SPE-SAS-321

Estimating Layers Deliverability in Multi-Layered Gas Reservoirs Using Artificial Intelligence

Alarfaj, M. K., Saudi Aramco; Abdulraheem, A., KFUPM; Al-Majed, A.; KFUPM; Hossain, E., KFUPM

This paper was prepared for presentation at the 2012 SPE Saudi Arabia Section Technical Symposium and Exhibition held in AlKhobar, Saudi Arabia, 8–11 April 2012.

Abstract

In this paper, an artificial intelligence (AI) model has been created to estimate the production rate of each layer in a multi-layered gas reservoir using static properties (such as those obtained from well logging)and dynamic properties (such as pressure). This helps in several reservoir engineering applications, such as understanding depletion in layers, or targeting specific layers for workover. It could also be used for PLT analysis where the measured PLT values are compared to the expected values and a variance analysis could be performed.

Data were collected from more than 100 wells in a certain reservoir spanning over four fields. They were clustered in related input variables and fed to the AI model for learning purposes. To compare the AI methods, the data were fed to 4 different methods (MLP, RBF, SVM, and GRNN) and the results were optimized for each method.

Between the tested AI methods, SVM and GRNN performed best with a low mean absolute error percentage and a very high correlation coefficient. This paper shows a high potential for AI methods in estimating production rate from each layer in a multi-layered gas reservoir.

Introduction

The ultimate goal of this paper is to create a model that can estimate the production ratio of each layer in a well to generate what could be considered as a “virtual PLT curve”. Our approach consists of the following steps:

- Gather relevant data from well logs, pvt tests, and well tests in a table that can be processed by the AI Application to create an AI model using these data.
- Include the results of flow equations (Darcy & Forchheimer) in the table to help in guiding the artificial model.
- Use several artificial intelligence methods to develop the model and recommend the method with the highest accuracy.

We have built artificial intelligence models using the aforementioned inputs with *r*ate as the output. Several artificial intelligence methods such as ANN, GRNN, RBF, and SVM are presented in this study.

We have used two different data sets. The first one includes the inputs in their original formats and the second one has normalized inputs. The results of each data set are compared for their performance and efficiency.

Literature Review

Well Deliverability

Well deliverability equations describe the relationship between the well production rate and the drawdown pressure, i.e. the difference between the reservoir pressure and the flowing bottomhole pressure. Presenting the production rate as a function of the drawadown pressure helps in comparing wells as well as in estimating the production rate under various conditions. This is also known as the “inflow performance relationship” or IPR.

In a single-layered gas reservoir, the gas well deliverability can be approximated using the pseudo-steady state relationship developed from Darcy’s law [Economides et al., 1994]:

$$\bar{p}^2 - p_{wf}^2 = \frac{1424 q \mu Z T}{kh}\left(\ln\left(0.472\frac{r_e}{r_w}\right) + s\right) \tag{1}$$

Which can be rearranged as:

$$q = \frac{kh}{1424 \mu Z T\left(\ln\left(0.472\frac{r_e}{r_w}\right)+s\right)}\left(\bar{p}^2 - p_{wf}^2\right) \tag{2}$$

The gas rate is in MSCF/d and the properties μ and Z are average properties between $\bar{p}$ and p_{wf}.

The above equation is commonly expressed as:

$$q = C(\bar{p}^2 - p_{wf}^2) \tag{3}$$

Where C is defined as:

$$C = \frac{kh}{1424 \mu Z T\left(\ln\left(0.472\frac{r_e}{r_w}\right)+s\right)} \tag{4}$$

Since this approximation assumes Darcy flow in the reservoir, this approximation is only acceptable for low gas flow rate. However for larger gas flow rates, where non-Darcy flow is dominant, we use the solution of the Forchheimer equation for gas flow through porous media and get:

$$q\left(\frac{MSCF}{d}\right) = \frac{kh\left(\bar{p}^2 - p_{wf}^2\right)}{1424 \bar{\mu}\, \bar{z}\, T\, [\ln(r_d/r_w) + s + Dq\,]} \tag{5}$$

This equation can be rearranged to come up with the following equation:

$$\bar{p}^2 - p_{wf}^2 = \frac{1424 \bar{\mu}\, \bar{z}\, T}{kh}\left(\ln\frac{0.472 r_e}{r_w} + s\right) q + \frac{1424 \bar{\mu}\, \bar{z}\, T D}{kh} q^2 \tag{6}$$

