mass Flow	Density	Density Drop	Mass Flow	Density	Density Drop
[kg/s]	[kg/s]	[kg/m ³]	[%]	[kg/s]	[kg/m ³]	[%]	[%]	[%]	[%]
0.40	0.388	881.99	0.00	0.382	882.35	-0.04	-1.51	0.04	0.04
0.40	0.395	864.93	1.94	0.373	870.72	1.27	-5.57	0.67	0.66
0.40	0.396	839.05	4.88	0.374	853.64	3.20	-5.43	1.74	1.66
0.40	0.384	793.84	10.02	0.349	826.06	6.32	-9.11	4.06	3.67
0.40	0.380	750.51	14.96	0.333	801.33	9.12	-12.56	6.77	5.78
0.40	0.398	713.28	19.19	0.343	779.53	11.59	-13.86	9.29	7.54
0.40	0.392	669.94	24.13	0.323	756.40	14.20	-17.64	12.91	9.84
0.40	0.387	537.32	39.22	0.262	692.25	21.46	-32.26	28.83	17.61
0.40	0.405	451.57	48.99	0.259	636.16	27.82	-36.00	40.88	20.98
0.40	0.402	363.27	59.05	0.202	578.18	34.39	-49.89	59.16	24.42
0.60	0.597	880.69	0.00	0.594	881.11	-0.05	-0.49	0.05	0.19
0.60	0.602	864.00	1.90	0.587	868.56	1.37	-2.61	0.53	0.67
0.60	0.620	839.83	4.65	0.609	850.42	3.42	-1.71	1.26	1.36
0.60	0.606	797.65	9.45	0.588	823.57	6.47	-2.97	3.25	3.10
0.60	0.595	752.13	14.64	0.549	796.28	9.56	-7.62	5.87	5.16
0.60	0.577	708.05	19.67	0.514	770.35	12.50	-10.87	8.80	7.22
0.60	0.585	619.42	29.77	0.477	718.84	18.34	-18.59	16.05	11.43
0.60	0.585	575.30	34.79	0.433	690.70	21.53	-25.90	20.06	13.24
0.60	0.626	545.74	38.16	0.449	670.73	23.79	-28.38	22.90	14.33

0.60	0.603	451.99	48.86	0.358	602.77	31.50	-40.61	33.36	17.25
0.80	0.793	881.90	0.00	0.786	882.26	-0.04	-0.91	0.04	0.05
0.80	0.797	864.65	1.96	0.779	868.54	1.51	-2.34	0.45	0.46
0.80	0.810	839.35	4.83	0.795	849.98	3.60	-1.82	1.27	1.23
0.80	0.799	795.69	9.79	0.777	819.66	7.03	-2.73	3.01	2.75
0.80	0.797	751.82	14.78	0.748	790.32	10.35	-6.13	5.12	4.41
0.80	0.820	712.36	19.27	0.747	763.67	13.36	-8.94	7.20	5.87
0.80	0.822	669.57	24.14	0.715	738.45	16.21	-12.95	10.29	7.87
0.80	0.798	622.25	29.52	0.665	709.69	19.47	-16.58	14.05	9.98
0.80	0.786	573.00	35.13	0.603	677.80	23.08	-23.23	18.29	11.95
0.80	0.807	534.67	39.49	0.567	650.49	26.17	-29.74	21.66	13.21
0.80	0.795	444.58	49.76	0.455	589.61	33.07	-42.71	32.62	16.52
0.80	0.777	350.49	60.48	0.343	528.44	40.01	-55.91	50.77	20.25
1.00	1.007	882.02	0.00	0.999	882.47	-0.05	-0.74	0.05	0.05
1.00	0.995	864.64	1.98	0.982	867.32	1.66	-1.31	0.31	0.30
1.00	1.010	838.76	4.92	0.982	848.04	3.85	-2.69	1.11	1.05
1.00	1.010	795.57	9.84	0.979	817.02	7.36	-3.07	2.70	2.44
1.00	1.008	751.63	14.84	0.953	786.72	10.80	-5.47	4.67	3.98
1.00	0.999	706.94	19.93	0.919	757.48	14.11	-8.06	7.15	5.74
1.00	0.999	663.28	24.90	0.871	728.24	17.42	-12.83	9.79	7.37
1.00	1.004	620.86	29.73	0.855	701.82	20.42	-14.77	13.04	9.19
1.00	1.009	578.74	34.53	0.810	673.59	23.62	-19.75	16.39	10.76
1.00	1.003	532.47	39.79	0.769	643.31	27.05	-23.32	20.82	12.58
1.00	0.967	435.66	50.82	0.647	580.63	34.16	-33.05	33.28	16.45
1.00	1.022	360.70	59.36	0.613	525.97	40.36	-40.07	45.82	18.75
1.20	1.203	881.99	0.00	1.198	882.47	-0.05	-0.43	0.05	0.05
1.20	1.201	864.28	2.02	1.189	865.41	1.88	-0.99	0.13	0.13
1.20	1.195	837.97	5.01	1.173	845.77	4.11	-1.84	0.93	0.88
1.20	1.202	794.53	9.96	1.158	813.65	7.75	-3.66	2.41	2.17
1.20	1.200	749.72	15.06	1.136	782.14	11.32	-5.29	4.32	3.67
1.20	1.193	705.04	20.15	1.098	751.13	14.84	-7.97	6.54	5.23
1.20	1.189	660.22	25.25	1.060	723.07	18.02	-10.86	9.52	7.13
1.20	1.183	615.19	30.38	1.027	692.50	21.49	-13.18	12.57	8.76
1.20	1.223	576.44	34.79	1.032	665.90	24.50	-15.64	15.52	10.14
1.20	1.212	531.04	39.96	0.969	635.92	27.90	-20.01	19.75	11.89
1.20	1.216	440.92	50.22	0.907	576.59	34.62	-25.38	30.77	15.38
1.60	1.598	881.99	0.00	1.596	882.38	-0.04	-0.13	0.04	0.04
1.60	1.595	864.32	2.01	1.578	862.52	2.20	-1.02	-0.21	-0.20
1.60	1.593	838.01	5.01	1.556	840.78	4.67	-2.33	0.33	0.32
1.60	1.598	794.29	9.98	1.538	807.89	8.40	-3.77	1.71	1.54
1.60	1.601	750.28	15.00	1.524	776.19	11.99	-4.83	3.45	2.94

