www.s2is.org
International Journal on Smart Sensing and Intelligent Systems
ISSN 1178-5608

DEVELOPMENT OF SOFT SENSOR TO ESTIMATE MULTIPHASE FLOW RATES USING NEURAL NETWORKS AND EARLY STOPPING

Tareq Aziz AL-Qutami[*1], Rosdiazli Ibrahim[1], Idris Ismail[1], and Mohd Azmin Ishak[2]

[1] Department of Electrical & Electronics Engineering, Universiti Teknologi Petronas, 32610, Perak, Malaysia

[2] Project Delivery & Technology, Petronas Nasional Berhad, 50050, Kuala Lumpur, Malaysia

Emails: *tareqaziz2010@gmail.com

Submitted: Jan. 25, 2017 ***Accepted: Feb. 1, 2017*** ***Published: Mar. 1, 2017***

Abstract- This paper proposes a soft sensor to estimate phase flow rates utilizing common measurements in oil and gas production wells. The developed system addresses the limited production monitoring due to using common metering facilities. It offers a cost-effective solution to meet real-time monitoring demands, reduces operational and maintenance costs, and acts as a back-up to multiphase flow meters. The soft sensor is developed using feed-forward neural network, and generalization and network complexity are regulated using K-fold cross-validation and early stopping technique. The soft sensor is validated using actual well test data from producing wells, and model performance is analyzed using cumulative deviation and cumulative flow plots. The developed soft sensor shows promising performance with a mean absolute percent error of around 4% and less than 10% deviation for 90% of the samples.

Index terms: Multiphase Flow, Soft Sensor, Virtual Flow Meter, Neural Network, early stopping.

I. INTRODUCTION

Soft Sensors or sometimes called “Inferential Sensors” are computational models that can provide real-time approximations of hard-to-measure or economically inconceivable process variables [1]. They have been widely implemented in the last decades in many industries including refineries, chemical plants, power plants, polymer industry, healthcare etc. [1-6]. Soft sensors are valuable when desired-to-measure variables have long delays, or physical sensors cannot be deployed due to hostile environment or financial considerations.

Soft sensors utilize readily available measurements or cheaper sensors to infer or estimate the desired variables. For example, using temperature and pressure measurements to estimate gas quality (composition) in real-time in-between laboratory analysis runs [7], estimation of rocks porosity and permeability for uncored wells by correlating well logs and core data [8], and predication of drum pressure and level in coal-fired power plants using heat and flow rates [9], etc.

There are generally two approaches for soft sensor development. The first approach is using Mechanistic modeling which is based on physical and chemical laws that describes system behaviors and formulation of mathematical relationships among system variables. Assumptions are usually required to simplify the complexity of such models [1]. Second approach is data-driven modeling which exploits the actual measurements available to identify the soft sensor model employing multivariate statistics and soft-computing techniques [10].

Several data-driven techniques have been used to design soft sensors such as principle component analysis (PCA) [11], partial least squares (PLS) [3], support vector machine (SVM) [12,13], neural networks (NN) [14,15], and ensemble methods [16].

An area where soft sensors are of great benefits is multiphase fluid flow in petroleum and petrochemical industries. Multiphase flow is a flow of several components such as oil, gas and water in the same pipe. Two-phase flow of oil and gas is the most common multiphase flow in oil and gas production lines. Measuring individual phase flow rates (oil phase flow rate and gas phase flow rate) is of great importance for production monitoring, optimization and reservoir management [17].

Conventional techniques to physically measure phase flow rates are using common test separator or common Multiphase Flow Meter (MPFM). This approach rotates and samples each production well regularly in a weekly or monthly intervals. This however does not provide real-time monitoring and usually leads to late diagnosis and correction actions of production issues [18]. The use of a common physical meter for several production wells is primarily due to the high costs associated with dedicated test separators or MPFMs [17,19]. Furthermore, MPFM requires high operating and maintenance costs due to the complexity of instruments used within the meter. This complexity is also a source of measurement uncertainty and error propagation; hence frequent calibration is necessary [20].

Emerging soft sensor technology used to estimate multiphase flow rates, sometimes referred to as Virtual Flow Metering (VFM), is often based on empirical correlations [21] or mechanistic approach [22]. Mechanistic models are however highly dependent on fluid properties and production regimes. Hence they are computationally expensive and are sensitive to changing Gas-Oil ratio (GOR) [23,24]. Consequently, they require extensive tuning on filed data [17].

In this article, a soft-computing technique based on neural networks is proposed to design a soft sensor for multiphase flow rates estimation in oil and gas production pipes. This soft sensor can augment MPFM measurements and act as a backup when it fails. It also can provide estimations between well test runs if a common MPFM or test separator is installed. This could ultimately reduce operational and maintenance costs as well as support integrated operations [23,25].

II. NEURAL NETWORKS

Neural Network (NN) is a machine learning technique that mimics cognitive abilities of biological nervous system. The network consists of units called neurons which are arranged in layers and are interconnected to each other through weights as illustrated in Figure 1.

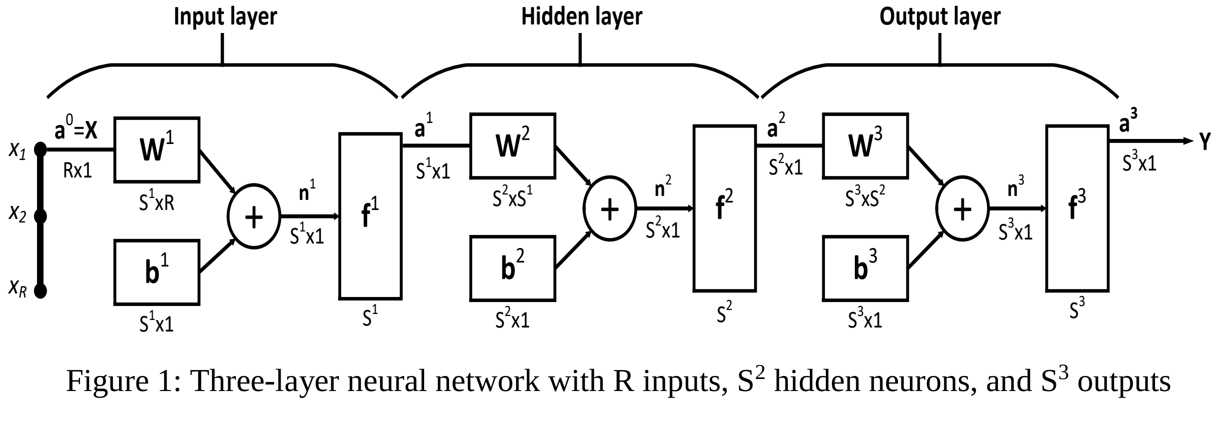


Figure 1: Three-layer neural network with R inputs, S^2 hidden neurons, and S^3 outputs

Every neuron performs a computational task based on the inputs received and generate output to next neurons. The net input to neuron i in layer $v+1$ is calculated by:

$$n^{v+1}(\mathrm{i}) = \sum_{j=1}^{S^v} w^{v+1}(i,j)a^v(j) + b^{v+1}(i) \qquad (1)$$

Where, w is neuron weight, b is neuron bias, and a is input from previous neurons 1 to S^v. The output of neuron i will be:

$$a^{v+1}(\mathrm{i}) = f^{v+1}\left(n^{v+1}(i)\right) \qquad (2)$$

For V-layer network, layer output can be represented in matrix form as:

$$\boldsymbol{a}^{v+1} = \boldsymbol{f}^{v+1}(\boldsymbol{W}^{v+1}\boldsymbol{a}^v + \boldsymbol{b}^{v+1})$$
$$v = 0,1,\ldots,V-1 \qquad (3)$$

The transfer function f in hidden layer is usually a nonlinear function such as sigmoid function given by:

$$f(n) = \frac{2}{1+e^{(-2n)}} - 1 \qquad (4)$$

While output transfer function is usually linear for regression problems.