Or in different terms:

$$\bar{p}^2 - p_{wf}^2 = a\, q + b\, q^2 \tag{7}$$

Where

$$a = \frac{1424\bar{\mu}\,\bar{z}\,T}{kh}\left(\ln\frac{0.472r_e}{r_w} + s\right) \quad (8)$$

and

$$b = \frac{1424\bar{\mu}\,\bar{z}\,TD}{kh} \quad (9)$$

The Dq term in equation 5 refer to the turbulence skin effect which could be quite high for some high rate wells. Several authors proposed approximations for the non-Darcy coefficient (D). One is the following empirical correlation [Economides et al., 1994]:

$$D = \frac{6\times10^{-5}\gamma k_s^{-0.1}h}{\mu r_w h_{perf}^2} \quad (10)$$

To estimate the rate in a multi-layered reservoir, we use the principle of superposition [Juell et al., 2011]:

$$q = \sum_{l=1}^{N} q_l \quad (11)$$

Artificial Intelligence

Artificial Intelligence is defined as “the subfield of computer science concerned with the use of computers in tasks that are normally considered to require knowledge, perception, reasoning, learning, understanding and similar cognitive abilities” [Duda, 1981]. It uses soft computing techniques to provide better results than the conventional solutions. It includes, amongst many things, perceptrons, problem solving, language, conscious, and unconscious processes.

Artificial intelligence has become more and more popular in the last two decades in the petroleum industry. It has been extensively used and many SPE papers showed successful usage of artificial intelligence methods to solve petroleum engineering problems [Mohaghegh, 2005].

Artificial intelligence applications in the petroleum industry includes oil field optimization [Saputelli et al., 2002] lithofacies identification PVT properties estimation, production optimization, reserve estimation, history matching, MWD data analysis, drill bit diagnosis, hydraulic fracture analysis, bottomhole pressure prediction, well test analysis, critical gas flow rate prediction, and gas-lift optimization [Al-Dhufairi, 2011].

ANN

There are several types of Artificial Neural Networks (ANN). The most common ones are Multilayer Perception Networks (MLP), Probabilistic Neural Networks (PNN) & General Regression Neural Networks (GRNN), and Radial Basic Functions (RBF). We will briefly discuss each of these types.

MLP

It is the most common type of ANN, usually when the term ANN is used without qualification, it refers to MLP. An MLP Network usually consists of a single input layer, a single (or multiple) hidden layer, and a single output layer. Each layer consists of at least one neuron. For each predictor (input) variable, there is a neuron in the input layer. Similarly, for each target (output) variable there is a neuron in the output layer [Beale et al.].

For optimum results, the input variables should be normalized. The input layer will feed each input variable to all of the neurons in the next hidden layer. Moreover, the bias, which is a constant equals to 1, is also fed to all of the neuron in each hidden layer. After getting multiplied by a weight, the bias is added to the sum which is fed to the neuron.

Once received by the hidden layer, the values from the input layer get multiplied by a weight. Then all of the weighted received values get summed together and fed to the transfer function in the neuron. The outputs from each of the neurons in the hidden layer are fed to the next hidden layer or to the output if that was the last hidden layer.

The process gets repeated until the values are received by the output layer. Once received by the output layer, the values from each hidden layer neuron get once again multiplied by a another weight and summed together to be fed to be fed to a transfer value which will outputs the results of the network. If any normalization was done on the input layer, the output variables are transformed back to the same order of the input variables using the inverse of the normalization functions.

Most problems can be solved using one hidden layer. Only data with discontinuities require two hidden layers. Adding a layer won't usually improve the model, rather it may introduce the risk of converging to a local minimum. Theoretically, there is no reason to build a model with more than two hidden layers.

Figure 1 shows a basic diagram of a fully connected 4-layer feed forward perception neural network.The difference between the feed forward and the back propagation networks is that the feed forward networks the values can only move from the input to the hidden layer and from the hidden layer to the next one and so on with no values fed back to earlier layers, while in the back propagation networks allow the values to be fed backward.

