

Support Vector Machine modeling applied to benchmark data set for two-phase Coriolis mass flow metering

Olga L. Ibryaeva¹, Denis K. Lebedev¹, Manus P. Henry^{1,2,3}

¹South Ural State University, Russia.

²Coventry University, UK

³University of Oxford, UK

Abstract

An earlier paper introduced a dataset of Coriolis meter mass flow and density errors, induced by the effects of two-phase (gas/liquid) flow, as a benchmark for which various error correction strategies might be developed. That paper further presented a series of error correction models based on neural nets. The current paper presents an alternative analysis of the same data set, using a support vector machine (SVM) approach. The analysis demonstrates that, for the benchmark data set, more accurate models are generated than those developed using neural nets. More specifically, it is found that a linear SVM model provides better performance than non-linear SVM. This improved performance may arise from over-fitting by both non-linear SVM and neural nets on this relatively small data set.

Keywords: Coriolis mass flow metering, artificial neural network, support vector machine (SVM), two-phase flow.

1. Introduction

Two-phase (liquid/gas) flow is a common feature in many industrial processes. Accurate measurement of the flowrate of a two-phase mixture is often challenging. Over the past three decades, significant progress has been made in developing new techniques that may offer solutions to the industrial measurement challenges. These techniques can be classified into direct and indirect measurement groups according to measurement strategies being deployed [1].

For single phase flow, typical Coriolis meter mass flow accuracy is around 0.1 % for liquids and 0.5 % for gases, over a turndown (the ratio between the maximum and minimum flow) of typically 50:1 or more for liquids. Most Coriolis meter designs also output a density measurement with a similarly high accuracy specification for pure liquids. The introduction of gas into a liquid flow can be described in terms of the gas volumetric fraction (GVF) i.e. the percentage by volume of

the flow that is gas. The presence of a two-phase liquid/gas mixture results in errors, often repeatable, in the mass flow and density measurements generated by the Coriolis meter. A longstanding technical problem [2] is to find means of correcting these erroneous measurements. While in a reference laboratory the true GVF is known, for industrial applications a useful additional metric is the observed density drop, usually expressed as a percentage, describing the difference between the (assumed known) true liquid density and the apparent density when some gas is entrained. Note that because of two-phase induced errors, the observed density drop is not accurate, but is readily available and often repeatable. Accordingly, a common framework for two-phase flow correction is to develop a mapping from observed mass flow rate and density drop to corrected mass flow and density drop (and hence GVF).

With the recent development of artificial intelligence and machine learning, soft computing techniques provide alternative approaches to traditional statistical methods and extend the capabilities of empirical models. The indirect techniques include artificial neural networks (ANNs) [2, 3], Support Vector Machines (SVM), genetic programming [4], hybrid models [1], etc.

Our earlier paper [5] provides a Coriolis meter two-phase flow data set (which is a subset of the data used in [6]) offered to the research community as a benchmark i.e. a common data set for comparing correction techniques. In this paper, four subsets of the full benchmark data set were modeled: 1) the full benchmark data set, with 103 data points, 2) all flow lines, but excluding every other GVF point, yielding 53 data points, 3) every other flow line excluded, yielding 51 data points, 4) every other flow line excluded, and every other GVF point excluded in the remaining flow lines, yielding 27 points. The investigation explored on the one hand the trade-off between data scarcity (i.e. the spacing between experimental data points) and the resulting model accuracy, given the high cost of obtaining each experimental point in a flow test facility, while on the other hand evaluating a range of options and parameters relating to the neural net implementation. The data set is available at the website [7].

SVM was first applied [9] to measure the overall mass flowrate of oil-water two-phase flow, which is a simpler problem than for liquid/gas mixtures. The application of SVM to two-phase (air/water) measurement was investigated by Wang et al [4]. The performance of ANN, SVM, and GP models were assessed and compared. The modeling results suggest that the SVM models are superior to the ANN and GP models for two-phase flow measurement, in terms of robustness and accuracy. The SVM method, using a grid search, is applied in [10] to establish the mass flow measurement error (MFME) model for gas-liquid two-phase flow. The results demonstrate that the SVM method

has good generalization performance, and can be used to reduce the MFME in real-time for gas-liquid two-phase flow.