Feed-forward neural network is the most famous architecture and is considered a universal function approximator [26]. The parallel distributed structure of neural network has several advantages including high nonlinearity, capability of input-output mapping, adaptable learning, fault tolerance, and generalization [27].

Several neural network training algorithms are available in the literature. The most popular ones are backpropagation based-algorithms such as Levenberg Marquardt algorithm [28].

Neural networks are nevertheless prone to over-fitting [28,29]. Two regularization approaches are available to avoid over-fitting and improve generalization capabilities. The first is training regularization or commonly known as early stopping. While the second approach is performance function regularization which modifies the performance function to promote simplicity such as L2 regularization. This article will adapt the first approach, namely early stopping, to develop a neural network soft sensor.

III. PROPOSED METHODOLOGY

Neural network is proposed to develop multiphase flow soft sensor that can predict phase flow rates in multiphase production lines. The model is trained using actual well test data, and K-fold cross-validation is used to validate the model performance. Flow chart of the proposed methodology is shown in Figure 2. First, data is preprocessed and divided into training, validation, and testing sets.

Neural network architecture is then selected and trained. Lastly the model is tested, and its performance is analyzed.

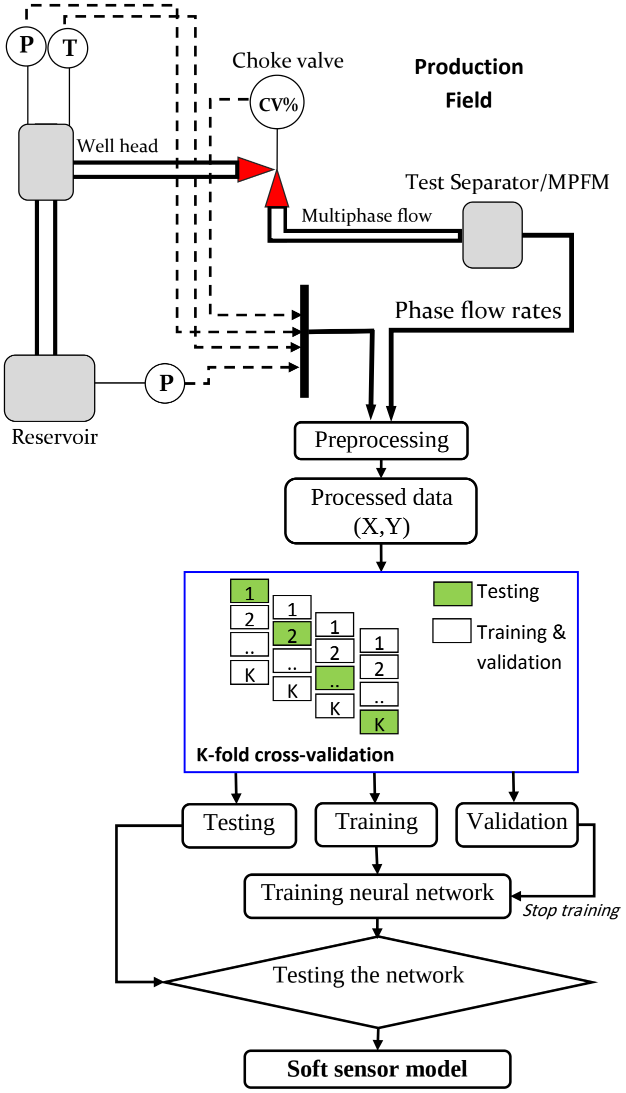


Figure 2. Flow chart of proposed soft sensor methodology

a. Data Collection and Preprocessing

In order to develop a soft sensor to estimate gas and oil flow rates, representative measurements that are correlated to the desired variables have to be identified. Typical measurements that often exists in production fields and are correlated to multiphase flow rates are pressure (P) and temperature (T) down-hole the reservoir, pressure and temperature at the well-head, and choke valve opening percentage, which controls the amount of multiphase production. Figure 2 shows positions of these measurements in production field.

Once these measurements are collected, they are normalized and checked for any outliers using Turkey boxplot and Z-score. If there are potential outliers, they will be investigated further. If proven to be faulty or represents irregular rare behavior, such as shut-in or water flashing, they will be removed.

Processed data containing m samples are expressed as $D = \{(x_i, y_i)\}_{i=1}^{m}$, where $x_i \in \mathcal{R}^{v\times 1}$ are inputs (v is the number of input variables) and $y_i \in \mathcal{R}^{2\times 1}$ are outputs (gas flow rate and oil flow rate). K-fold cross-validation is used to split the data into training set D_{train}, validation set D_{val}, and testing set D_{test}. Data are partitioned into K folds. Each fold is used as testing set while the remaining K-1 folds are divided into training set (80%) and validation set (20%). The folds are rotated to generate K training/validation/testing sets as illustrated in Figure 2.

b. Parameter Selection and Model Training

The neural network architecture considered in this article contains one hidden layer with hyperbolic tangent sigmoid transfer function represented by equation (4), and linear output layer transfer function. Nguyen-Widrow method [30] is used for weight initialization to ensure even distribution across input space. Furthermore, number of neurons is selected using K-fold cross-validation in order to ensure the network is complex enough to represent the underlying patterns within the data, and not too complex to avoid over-fitting. Sum of squares of network errors (SSE) is used in this article as the performance function for training the feed-forward neural network and can be represented by:

$$E(x,w) = \frac{1}{2}\sum_{p=1}^{P}\sum_{m=1}^{M} {e_{p,m}}^2$$

$$e_{p,m} = y_{p,m} - \hat{y}_{p,m} \qquad (5)$$

Where x is inputs, w is network weights, P is the number of training samples, M is the number of outputs, and $e_{p,m}$ is training error when applying sample p and is the difference between desired output y_m and current network output $\hat{y}_m$.

One way to minimize performance function is using Steepest Descent (or Gradient Descent) algorithm which uses first-order gradient vector of performance function $g = \frac{\partial E(X,W)}{\partial W}$. Update rule of steepest decent in matrix form is:

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k - \alpha \boldsymbol{g}_k \qquad (6)$$

Where, α is learning rate. However, the training process of the steepest descent algorithm converges asymptotically and takes many iterations to get desired performance. More efficient training is accomplished using second-order methods such as Newton's method which assumes weights are linearly independent and all gradients are functions of weights. Newton's method relies on second-order derivatives of performance function Hessian matrix $\boldsymbol{H}$, which is computationally expensive. Update function of Newton's method is given by:

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k - \boldsymbol{H}_k^{-1}\boldsymbol{g}_k \tag{7}$$

Gauss-Newton algorithm alternatively uses Jacobian matrix $\boldsymbol{J}$ that involves first derivatives only to approximate Hessian matrix $\boldsymbol{H}$ [31]. Hessian matrix $\boldsymbol{H}$ can be approximated by:

$$\boldsymbol{H} \approx \boldsymbol{J}^T\boldsymbol{J} \tag{8}$$

While gradient vector $\boldsymbol{g}$ can be computed by:

$$\boldsymbol{g} = \boldsymbol{J}\boldsymbol{e} \tag{9}$$

Where, $\boldsymbol{e}$ is network error vector and has the following form:

$$\boldsymbol{e} = \left[e_{1,1}, e_{1,2}, e_{1,3} \dots, e_{1,M}, \dots, e_{P,1}, e_{P,2}, e_{P,3}, \dots, e_{P,M}\right]^T \tag{10}$$

The weights update rule of Gauss-Newton method becomes:

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k - (\boldsymbol{J}_{\boldsymbol{k}}^T\boldsymbol{J}_{\boldsymbol{k}})^{-1}\boldsymbol{J}_k\boldsymbol{e}_k \tag{11}$$

Even though Gauss-Newton method does not require computing Hessian matrix, it still faces convergence issues just like Newton method for complex error surfaces, i.e. $\boldsymbol{J}^T\boldsymbol{J}$ matrix may not be invertible. In order to ensure $\boldsymbol{J}^T\boldsymbol{J}$ is invertible, Levenberg Marquardt algorithm modifies the Hessian equation to:

$$\boldsymbol{H} \approx \boldsymbol{J}^T\boldsymbol{J} + \boldsymbol{\mu}\boldsymbol{I} \tag{12}$$

Where, $\boldsymbol{\mu}$ is positive combination coefficient, and $\boldsymbol{I}$ is the identity matrix.