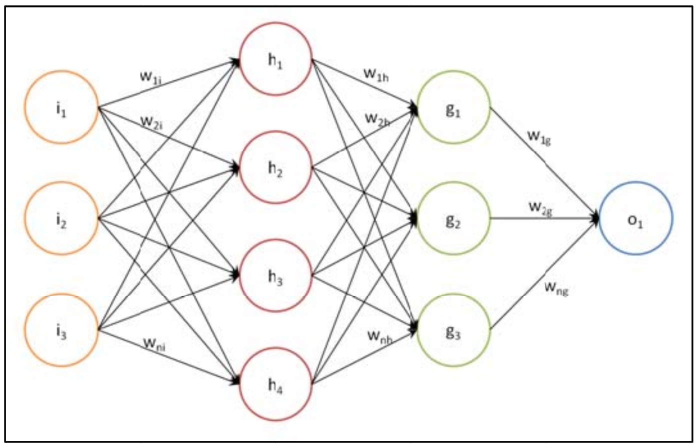

Figure 1: Basic Diagram of a Fully Connected 4-Layer Feed Forward Perception Neural Network

PNN/GRNN

The probabilistic and the general regression neural networks are very similar with one very important difference: PNN is designed for categorical outputs whereas GRNN is designed for continuous outputs. Training a PNN or a GRNN is usually much faster than training an MLP Network. They are also usually more accurate and relatively insensitive to outliers. However, PNN or GRNN models require much more memory than MLP for storing the model.

The basic idea of PNN/GRNN models is that the target variable is probably very close to the value of other variables that have very similar predictor (input) variables.

There are always 4 layers in PNN and GRNN. For each predictor (input) variable, there is one neuron. For PNN, there are N-1 neurons for N number of categories. The input is normalized by subtracting the median and dividing by the interquartile range. Subsequently the data is fed to all of the neurons in the hidden layer.

There is one neuron in the hidden layer for each case in the input data. The neuron in the hidden layer stores both the values of input variable for that specific case and the target value. The hidden neuron calculates the Euclidean distance between the center of the neuron and the test case. After that, using the sigma variables it applies the RBF kernel function using the sigma values. Then the data is fed to the neurons in the pattern/summation layer.

For PNN network the third layer contains one pattern neuron for each of the categories of the target (output) layer. The weighted value that came from the hidden layer is fed only to the pattern neuron that is related to the category of the hidden neuron. The values of each class are added by the representative neuron.

On the other hand, the GRNN network has only 2 neurons in the summation layer. They are called the denominator summation unit and the numerator summation unit. The weight values coming from each of the hidden layers are summed in the denominator summation unit while the numerator summation unit sums the weights multiplied by the actual target (output) value of each hidden neuron.

The fourth and final layer is the decision layer. For PNN networks, the decision layer look at the values of the weighted votes for each target category in the pattern layer and uses the largest accumulation to predict the target category (the output). For the GRNN networks, in the division layer, the value which accumulated in the numerator summation unit is divided by the value accumulated in the denominator summation unit and the result is used to predict the target value [Sherrod, 2008].

RBF

Radial Basis Function Neural Networks (RBF) are very similar to GRNN networks with one main difference which is that while GRNN have one neuron for each input data point, RBF have a number of neurons which is most of the time much less than the number of input data points. It is recommended to use GRNN for small to medium-sized data sets, since they will deliver more accurate results than RBF. However, RBF is more suited for large and very large data sets since GRNN is almost not practical for these kinds of data sets.

The basic idea of RBF models is, like PNN/GRNN, that the target variable is probably very close to the value of other variables that have very similar predictor (input) variables. What RBF networks do is that they place at least one RBF neuron in the space which is described by the input variables. The dimensions of this space have the same number as the predictor variables. The neuron being evaluated calculates the Euclidean distance between the center of the neuron and the center of each neuron in that space. An RBF function (usually Gaussian) is applied to each distance to estimate the weight based on the influence of each neuron. The further away a neuron is, the less influence it has on target neuron. The predicted value is best estimated by multiplying the weight of the connection by the output value of the RBF function.

RBF networks have three layers, input, a hidden layer which contains an RBF function (like Gaussian function), and an output layer. The input is normalized by subtracting the median and dividing by the interquartile range. Subsequently the data is fed to all of the neurons in the hidden layer.

The number of neurons in the hidden layer is variable and is determined by the training process. The neuron in the hidden layer stores both the values of input variable for that specific case and the target value. The hidden neuron calculates the Euclidean distance between the center of the neuron and the test case. After that, using the sigma variables it applies the RBF kernel function using the sigma values. Then the data is fed to the neurons in the summation layer.