1.60	1.606	706.82	19.94	1.518	746.88	15.32	-5.45	5.67	4.55
1.60	1.616	663.61	24.86	1.509	715.72	18.85	-6.64	7.85	5.91
1.60	1.597	617.04	30.16	1.470	682.74	22.59	-7.93	10.65	7.45
1.60	1.608	573.48	35.12	1.439	652.08	26.06	-10.48	13.71	8.92
1.60	1.622	530.72	39.99	1.415	621.17	29.57	-12.78	17.04	10.26
1.60	1.593	435.57	50.82	1.254	550.48	37.58	-21.28	26.38	13.03
2.00	2.002	881.87	0.00	2.001	882.24	-0.04	-0.05	0.04	0.05
2.00	2.000	864.93	1.93	1.982	862.01	2.25	-0.87	-0.34	-0.32
2.00	2.000	839.48	4.83	1.961	838.11	4.96	-1.95	-0.16	-0.15
2.00	1.999	796.85	9.69	1.945	804.38	8.79	-2.69	0.94	0.86
2.00	2.007	753.55	14.62	1.935	772.11	12.45	-3.57	2.46	2.11
2.00	2.007	708.33	19.65	1.924	734.36	16.61	-4.12	3.67	3.08
2.00	1.997	666.24	24.56	1.888	706.56	19.88	-5.44	6.05	4.58
2.00	2.004	624.79	29.15	1.891	671.01	23.78	-5.65	7.40	5.38
2.00	2.004	579.17	34.47	1.853	641.25	27.29	-7.56	10.72	7.05
2.40	2.408	881.63	0.00	2.405	881.98	-0.04	-0.15	0.04	0.08
2.40	2.401	864.19	1.99	2.367	859.92	2.46	-1.41	-0.49	-0.44
2.40	2.400	837.99	4.97	2.328	833.42	5.47	-3.01	-0.55	-0.48
2.40	2.412	794.85	9.88	2.344	797.21	9.58	-2.82	0.30	0.30
2.40	2.407	750.48	14.94	2.334	761.48	13.63	-3.04	1.47	1.28
2.40	2.406	706.38	19.96	2.312	726.07	17.65	-3.91	2.79	2.26
2.40	2.405	661.06	25.12	2.288	690.46	21.69	-4.89	4.45	3.36
2.40	2.405	616.46	30.20	2.250	654.95	25.71	-6.47	6.25	4.39
2.40	2.412	572.11	35.25	2.226	620.92	29.57	-7.70	8.53	5.56
2.40	2.417	526.67	40.42	2.172	584.65	33.69	-10.12	11.01	6.60
2.80	2.808	881.26	0.00	2.808	881.31	-0.01	-0.01	0.01	0.09
2.80	2.803	863.35	2.04	2.756	857.80	2.67	-1.70	-0.64	-0.55
2.80	2.802	836.41	5.12	2.721	829.16	5.92	-2.89	-0.87	-0.75
2.80	2.801	791.33	10.26	2.720	788.89	10.49	-2.89	-0.31	-0.21
2.80	2.794	745.37	15.49	2.699	749.80	14.93	-3.40	0.59	0.56
2.80	2.802	700.20	20.64	2.690	711.86	19.24	-4.02	1.66	1.37
2.80	2.806	654.01	25.91	2.671	672.43	23.72	-4.80	2.82	2.13
2.80	2.802	606.13	31.36	2.668	633.98	28.08	-4.77	4.60	3.20
2.80	2.803	559.41	36.68	2.588	592.81	32.75	-7.67	5.97	3.82
2.80	2.809	516.85	41.53	2.548	556.54	36.87	-9.28	7.68	4.53
3.20	3.208	881.14	0.00	3.210	881.31	-0.02	0.04	0.02	0.12
3.20	3.199	862.53	2.12	3.143	855.80	2.88	-1.76	-0.78	-0.67
3.20	3.193	834.26	5.34	3.117	824.70	6.41	-2.37	-1.15	-1.00
3.20	3.199	787.08	10.72	3.134	780.67	11.41	-2.03	-0.81	-0.65
3.20	3.203	741.86	15.87	3.115	740.10	16.01	-2.75	-0.24	-0.12
3.20	3.207	697.25	20.95	3.105	700.88	20.46	-3.18	0.52	0.48

3.20	3.210	655.64	25.70	3.052	664.83	24.56	-4.93	1.40	1.11
3.20	3.207	614.09	30.43	3.055	629.31	28.58	-4.74	2.48	1.79