The Levenberg–Marquardt update rule hence becomes:

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k - (\boldsymbol{J}_{\boldsymbol{k}}^T\boldsymbol{J}_{\boldsymbol{k}} + \boldsymbol{\mu}\boldsymbol{I})^{-1}\boldsymbol{J}_k\boldsymbol{e}_k \tag{13}$$

μ is decreased every step the performance function reduces and is incremented when the performance function increases. When μ is nearly zero, the update rule becomes just like Gauss-Newton which is faster and more accurate near error minimum. And when μ is large, the algorithm behaves like steepest descent with a small step size [32]. This way, Levenberg Marquardt combines both steepest descent and Gauss-Newton algorithms.

Several networks with different number of hidden neurons are trained using Levenberg Marquardt algorithm and then evaluated on validation folds of K-fold cross-validation. The number of neurons that generates the minimum validation error is selected. Early stopping is implemented to regularize the training process and avoid over-fitting [33]. Validation error is tracked during network training and whenever the validation error starts growing, the training is terminated. In this study, the validation error is checked every epoch. When the error increases 10 times the training is stopped. This way, training process will usually stop before reaching maximum number of iterations. Therefore, early stopping not only avoids over-fitting, it is also more efficient and guarantees better generalization.

c. Model Testing and Validation

After selecting the architecture and parameters of neural network, the network is trained using the training folds. Then testing is carried out on the K testing folds. Then average performance and generalization capability of the developed model are reported. Furthermore, variable importance is checked, where input measurements influence on soft sensor performance is examined. This is important as some of these measurements are not always available or frequently fail.

Performance indicators used in this article are mean squared error (MSE), mean absolute percentage error (MAPE), minimum MSE, maximum MSE, and MSE standard deviation. MSE and MAPE are calculated as follow:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \qquad (14)$$

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \qquad (15)$$

Where m is number of samples, y_i are actual outputs and $\hat{y}_i$ are predictions by the developed model. Besides these generic performance indicators, two other indicators are of great significance to validate a virtual flow meter model, namely cumulative flow plot and cumulative deviation plot. Cumulative flow represents the sum of all previous points up to the current point. It helps identify if the model has any systematic offset or drift. While cumulative deviation plot, recommended by Norwegian Society for Oil and Gas Measurement [34], indicates the percentage of test points that fulfil certain deviation criteria.

IV. RESULTS AND DISCUSSION

In this section, results of the proposed NN soft sensor are presented. Data used to develop the model consists of 591 samples, and was collected over a period of 1.5 years. Flow can be considered two-phase since Water Cut (WC) only represents a maximum of 1% of total flow rate. The measurements available as inputs to the soft sensor model are choke valve opening percentage (CV%), well-head pressure (WHP), well-head temperature (WHT), and bottom-hole pressure (BHP). While the outputs that will be predicted by the soft sensor are gas flow rate (Q_{gas}) and oil flow rate (Q_{oil}). The soft sensor will consist of two neural networks to estimate gas and oil flow rates separately. Table 1 shows the characteristics of measurements used to develop the soft sensor. Where, STD is standard deviation.

Table 1. Characteristic of dataset

Measurement	Mean	Min	Max	STD
Choke valve (%)	24.3	10.9	34.3	5.6
Well-head pressure (bar)	279.1	240.2	320.1	20.8
Well-head temperature (°C)	63.8	61.7	65.5	0.8
Bottom-hole pressure (bar)	425.3	417.7	435.2	4.5
Gas flow rate (mmscf/d)	74.4	47.2	88.3	9.3
Oil flow rate (bbl/d)	9583.4	4712.9	14190.0	2248.6

a. Data Pre-processing

Data set is first normalized to the range [-1, 1] using min-max normalization:

$$y = \frac{2(x-x_{min})}{(x_{max}-x_{min})} - 1 \tag{16}$$

Figure 3 shows Tukey boxplot of the normalized data with coverage percent of 97%.