The last layer is the summation layer. It gets the output of the hidden layer multiplied by a specific weight that is specific for that neuron. It then sums all of the incoming values to get the output [Sherrod, 2008].

SVM

Support vector machine model classify data by creating an N-dimensional hyperplane which separates the data optimally. In SVM, the predictor variable is called an attribute. When it is transformed to define the hyperplane, it is called a feature. The set of features that describes one case of predictor values is called a vector.

Ultimately, the goal of support vector machine is to find an optimal hyperplane where one category of the target variables is on one side, and another category is on the other. SVM uses kernel functions such as linear, polynomial, sigmoid, and radial based functions [Sherrod, 2008].

Data Inputs

It is generally a good idea to include physical relationships or correlations in the input of the artificial intelligence model. It helps in guiding the model in the training phase. After we look at the solution of Darcy's equation for flow in porous media:

$$q = C(\bar{p}^2 - p_{wf}^2) \tag{3}$$

and the solution of Forchheimer equation as well:

$$\bar{p}^2 - p_{wf}^2 = a\,q + b\,q^2 \tag{7}$$

We observed that the gas rate is directly related to the difference of the squared pressures, so we defined as an input:

$$d(p^2) = \bar{p}^2 - p_{wf}^2 \tag{12}$$

We have also decided to use the constants used in these equations (C, a, and b) as inputs:

$$C = \frac{kh}{1424\mu ZT\left(\ln\left(0.472\frac{r_e}{r_w}\right)+s\right)} \tag{4}$$

$$a = \frac{1424\bar{\mu}\,\bar{z}\,T}{kh}\left(\ln\frac{0.472r_e}{r_w} + s\right) \tag{8}$$

$$b = \frac{1424\bar{\mu}\,\bar{z}\,TD}{kh} \tag{9}$$

Moreover, to account for the effects of turbulent flow, we included the non-Darcy flow coefficient:

$$D = \frac{6\times10^{-5}\gamma k_s^{-0.1}h}{\mu r_w h_{perf}^2} \tag{10}$$

And finally since the permeability is a key difference between the wells in our study, we decided to include it as an input as well.

So, we ended up with six inputs [a, b, C, D, $d(p^2)$, and permeability] and one output [q_{layer}].

We have also performed statistical analysis (Tables 1-3) on the input data to eliminate any anomalies and to learn the limits of the artificial intelligence model.

Property	**Reservoir Pressure psi**	**Perforation Thickness ft**	**Flowing Temperature °f**	**Flowing Pressure psi**
Minimum	2298	2.60	238.40	1665
Maximum	6960	61.70	292.03	6630
Mean	5334	14.79	256.47	4650
Median	5340	10.90	254.43	4708
Standard Deviation	1036	10.03	9.14	1154
Coefficient of Correlation	0.0040	0.2523	0.0500	-0.0296

Table 1: Statistical Analysis of the Reservoir Pressure, Perforation Thickness, Flowing Temperature, and Flowing Pressure

Property	Diameter Ft	Flow Zone Thickness ft	Water Saturation Fraction	Porosity Fraction
Minimum	3.650	5.25	0.0454	0.0374
Maximum	9.020	66.50	0.4752	0.2303
Mean	5.361	18.37	0.2452	0.1067
Median	6.004	14.00	0.2321	0.1072
Standard Deviation	1.029	11.96	0.0809	0.0302
Coefficient of Correlation	0.1116	0.2484	-0.1265	0.2478

Table 2: Statistical Analysis of the Diameter, Flow Zone Thickness, Water Saturation, and Porosity

Property	Permeability Md	Gas Specific Gravity Ratio	Skin	Gas Flow Rate MMSCF
Minimum	0.1035	0.6333	-5.90	104
Maximum	99.69	0.6724	-1.59	25385
Mean	12.78	0.6480	-4.72	3349
Median	6.86	0.6490	-4.70	2023
Standard Deviation	15.90	0.0115	0.69	3700
Coefficient of Correlation	0.6323	0.0171	0.0062	1.0000

Table 3: Statistical Analysis of the Permeability, Gas Specific Gravity, Skin, and Gas Flow Rate

Results

Hundreds of runs were performed using different artificial intelligence methods. In all runs we used 70:30 training to testing and validation ratio. Between the tested AI methods, SVM and GRNN performed best with a low mean absolute error percentage and a very high correlation coefficient. This paper shows promising use for AI methods in estimating production rate from each layer in a multi-layered gas reservoir.