The purpose of the current paper is to apply SVM techniques to the same benchmark data set and to compare the resulting SVM modeling accuracies to the ANN results reported previously [5]. Mass flow and density drop measurement errors are taken as the output variables of the SVM models and the ‘observed’, or equivalently the ‘apparent’, mass flowrate and density drop are taken as the input variables. The models have been developed using Python with a free machine-learning library Scikit-learn [12]. One significant difference between this work and previous papers on the application of SVM to Coriolis two-phase measurement is that here, alongside the conventional non-linear SVM, we demonstrate improved performance using linear SVM with augmented parameters.

The structure of the paper is as follow. Section 2 describes the basic **theory** of Support Vector Regression, and the methods used in this paper. The results and analysis of the experiments are the focus of Section 3. We conclude in Section 4 with a summary.

2. Linear and non-linear SVR

2.1 Basic **theory** of Support Vector Regression

The application of the Support Vector Machine approach specifically to regression problems (as opposed to say, classification problems) is often denoted as Support Vector Regression (SVR), and this terminology will be used for the rest of the paper. We employ the SVR algorithm proposed by Vapnik [11], a short summary of which is given below.

Let $F = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ be the set of N samples, where $x_i \in R^n$ is the input and y_i is the output of the regression problem. The input-output relation can be described as a linear regression model **with weights w , and biases b** , as follows (1):

$$f(x, w) = \sum_{i=1}^N w_i x_i + b = w^T x + b, \quad (1)$$

where $x = (x_1, \dots, x_N)^T$, $w = (w_1, \dots, w_N)^T$ and $f = (f_1, \dots, f_N)^T$ **is the output of the model.**

The loss function applied here is the ε -tolerance loss function defined as follows (2):

$$Y_\varepsilon = \begin{cases} 0 & \text{if } |y - f(x, w)| < \varepsilon \\ |y - f(x, w)| - \varepsilon & \text{other} \end{cases} \quad (2)$$

The philosophy of regression by support vector machine is that a tube or a band, which has radius ε , is defined around the estimating function $f(x, w)$. If the value y is inside the tube, there is no

loss. The loss for all points falling outside the tube is set to be the absolute value of the difference between that estimating point and the radius of ε .

The optimization problem is given by (3)

$$\min R(w, \xi, \xi^*) = \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) \quad (3)$$

subject to constraints (4)

$$\begin{cases} y_i - w^T x_i - b \leq \varepsilon + \xi_i \\ w^T x_i + b - y_i \leq \varepsilon + \xi_i^* \end{cases} \quad \xi_i \geq 0, \quad \xi_i^* \geq 0, \quad (4)$$

where C is the regularization, tunable parameter which determines the trade-off between the flatness of f and the extent to which deviations larger than ε are tolerated.

Adopting a soft-margin approach similar to that employed in SVM, slack variables ξ_i, ξ_i^* can be added to guard against outliers. These variables determine how many points can be tolerated which fall outside the ε – tube.

The dual formula for solving non-linear SVR is obtained by using Lagrange Multipliers from the primal function, introducing non-negative multipliers α_i and α_i^* for each observation x_i , given as (5):

$$\min \frac{1}{2} \sum_{i,j=1}^N (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(x_i, x_j) + \varepsilon \sum_{i=1}^N (\alpha_i^* + \alpha_i) - \sum_{i=1}^N y_i (\alpha_i^* - \alpha_i) \quad (5)$$

where K is the kernel function defined as $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$; $\varphi(x)$ is the transformation that maps x into a high dimensional space subject to constraints (6):

$$\sum_{i=1}^N (\alpha_i^* - \alpha_i) = 0, \quad 0 \leq \alpha_i, \alpha_i^* \leq C, \quad i = 1, \dots, N. \quad (6)$$

Using the Kernel function $K(x_i, x)$, the optimum form of linear regression function can be written as (7):

$$f(x, w) = \sum_{i=1}^N (\alpha_i^* - \alpha_i) K(x_i, x) + b. \quad (7)$$

The most commonly used kernel functions are: Linear $K(x_i, x_j) = x_i^T x_j$, Polynomial $K(x_i, x_j) = (x_i^T x_j + 1)^p$, and Radial Basis Function (RBF) $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$, where γ is the parameter of the kernel function.

Usually, linear and polynomial kernels require less computational effort but provide lower accuracy than the RBF kernels. Given the high cost of the initial data collection [5], it is initially

assumed that accuracy is considered to be more important than modelling cost for this application, and so a nonlinear SVM is used. Following [4, 10], we begin with non-linear SVM, using RBF as the kernel function.