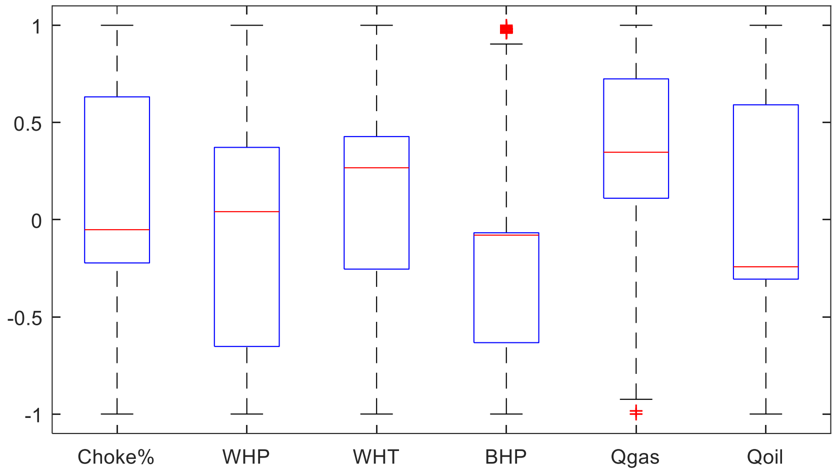


Figure 3. Boxplot of normalized data set

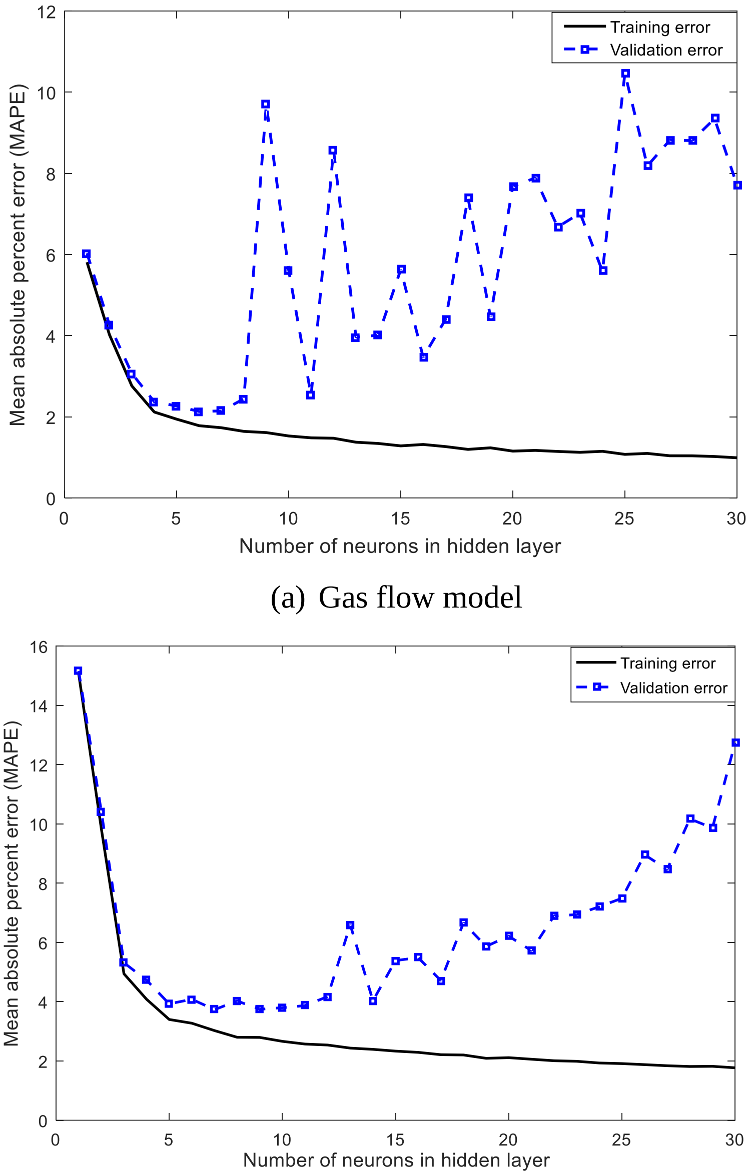
In the boxplot, potential outliers are represented by red plus signs. There are 48 points identified as potential outliers (46 in BHP and 2 in Q_{gas}). Z-score is used to confirm if a point is an outlier that should be removed. If Z-score is greater than 3.0, then a point is considered outlier and is removed from the dataset. Z-score is calculated using the following formula:

$$Z_{score} = \frac{X - mean(X)}{STD(X)} \tag{17}$$

Z-scores intervals are [-1.69, 2.17] and [-2.91, 1.50] for BHP and Qgas respectively. It is observed that none of these potential outliers are outside the interval [-3, 3]. Hence, none of these points is removed, and the complete data set is used to develop and validate the neural network model.

b. Parameter Selection

The dataset is divided into training set, validation set and testing set according to subsection III.a, where K is set to 10 in the K-fold cross-validation. The validation set is used to decide the number of hidden neurons in the neural network. The range of candidate values are from 1 to 30 with increment of 1. Figure 4 shows the validation and training MAPE errors of neural network with different number of neurons for gas flow rate (a) and oil flow rate (b). Values represent average of the 10 folds in K-fold cross-validation. MAPE is presented rather than MSE due to clearer trend visualization when MAPE is plotted.



(a) Gas flow model

(b) Oil flow model

Figure 4. Number of neurons VS network performance. (a) Gas flow model. (b) Oil flow model.

It is observed that the validation error decreases in the beginning indicating improved generalization. However, at some point the error starts increasing as the number of neurons increases even though training error is still reducing. This indicates over-fitting and loss of generalization. Minimum validation MAPE of 2.13 and 3.73 occur when number of hidden neurons are 6 and 7 for gas and oil flow rate models respectively. Note that testing error is not used to select number of neurons as it can only be used to validate generalization capability of the model and should not be used for parameter selection process.

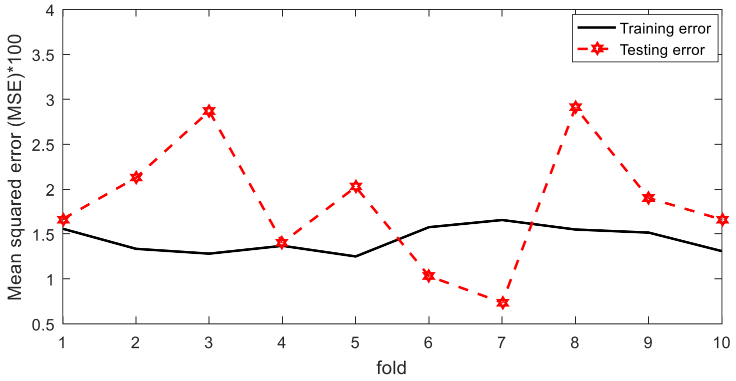
c. Neural Network Model Performance

After determining numbers of neurons in hidden layer. Two NNs are developed to estimate gas and oil flow rates based on the available inputs: BHP, WHP, WHT and CV%. Maximum training epochs is set to 500. Number of hidden neurons are set to 6 and 7 for gas and oil flow models respectively. In order to eliminate influence of data variability, performance of 10-folds cross-validation are averaged and reported in Table 2. Average of the 10 folds MSE minimum, MSE maximum and MSE standard deviation (STD) are also reported for both training and testing sets. MSE, Min, Max and STD values are multiplied by 100 for clearer presentation.

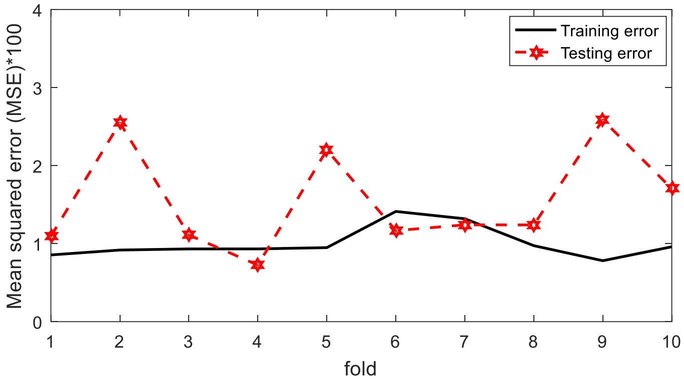
Table 2. Average performance on 10 folds training and testing sets. MSE, Min, Max and STD are multiplied by 100

Model	**Data set / indicator**	**MAPE**	**MSE**	**Min**	**Max**	**STD**
Gas flow	Training	2.09	1.44	1.25	1.66	0.15
	Testing	2.27	1.83	0.74	2.91	0.70
Oil flow	Training	3.65	1.00	0.78	1.41	0.20
	Testing	4.22	1.56	0.72	2.59	0.66

MAPEs of the networks are below 5% on testing sets and this is considered an excellent performance. MSEs and standard deviations, using different data divisions, are also small indicating the models are able to generalize well. Influence of data variability can be observed in Figure 5 where the trend of testing and training MSEs of the 10 folds are shown. It can be seen that performance is within acceptable standard deviation across all folds, 0.0070 and 0.0066 STD for gas and oil flow models respectively.



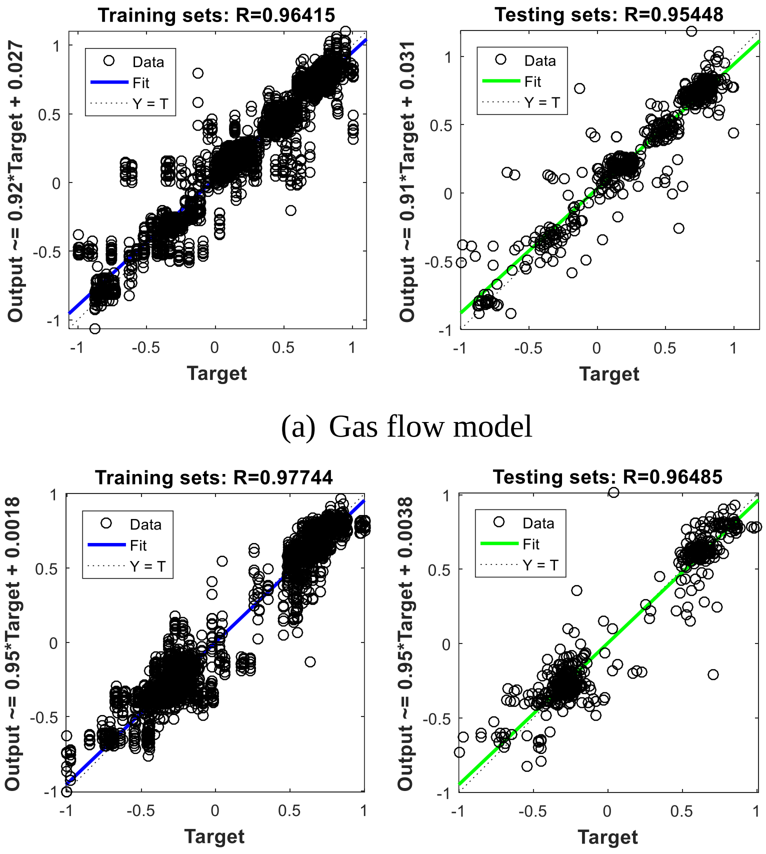
(a) Gas flow model



(b) Oil flow model

Figure 5. Training and testing MSE of all 10 folds. (a) Gas flow model. (b) Oil flow model.