Overall, the normalized set has shown better performance (major improvement in the cases of MLP and SVM). Figure 2 shows the mean absolute error of the model in MSCF. It shows that the normalized SVM model has peroformed best with a mean absolute error of 77 MSCF. Figure 3 shows the mean absolute percentage error of the different AI methods used in this study. Once again, the normalized SVM model has performed best with a mean absolute percentage error of 2.25%.

Figure 4 shows that all the models had a high coefficient of correlation. However, the highest were the normalized GRNN and SVM sets.

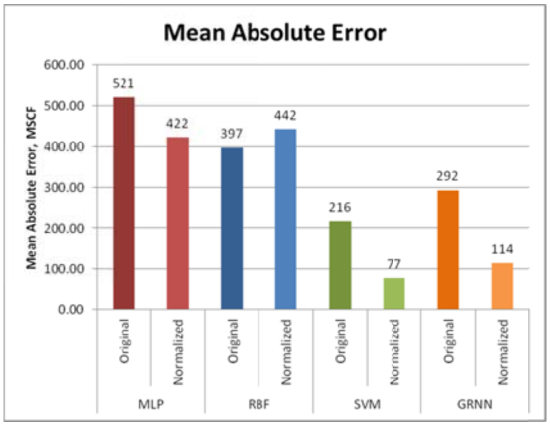

Figure 2: Mean Absolute Error of the different AI methods

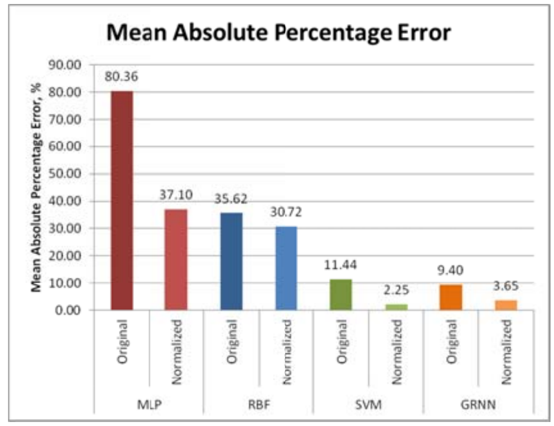

Figure 3: Mean Absolute Percentage Error of the different AI methods

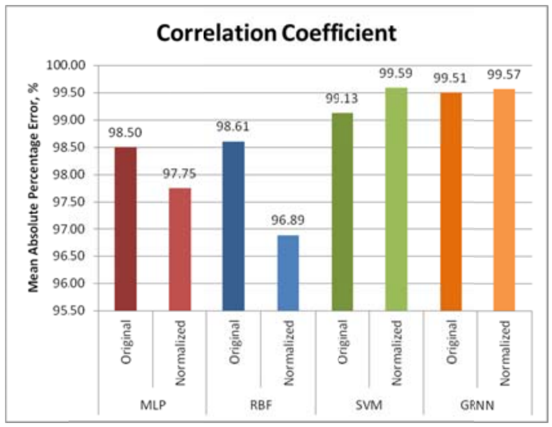

Figure 4: Correlation Coefficient of the different AI methods

Figures 5 and 6 show the crossplot between the measured rate and the rate estimated by the SVM model. They show a good linear correlation between the two. Figure 6 shows that the normalized set has less dispersion than the non-normalized set. Similar observation is noted with the other AI methods, which indicates the benefits of normalizing the data prior to training the model.

Figures 7 and 8 show the model error for both the non-normalized and the normalized sets respectively. We could observe that the non-normalized model results are more dispersed around the x-axis which indicates departure from the desired value. The figures also show a higher error in the model as the rate increases. However, as indicated by figures 9 and 10, the percentage error decreases as the rate increases signifying that the model accuracy increases as the rate increases. Figures 9 and 10 also show less dispersion in the normalized data set results.

Similar observations can be made for the GRNN model (Figures 11-16). Moreover, figures 13-16 show that the GRNN model tends to underestimate the layer production rate.

Overall, it is clear that normalized data sets performed better than the non-normalized data set. Furthermore, the Support Vector Machine model has performed better than the General Regression Neural Networks model. It should be noted that the accuracy of both models are limited to the range of the training data. If need to use the model outside the training range arises, the model should be re-trained with new data points that widen the range.