Support Vector Machine algorithms are not scale invariant, so it is good practice to scale the data. We standardize features by subtracting the mean and scaling to unit variance across each data set examined.

2.2 Non-linear SVR with grid search and cross validation

When training an SVR with the RBF kernel, two key parameters must be selected: C and γ [10]. The parameter C , common to all SVM kernels, trades off misclassification of training examples against the simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly. γ controls local model sensitivity.

Suitable choices for C and γ are critical to SVR modeling performance. Accordingly, in this study, a grid search method with cross-validation is used to select appropriate values. Various pairs of (C, γ) values are tried and the one with the best cross-validation accuracy is chosen. For an initial search, a logarithmic grid using base 10 is often helpful. Using base 2, finer tuning can be achieved but at a higher computational cost.

In SVR model development there is no general rule for selecting the ranges of the parameters (C, γ) . In practice, if the best parameter values are found to lie on the boundaries of the grid, the search can be extended in that direction over further iterations. In our numerous experiments on the benchmark data set, the optimal values of C and γ were always found to lie within the following ranges: $C = 2^{-3}, \dots, 2^{10}, \gamma = 2^{-10}, \dots, 2^3$.

We applied these exponentially increasing sequences of C and γ , where the exponent step was set to 0.1. The meshing of the variable pairs (C, γ) is shown in Table 1.

Table 1. The meshing of variable pairs (C, γ)

C	γ				
	2^{-10}	$2^{-9.9}$...	$2^{2.9}$	2^3
2^{-3}	$(2^{-3}, 2^{-10})$	$(2^{-3}, 2^{-9.9})$...	$(2^{-3}, 2^{2.9})$	$(2^{-3}, 2^3)$
$2^{-2.9}$	$(2^{-2.9}, 2^{-10})$	$(2^{-2.9}, 2^{-9.9})$...	$(2^{-2.9}, 2^{2.9})$	$(2^{-2.9}, 2^3)$
...
$2^{9.9}$	$(2^{9.9}, 2^{-10})$	$(2^{9.9}, 2^{-9.9})$...	$(2^{9.9}, 2^{2.9})$	$(2^{9.9}, 2^3)$

2^{10}	$(2^{10}, 2^{-10})$	$(2^{10}, 2^{-9.9})$...	$(2^{10}, 2^{2.9})$	$(2^{10}, 2^3)$
----------	---------------------	----------------------	-----	---------------------	-----------------

Following the procedure used in our earlier paper [5], the full dataset (subset 1) was used for final evaluation of all models, i.e. it was a test dataset. For data subsets 2, 3 and 4 the full dataset was excluded from the process of SVR model training and selection of its optimal parameters. Note: for brevity we will refer to these data subsets as datasets 1, 2, 3 and 4. Also following [5], we use the mean absolute error MAE, which is the mean of the absolute differences between target and predicted values, as the cost function for the optimization problem. This allows a direct comparison between the SVR and neural networks model performance on the benchmark datasets.

For each of the datasets 1, 2, 3, 4, we used Leave-One-Out Cross-Validation (LOOCV) [14], which gives unbiased estimate of the true accuracy.

In this technique, each data point is excluded in turn, and used as a single point validation set for a mode trained on the remaining data set. The resulting error is averaged over all N trials to get the total effectiveness of the corresponding model. Every data point is used in a validation set exactly once, and is used in a training set $N - 1$ times.

In practice [14], $k = 5, 10$ or 20 are often used instead of $k = N$ as these k -fold cross validations give approximately the same accuracy estimation as LOOCV but with a reduced computational burden. However here it is assumed that obtaining the data points in a calibrated flow lab is likely to be expensive compared with examining N -fold cross validations, and so we apply full LOOCV.

2.3 Linear SVR with augmented parameters

The dimensionality of the dataset is small, which makes it possible that the powerful nonlinear SVR method is under-constrained and is able to learn the dataset perfectly. This risk of over fitting the model prompted us to explore the use of the Linear SVR method with augmented parameters as an alternative technique.

The dataset has only two features: mass flowrate (x_1) and density drop (x_2). Polynomial feature adding is a type of feature engineering [15], i.e. the creation of new input features, based on the existing features. New data columns may be added by including the squares of the original inputs x_1^2 , x_2^2 , as well as their product $x_1 \cdot x_2$. A squared or cubed version of input variables can help some machine learning algorithms make better predictions, and typically linear algorithms respond well to the use of polynomial input variables. Adding polynomial terms to the linear model can be

an effective way of allowing the model to identify nonlinear patterns [15]. Typically a small degree is used such as 2 or 3 because for larger values the model may once again be prone to over fitting.