Figure 6 shows regression line of predictions relative to outputs, and coefficient of determination (R^2) for the 10 training and testing folds combined. R^2 is close to 1.0 for both training and testing sets confirming that the developed models are able to explain most of the variance within gas and oil flow rate trends.

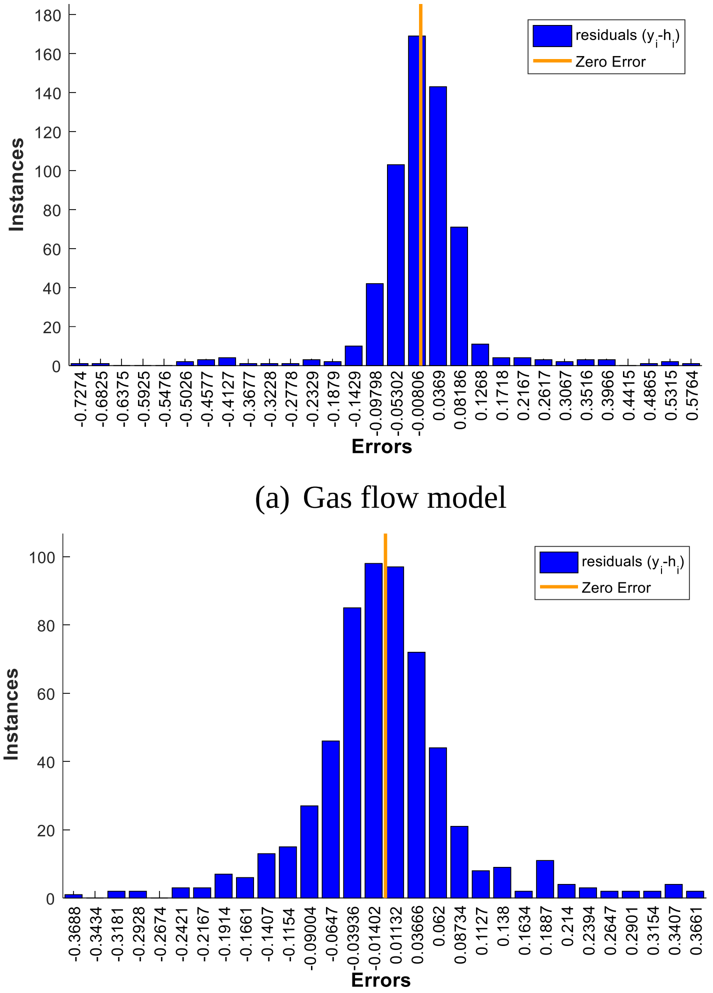


(a) Gas flow model

(b) Oil flow model

Figure 6. R2 of the combined 10 training and testing folds. (a) Gas flow model. (b) Oil flow model.

Testing folds are used to plot the histogram of prediction residuals and are presented in Figure 7. The residuals follow a Gaussian distribution with almost zero mean (-0.002 and -0.001 to be exact) and standard deviation less than 0.1.

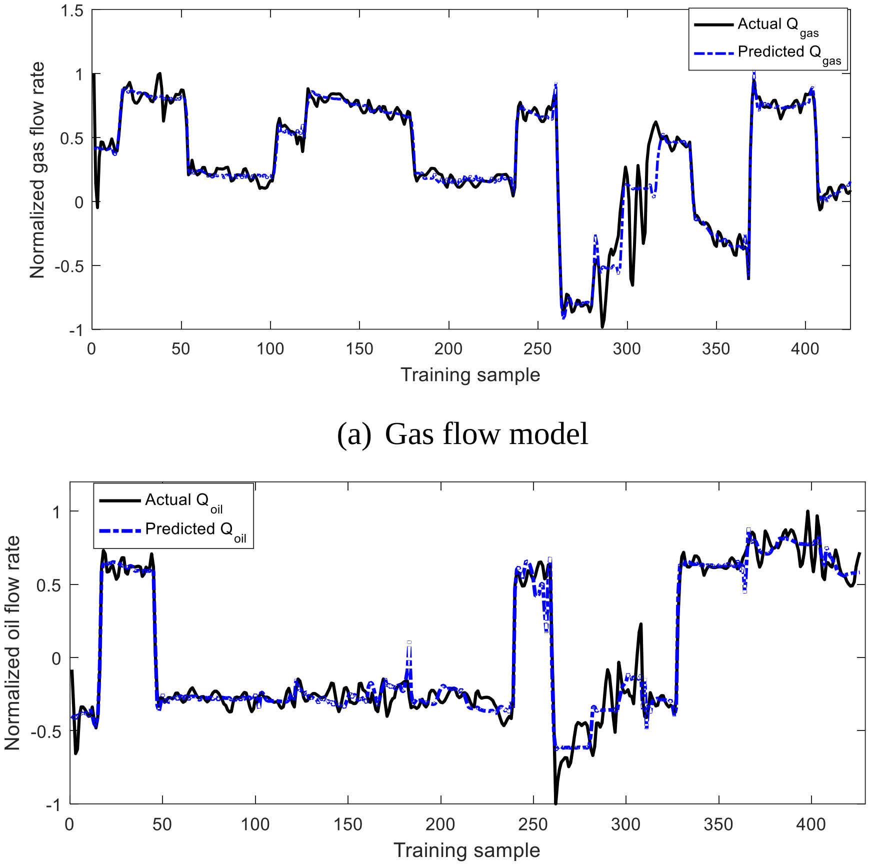


(a) Gas flow model

(b) Oil flow model

Figure 7. Histogram of average residuals across all folds. (a) Gas flow model. (b) Oil flow model.

Figure 8 illustrates the actual normalized flow rate trend of the first training fold compared to the neural network estimations. The plot shows good tracking ability without any signs of over-fitting.