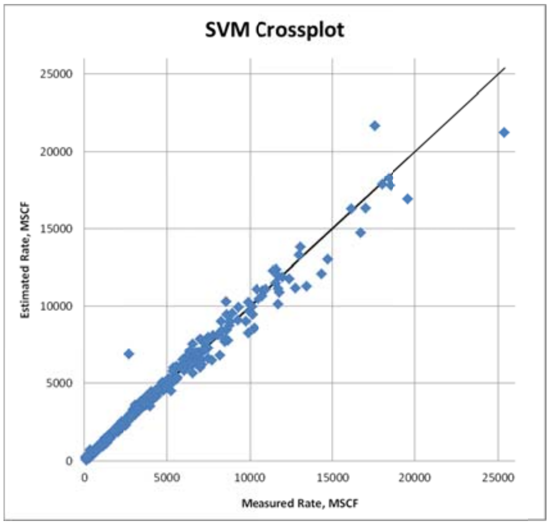

Figure 5: Crossplot of the measured and estimated rates using the SVM AI Method and the Original Data Set

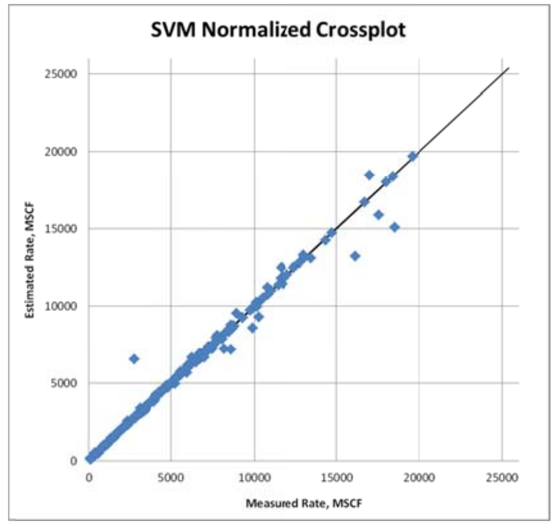

Figure 6: Crossplot of the measured and estimated rates using the SVM AI Method and the Normalized Data Set

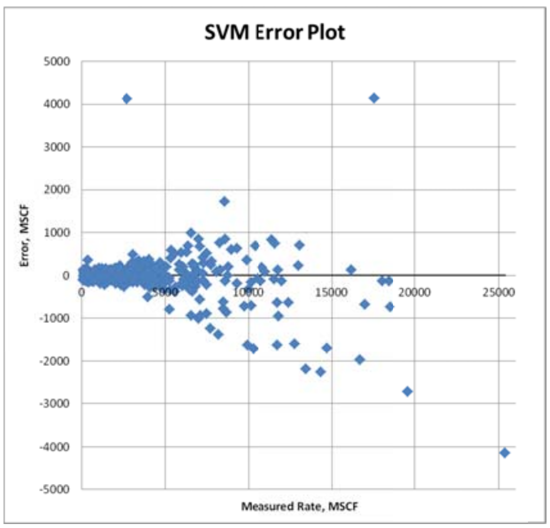

Figure 7: The Estimation Error using the SVM AI Method and the Original Data Set

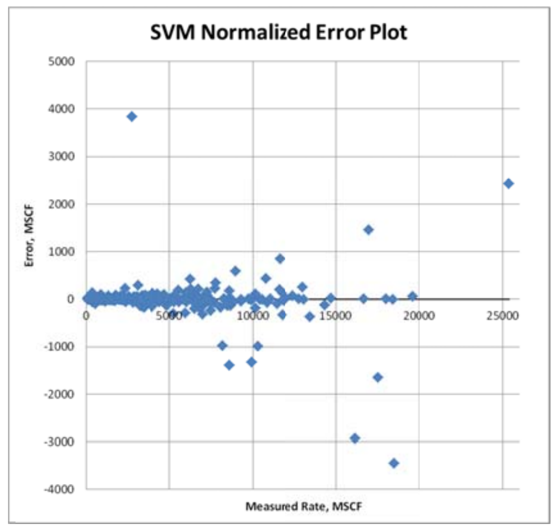

Figure 8: The Estimation Error using the SVM AI Method and the Normalized Data Set