The most important parameter of Linear SVR is C. From experience, we use a grid over the range $C = 2^{-4}, \dots, 2^7$.

As in the previous section, we use the full dataset (subset 1) for final evaluation, i.e. it is used as a test dataset. We use the mean absolute error MAE metric to evaluate modelling performance, in order to provide a direct comparison between the SVR and neural networks model performance on the benchmark datasets reported previously. We also applied LOOCV, which gives an unbiased estimate of the true accuracy.

3. Experimental results

For each of data sets 1-4, two SVR models (8), (9) using the RBF kernel,

$$MFR_{error} = f(MFR_{obs}, DD_{obs}) \quad (8)$$

$$DD_{error} = g(MFR_{obs}, DD_{obs}) \quad (9)$$

predicting the mass flow error MFR_{error} and the density drop error DD_{error} using the observed values MFR_{obs}, DD_{obs} , respectively, were trained to determine the best parameters (C, γ) using grid search and LOOCV, as described in Section 2. All datasets and program code, along with one of the resulting architectures, are available on the website [7].

The obtained MAE values are presented in Tables 2 and 3.

Table 2. Mass flow error RBF-SVR model performance

Dataset 1		Dataset 2		Dataset 3		Dataset 4	
(C, γ)	MAE	(C, γ)	MAE	(C, γ)	MAE	(C, γ)	MAE
(274.37, 0.13)	0.74	(34.30, 0.10)	1.06	(84.45, 0.06)	1.18	(25.99, 0.10)	1.28

Table 3. Density drop error RBF-SVR model performance

Dataset 1		Dataset 2		Dataset 3		Dataset 4	
(C, γ)	MAE	(C, γ)	MAE	(C, γ)	MAE	(C, γ)	MAE
(78.79, 0.08)	0.38	(7.46, 0.44)	0.45	(955.43, 0.17)	0.66	(9.85, 0.07)	0.50

In addition, Linear SVR models with added polynomial features were developed in similar way. Instead of just two features (mass flowrate x_1 and density drop x_2), we used an extended set of 10 features: 1, x_1 , x_2 , x_1^2 , $x_1 \cdot x_2$, x_2^2 , x_1^3 , $x_1^2 \cdot x_2$, $x_1 \cdot x_2^2$, x_2^3 .

Note that the inclusion of the constant “1” as a feature allows the introduction of a fixed offset to improve the overall fit of the model.

The corresponding optimal C values and MAE are shown in Tables 4, 5.

Table 4. Mass flow error, augmented Linear SVR model performance

Dataset 1		Dataset 2		Dataset 3		Dataset 4	
C	MAE	C	MAE	C	MAE	C	MAE
0.10	0.85	1.32	0.87	0.10	0.92	0.29	0.95

Table 5. Density drop error, augmented Linear SVR model performance

Dataset 1		Dataset 2		Dataset 3		Dataset 4	
C	MAE	C	MAE	C	MAE	C	MAE
1.32	0.35	2.83	0.38	0.22	0.42	1.02	0.43

Tables 6 and 7 show MAE achieved for each of the data sets for RBF-SVR, Linear SVR and the smallest MAE achieved for ANN from [5].

Table 6. MAE value for SVR and best MAE value for ANN [5] models for MFR error correction

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
RBF-SVR	0.74	1.06	1.18	1.28
Augmented Linear SVR	0.85	0.87	0.92	0.95
ANN	0.87	0.91	0.90	1.72

Table 7. MAE value for SVR and best MAE value for ANN [5] models for density drop error correction

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
RBF-SVR	0.38	0.45	0.66	0.50
Augmented Linear SVR	0.35	0.38	0.42	0.43
ANN	0.35	0.51	0.71	1.23

The obtained results suggest that the SVR models are comparable and often superior to ANN models. As can be seen, the augmented Linear SVR is generally superior to the nonlinear SVR. It is also interesting to note that the SVR method appear to be less sensitive to data decimation than the neural network technique. The error values remain approximately the same for datasets 2-4, in contrast to the case of neural networks, which gave a significantly overestimated error value for the smallest dataset 4. This is a potentially useful finding, given the expense of experimental data collection.