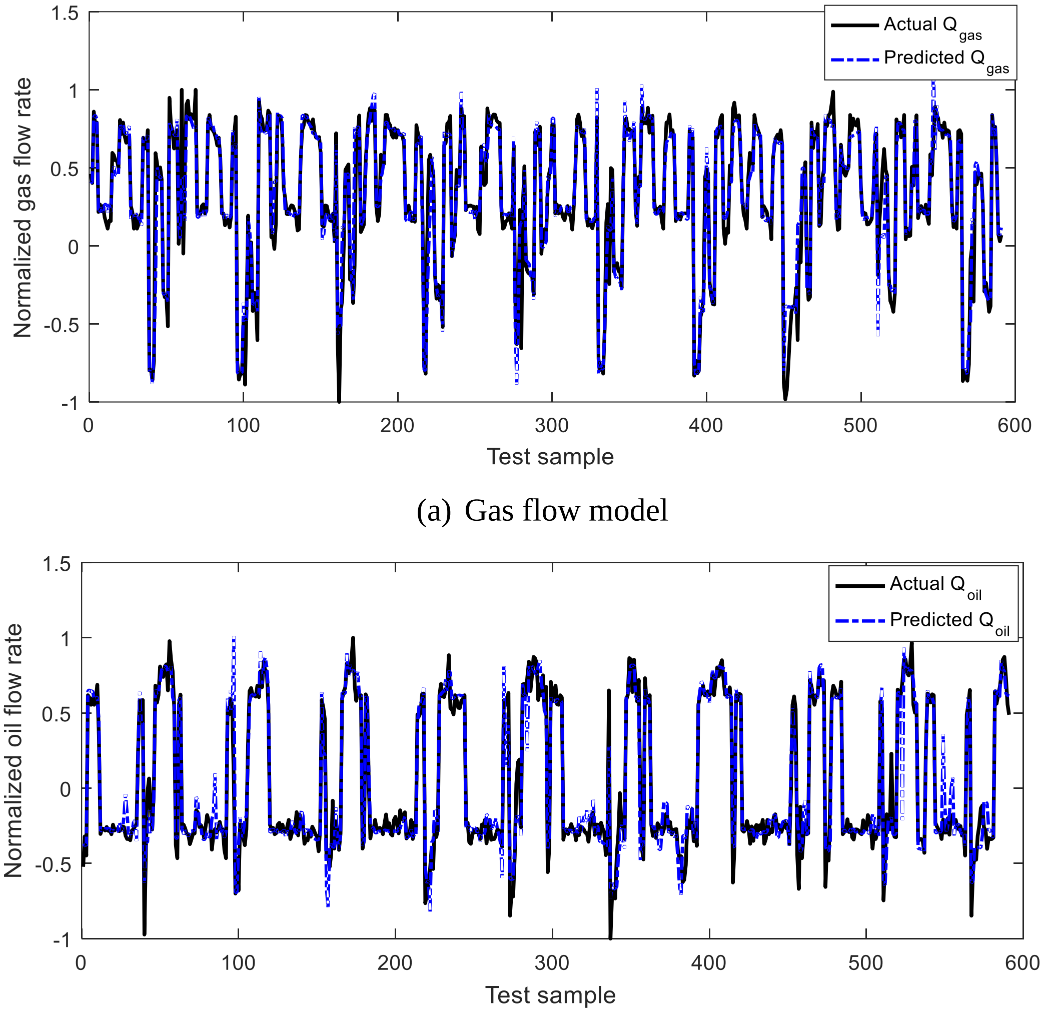


(a) Gas flow model

(b) Oil flow model

Figure 8. Actual versus predicted normalized flow rates for the first training fold. (a) Gas flow model. (b) Oil flow model.

Figure 9 captures the generalization ability of the developed models when presented with unseen data (testing sets). This shows model estimations follow closely with actual measurements. However, it is not enough to show the accuracy of the model. Next, soft sensor accuracy and cumulative flow performance are examined to confirm its accuracy and applicability.



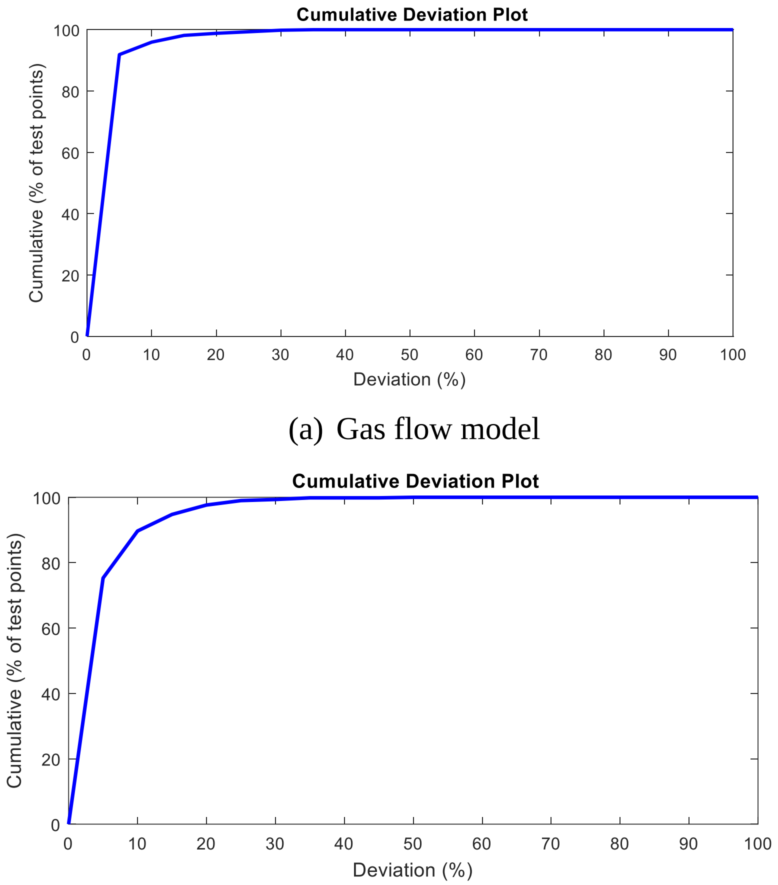
(a) Gas flow model

(b) Oil flow model

Figure 9. Actual versus predicted normalized flow rates of all testing folds. (a) Gas flow model. (b) Oil flow model.

d. Cumulative Deviation and Cumulative Flow Analysis

After validating the soft sensor model using the test data. It is imperative to also check soft sensor accuracy using cumulative deviation and cumulative flow plots. To plot cumulative performance, the relative error percent between actual and soft sensor estimate is calculated for each sample in all testing folds. Then, the number of samples with relative error below certain deviation percentage (5%, 10%, 15%, etc.) are counted. The counts are then divided by total number of samples then plotted against deviation percentage. Figure 10 shows the resulted cumulative deviation plots for gas and oil flow models.



(a) Gas flow model

(b) Oil flow model

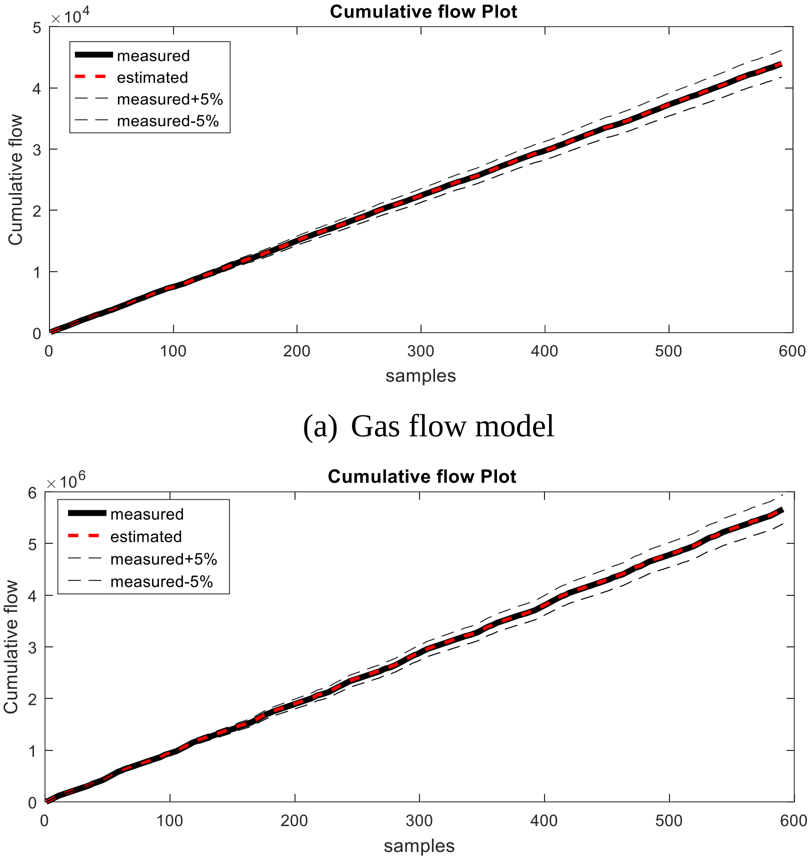
Figure 10. Cumulative deviation plot of soft sensor. (a) Gas flow model. (b) Oil flow model.