Note also the smaller errors on dataset 2 in comparison with dataset 3, which suggests that, in this example, better results are obtained using more flow lines with fewer GVF points than to have fewer flow lines with a large number of GVF points.

Figures 1 – 4 show the residual errors for the non-linear SVR (Figures 1 and 2), and augmented Linear SVR (Figures 3 and 4), in each case for the full dataset. While most data points seem well-modelled, larger residual errors are observed for the lines at 0.4 kg/s and 0.8kg/s for both mass flow and density, so that at higher data drops the 0.4 kg/s lines have relatively large negative errors while the 0.8 kg/s lines have relatively large positive errors. There are two aspects to this issue, relating firstly to the underlying raw measurements, and then to the modelling technique. From a Coriolis metering perspective, it is well-established both through experiment and modelling (e.g. see [5] and its references), that the mass flow and density errors are typically of larger magnitude, and show more local variation, at low flows, arising from the relative influence of various error sources at different liquid flow rates. Specifically, for this data set, it can be seen in figures 2 and 4 of [5] that the raw errors in the lowest three flow lines exhibit what may be informally characterized as ‘different’ behavior from the rest of the dataset, in terms of slope and smoothness. Accordingly, the SVM modelling, if prevented from over-fitting the data, is likely to reflect this greater variation in the corrected data at the lower flows. It is possible that additional parameterization might result in improved models that are better able to predict these low flow

characteristics, and this will be considered in a future paper in the context of the wider problem of three-phase flow.

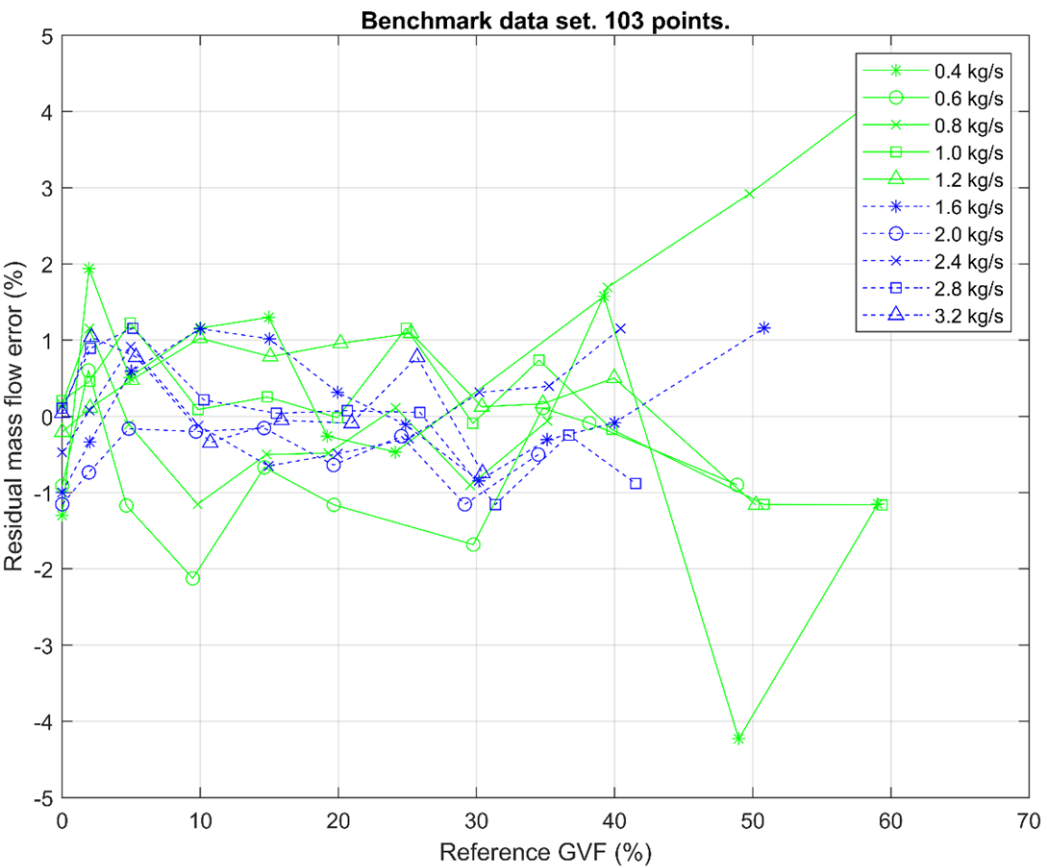


Figure 1: Residual mass flow errors for the RBF-SVR model based on dataset 1

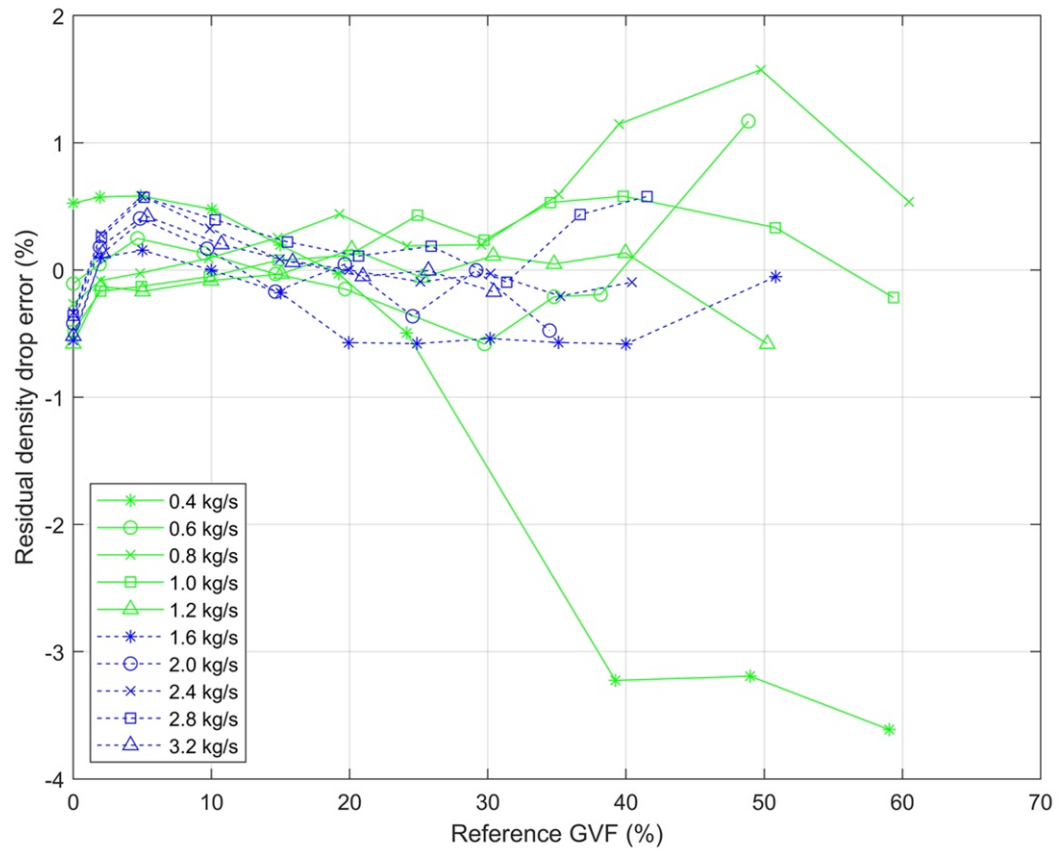