It can be seen from the plots that around 90% of points are within 10% deviation, while more than 97% of the points are within 20% deviation. Key deviation values are summarized in Table 3.

Table 3. Percent of test points at different deviation%

Model	**Deviation%**			
	5%	**10%**	**20%**	**30%**
Gas flow	91.9	95.9	98.8	99.4
Oil flow	75.3	89.7	97.6	99.8

From cumulative deviation results, the soft sensor is considered quite accurate for such application since physical well test measurements themselves have ±10% error, and it can be conveniently used for real-time monitoring, optimization and maintenance applications. To validate the soft sensor further, cumulative flow trends of un-normalized testing folds are presented in Figure 11.



(a) Gas flow model

(b) Oil flow model

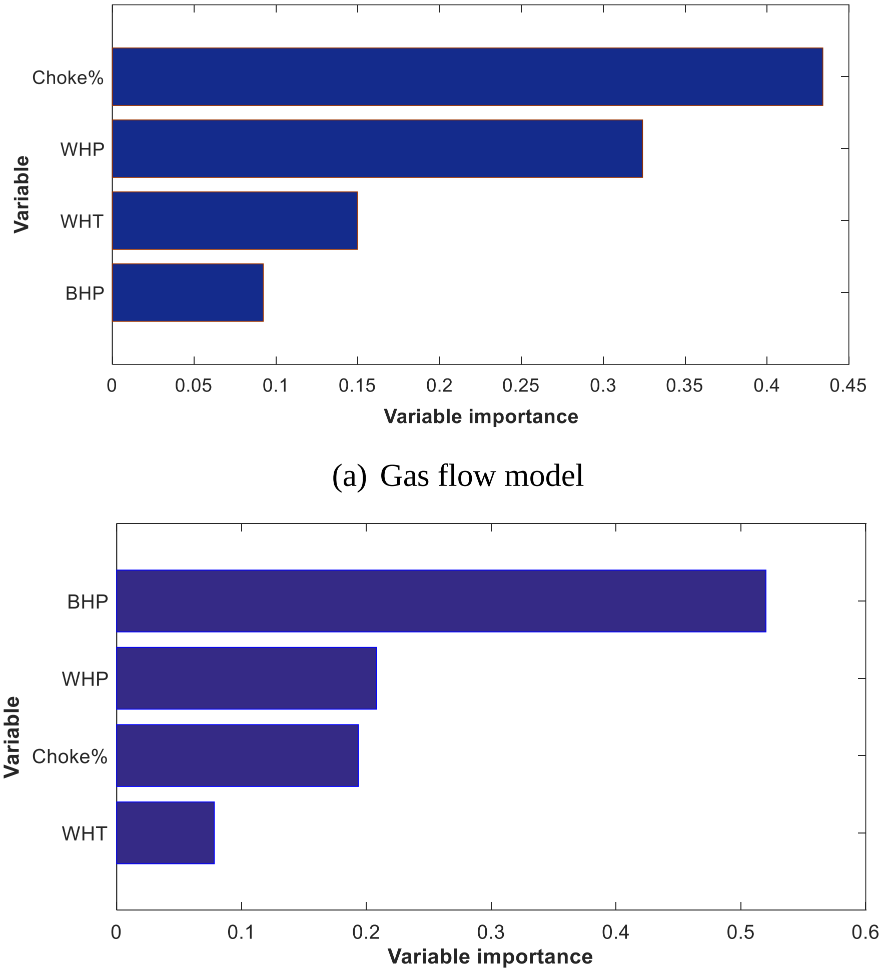
Figure 11. Cumulative flow plot of testing folds. (a) Gas flow model. (b) Oil flow model.

Cumulative flow of soft sensor predictions follows closely with actual cumulative flow without any apparent systematic drift. Deviation percent of final cumulated values, at last samples, are -0.07 % and -0.12% for gas and oil flow models respectively. This confirms that the model predictions are robust over time.

From reported results, it can be concluded that the developed soft sensor is accurate and can meet monitoring and optimization requirements. In Next section, the study of measurements importance is presented.

e. Measurements Importance

Not all oil and gas wells have permanent bottom-hole pressure (BHP) gauges installed. Furthermore, BHP frequently fails due to the harsh environment down the reservoir. Similarly, choke valve opening percentage (CV%) is not always available since some production wells are operated manually, hence no real-time update of CV% is possible. In this section, each measurement importance to soft sensor performance is examined. This is done by first analyzing the developed soft sensor sensitivity to a change in one of the inputs. The medians of inputs are fed to the soft sensor and outputs are recorded. Then 20% perturbation to one of the inputs is introduced and the change in output is recorded. After repeating this procedure for all inputs, the relative percent of change in output is plotted. The percentage signifies the input influence and importance. Figure 12 shows the contribution of each measurement to output change as percentage.



(a) Gas flow model

(b) Oil flow model

Figure 12. Input variables importance. (a) Gas flow model. (b) Oil flow model.

Secondly, a neural network is developed using a single measurement as its input, then its performance on testing folds is inspected. This is repeated for all input measurements. The performance summary is presented in Table 4.

Table 4. Model performance of individual input and of all inputs combined

Model	Indicator/Input	CV%	WHP	WHT	BHP	All
Gas flow rate	**R^2**	0.880	0.796	0.711	0.465	0.954
	MAPE	4.38	5.84	7.21	9.26	2.27
	MSE	4.65	7.56	10.19	16.19	1.83
	STD	1.58	2.75	3.10	2.96	0.74
	Min	2.22	4.00	4.56	12.76	2.91
	Max	6.57	12.38	16.09	21.58	0.70
Oil flow rate	**R^2**	0.857	0.861	0.500	0.899	0.965
	MAPE	8.10	7.76	16.89	6.75	4.22
	MSE	6.01	5.80	16.86	4.30	1.56
	STD	3.58	3.05	1.76	1.91	0.72
	Min	1.53	1.61	14.61	2.39	2.59
	Max	13.86	10.04	19.75	8.66	0.66

This summary confirms measurement importance in Figure 12 since the most significant measurement is able to perform well when it is used alone as input to the soft sensor. While the least significant measurements cannot predict flow rates accurately when used independently. Another observation is that some measurements are important to predict gas flow rate (CV%) but not as much when used to predict oil flow rate. The opposite is also true, BHP importance to predict oil flow rate. Finally, the most two significant measurements to obtain accurate predictions are BHP and CV% and this is in line with mechanistic multiphase flow modeling.

V. CONCLUSIONS

Feed-forward neural network soft sensor trained by Levenberg Marquardt algorithm and early stopping was developed to predict gas and oil phase flow rates in multiphase production wells. Neural network parameters were selected using K-fold cross-validation, and model was validated with actual well test data collected over a period of 1.5 years. The developed virtual flow meter displayed promising results with less than 10% deviation for 90% of the time. It is expected to meet real-time monitoring demands, to assist in production optimization and to reduce operational costs, as well as to backup and augment physical multiphase flow meters. For robust implementation of the proposed soft sensor, dynamic update of the neural network is recommended. This can be done using just-in-time learning techniques such as moving-window or any other adaptation techniques to accommodate any operational or reservoir changes over time.