Figure 2: Residual density drop errors for the RBF-SVR model based on dataset 1

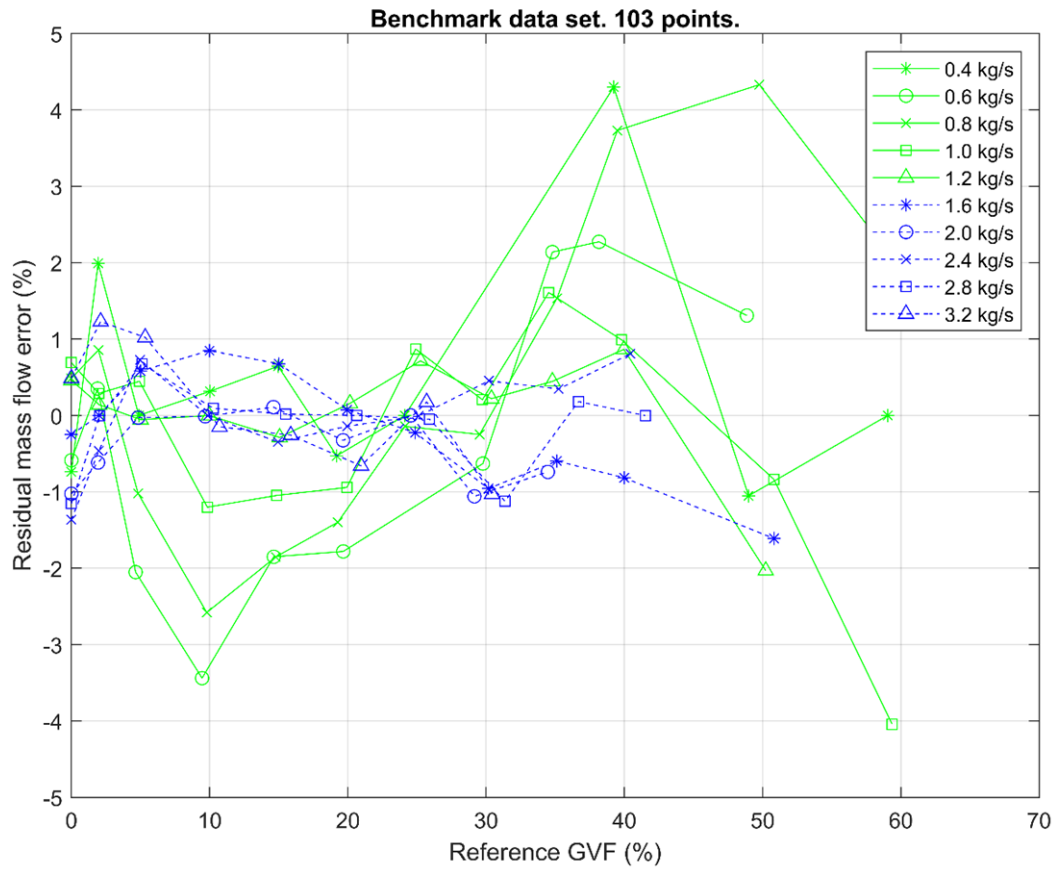


Figure 3: Residual mass flow errors for the augmented linear SVR model based on dataset 1.