VI. ACKNOWLEDGMENT

The authors acknowledge support provided by Universiti Teknologi Petronas.

REFERENCES

[1] L. Fortuna, S. Graziani, A. Rizzo, and M. G. Xibilia, Soft sensors for monitoring and control of industrial processes: Springer Science & Business Media, 2007, pp. 167-225.

[2] G. Wei and P. Tianhong, "An adaptive soft sensor based on multi-state partial least squares regression," in Control Conference (CCC), 2015 34th Chinese, 2015, pp. 1892-1896.

[3] M. Eastwood and P. Kadlec, "Interpretable, Online Soft-Sensors for Process Control," in 2011 IEEE 11th International Conference on Data Mining Workshops, 2011, pp. 581-587.

[4] D. R. Baughman and Y. A. Liu, Neural Networks in Bioprocessing and Chemical Engineering: Elsevier Science, 2014, pp. 228-433.

[5] J. Mohd Ali, M. A. Hussain, M. O. Tade, and J. Zhang, "Artificial Intelligence techniques applied as estimator in chemical process systems – A literature survey," Expert Systems with Applications, vol. 42, pp. 5915-5931, 8/15/ 2015.

[6] S. Ramasahayam and S. R. Chowdhury, "Non Invasive Estimation of Blood Urea Concentration using Near Infrared Spectroscopy," INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS, vol. 9, pp. 449-467, 2016.

[7] N. Mohamed Ramli, M. A. Hussain, B. Mohamed Jan, and B. Abdullah, "Composition Prediction of a Debutanizer Column using Equation Based Artificial Neural Network Model," Neurocomputing, vol. 131, pp. 59-76, 5/5/ 2014.

[8] H. Zargari, S. Poordad, and R. Kharrat, "Porosity and permeability prediction based on computational intelligences as artificial neural networks (ANNs) and adaptive neuro-fuzzy inference systems (ANFIS) in southern carbonate reservoir of Iran," Petroleum Science and Technology, vol. 31, pp. 1066-1077, 2013.

[9] E. Oko, M. Wang, and J. Zhang, "Neural network approach for predicting drum pressure and level in coal-fired subcritical power plant," Fuel, vol. 151, pp. 139-145, 7/1/ 2015.

[10] P. Kadlec, B. Gabrys, and S. Strandt, "Data-driven Soft Sensors in the process industry," Computers & Chemical Engineering, vol. 33, pp. 795-814, 4/21/ 2009.

[11] X. Yuan, Z. Ge, and Z. Song, "Locally Weighted Kernel Principal Component Regression Model for Soft Sensing of Nonlinear Time-Variant Processes," Industrial & Engineering Chemistry Research, vol. 53, pp. 13736-13749, 09/03 2014.

[12] H. Liu and C. Yoo, "A robust localized soft sensor for particulate matter modeling in Seoul metro systems," Journal of Hazardous Materials, vol. 305, pp. 209-218, 3/15/ 2016.

[13] S. Guptaa, S. B. Satpala, S. Banerjeeb, and A. Guhaa, "VIBRATION BASED HEALTH MONITORING OF HONEYCOMB CORE SANDWICH PANELS USING SUPPORT VECTOR MACHINE," International Journal on Smart Sensing & Intelligent Systems, vol. 9, pp. 233 – 255, 2016.

[14] D. Riordan, P. Doody, and J. Walsh, "The use of artificial neural networks in the estimation of the perception of sound by the human auditory system," International journal on smart sensing and intelligent systems, vol. 8, pp. 1806-1836, 2015.

[15] A. Thati, A. Biswas, S. R. Chowdhury, and T. K. Sau, "BREATH ACETONE-BASED NON-INVASIVE DETECTION OF BLOOD GLUCOSE LEVELS," International Journal on Smart Sensing & Intelligent Systems, vol. 8, pp. 1244-1260, 2015.

[16] F. A. A. Souza, R. Araújo, and J. Mendes, "Review of soft sensor methods for regression applications," Chemometrics and Intelligent Laboratory Systems, vol. 152, pp. 69-79, 3/15/ 2016.

[17] G. Falcone, G. Hewitt, and C. Alimonti, Multiphase flow metering: principles and applicationns vol. 54: Elsevier, pp. 20-31, pp. 191-224 and pp. 267-301, 2009.

[18] R. Thorn, G. Johansen, and B. Hjertaker, "Three-phase flow measurement in the petroleum industry," Measurement Science and Technology, vol. 24, pp. 1-17, 2012.

[19] R. Varyan, "Cost Saving of Implementing Virtual Flow Metering at Various Fields and Engineering Phases-A Case Study," in Offshore Technology Conference-Asia, Malaysia, March 22, 2016, doi:10.4043/26637-MS.

[20] R. API, "86-Recommended Practice for Measurement of Multiphase Flow (2005)," API Executive Committee on Drilling and Production Operations, American Petroleum Institute, pp. 26-47.

[21] M. J. Moghaddasi, I. Lotfi, and M. Moghaddasi, "Comparison of correlations for predicting oil flow rate passing through chokes," Energy Sources, Part A: Recovery, Utilization and Environmental Effects, vol. 37, pp. 1340-1345, 2015.

[22] P. Patel, H. Odden, B. Djoric, R. D. Garner, and H. K. Vea, "Model Based Multiphase Metering and Production Allocation," in Offshore Technology Conference-Asia, Malaysia, Mar 25, 2014, doi: 10.4043/25457-MS.

[23] A. Amin, "Evaluation of Commercially Available Virtual Flow Meters (VFMs)," in Offshore Technology Conference, USA, May 4, 2015, doi: 10.4043/25764-MS.

[24] O. Bello, S. Ade-Jacob, and K. Yuan, "Development of Hybrid Intelligent System for Virtual Flow Metering in Production Wells," in SPE Intelligent Energy Conference & Exhibition, The Netherlands, Apr 1, 2014, doi: 10.2118/167880-MS.

[25] R. Varyan, R. Haug, and D. Fonnes, "Investigation on the Suitability of Virtual Flow Metering System as an Alternative to the Conventional Physical Flow Meter," in SPE/IATMI Asia Pacific Oil & Gas Conference and Exhibition, Indonesia, 20 Oct, 2015, doi: 10.2118/176432-MS.

[26] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," Neural networks, vol. 2, pp. 359-366, 1989.

[27] S. S. Haykin, Neural networks and learning machines vol. 3: Pearson Education Upper Saddle River, pp. 1-46, 2009.

[28] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, Neural network design: Martin Hagan, pp. 363-431 and pp. 468-598, 2014.

[29] D. Hunter, H. Yu, I. M. S. Pukish, J. Kolbusz, and B. M. Wilamowski, "Selection of Proper Neural Network Sizes and Architectures; A Comparative Study," IEEE Transactions on Industrial Informatics, vol. 8, pp. 228-240, 2012.

[30] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," in Neural Networks, 1990., 1990 IJCNN International Joint Conference on, 1990, pp. 21-26.

[31] R. Battiti, "First-and second-order methods for learning: between steepest descent and Newton's method," Neural computation, vol. 4, pp. 141-166, 1992.

[32] H. Yu and B. M. Wilamowski, "Levenberg–marquardt training," Industrial Electronics Handbook, vol. 5, p. 1, 2011.

[33] R. Caruana, S. Lawrence, and L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in NIPS, 2000, pp. 402-408.

[34] S. Corneliussen, J.-P. Couput, E. Dahl, E. Dykesteen, K.-E. Frøysa, E. Malde, et al., "Handbook of multiphase flow metering," Norwegian Society for Oil and Gas Measurement (NFOGM), Revision, vol. 2, 2005.