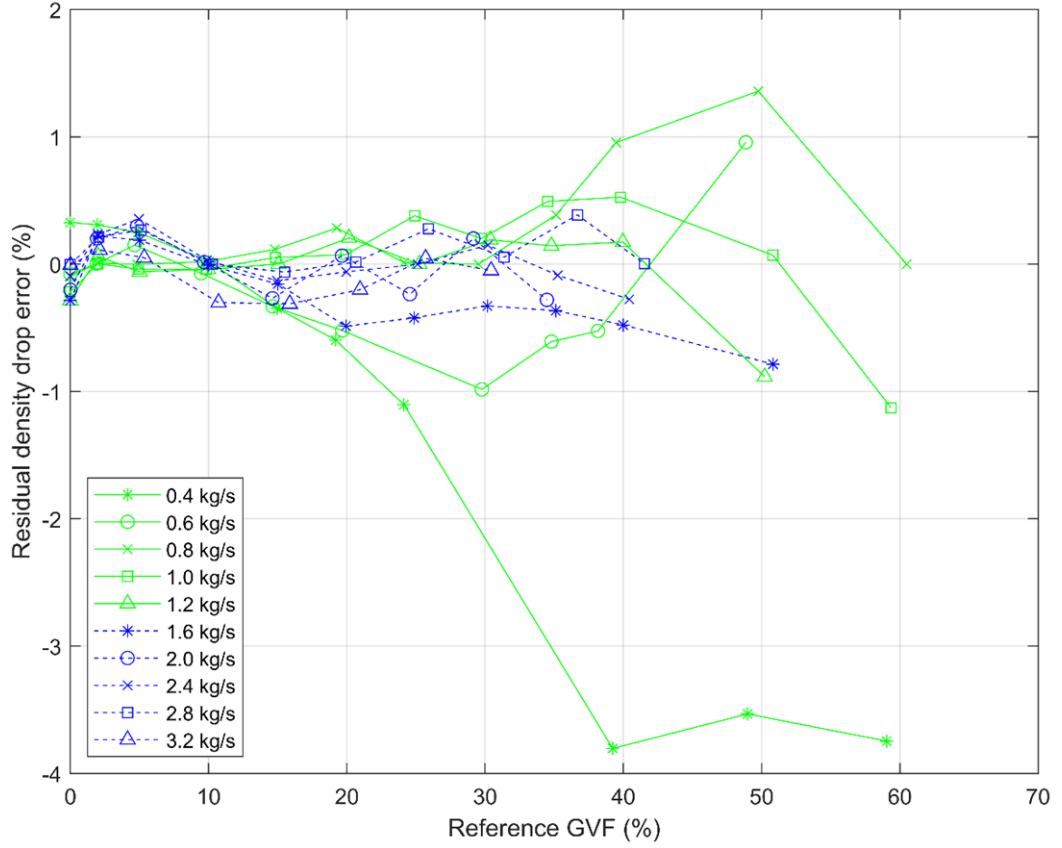


Figure 4: Residual density drop errors for the augmented linear SVR model, based on dataset 1

4. Conclusions and Future Work

In the paper, the Support Vector Regression algorithm has been applied to the benchmark dataset [7] for two-phase Coriolis metering obtained from a 50 mm Coriolis mass flow meter. The results of using SVR models for mass flow and density drop error correction show its advantage over the ANN models of the 2-N-1 architecture considered in [7].

As can be seen from Tables 6 and 7, SVR gives results comparable to a neural network for the full dataset 1 consisting of 103 data points. Mean absolute error value for RBF-SVR and ANN models for MFR error correction are 0.74 and 0.87, respectively. However, for the smallest dataset 4 (27 points), the accuracy of SVR significantly exceeds the accuracy of neural networks. Mean absolute error values for Linear SVR and ANN models of MFR correction are 0.95 and 1.72, respectively.

It is also worth noting that the augmented Linear SVR is less sensitive to data decimation than non-linear SVR and generally superior to it for small datasets 2, 3, 4. Most likely, for a small dataset, nonlinear SVM or neural networks is a too powerful method that gives an over fitted model. Such an observation is important, given the high cost of experimental data collection.

In future work, large benchmark data sets with three-phase data (oil/water/gas) will be provided, alongside proposed modelling methods and results.

References

1. 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. <https://doi.org/10.1016/j.flowmeasinst.2018.02.017>
2. 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, [https://doi.org/10.1016/S0955-5986\(00\)00045-5](https://doi.org/10.1016/S0955-5986(00)00045-5)
3. 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, <https://doi.org/10.1016/j.flowmeasinst.2013.01.003>.
4. L. Wang, J. Liu, Y. Yan, 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 Transactions on Instrumentation and Measurement*, Vol. 66, No. 5 (2017) 852-868, DOI: 10.1109/TIM.2016.2634630
5. O.L. Ibryaeva, V.V. Barabanov, M.P. Henry, M. Tombs, F. Zhou, A benchmark data set for two-phase Coriolis metering, *Flow. Meas. Instrum.* 72 (2020), <https://doi.org/10.1016/j.flowmeasinst.2020.101721>
6. M.S. Tombs, F.B. Zhou, M.P. Henry, “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>
7. <https://cmfdata.susu.ru>. The site manager is Olga Ibryaeva: ibriaevaol@susu.ru. O.L. Ibryaeva et al.
8. A. Ben-Hur, D. Horn, H.T. Siegelmann, V. Vapnik, Support vector clustering, *J. Mach. Learn. Res.* 2 (2001) 125–137
9. L. Ma, H. Zhang, H. Zhou, Q. He, Massflow measurement of oil water two-phase flow based on Coriolis flow meter and SVM, *J. Chem. Eng. Chin. Univ.* 21 (2007)
10. J. Yue, K. Xu, W. Liu, J. Zhang, Z. Fang, L. Zhang, H. Xu, SVM based measurement method and implementation of gas-liquid two-phase flow for CMF, *Measurement* 145 (2019) 160-171 <https://doi.org/10.1016/j.measurement.2019.05.051>
11. H. Drucker, C. Burges, L. Kaufman, A. Smola, V. Vapnik, "Support Vector Regression Machines", *Advances in Neural Information Processing Systems* 9, NIPS 1996, P. 155–161.

12. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
13. Kohavi R., A study of cross-validation and bootstrap for accuracy estimation and model selection, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. San Mateo, 1995. Vol. 2, № 12, P. 1137–1143.
14. James G., Witten D., Hastie T., Tibshirani R., An Introduction to Statistical Learning: with applications in R, Springer, 426 p. 2013
15. M. Kuhn, K. Johnson, Feature Engineering and Selection: A Practical Approach for Predictive Models, Chapman and Hall/CRC, 310 p., 2019.