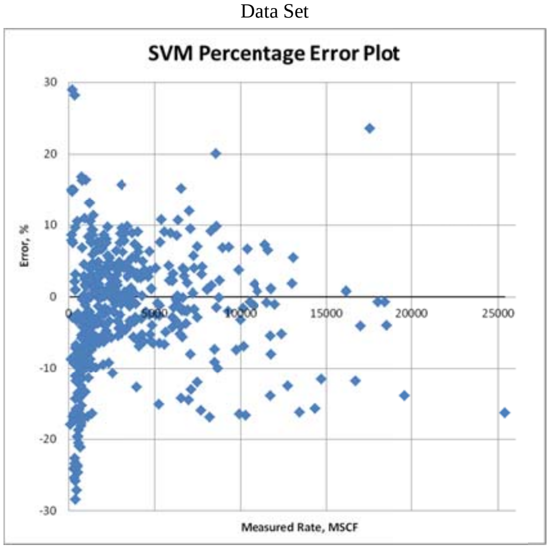

Figure 9: The Estimation Percentage Error using the SVM AI Method and the Original Data Set

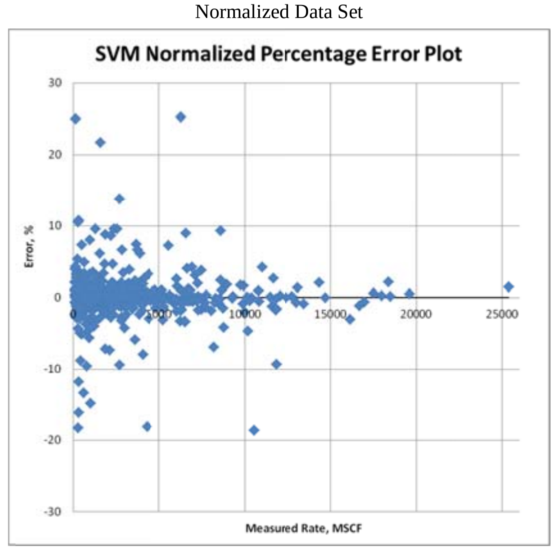

Figure 10: The Estimation Percentage Error using the SVM AI Method and the Normalized Data Set

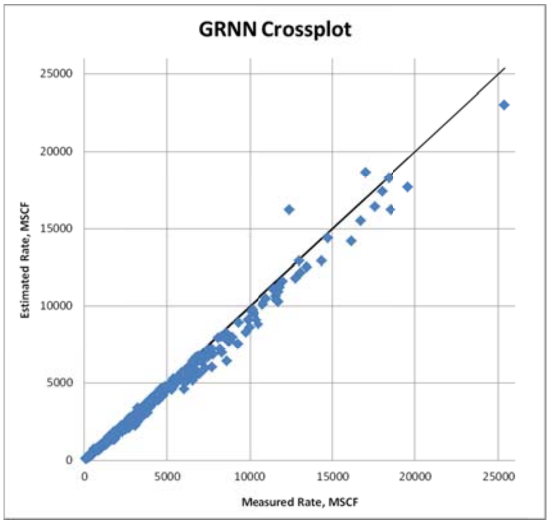

Figure 11: Crossplot of the measured and estimated rates using the GRNN AI Method and the Original Data Set

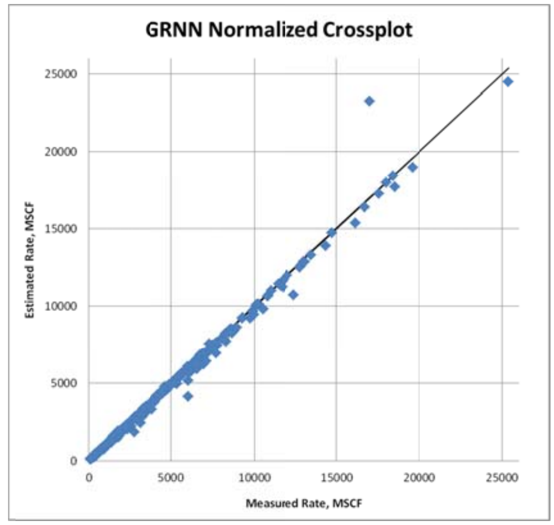

Figure 12: Crossplot of the measured and estimated rates using the GRNN AI Method and the Normalized Data Set

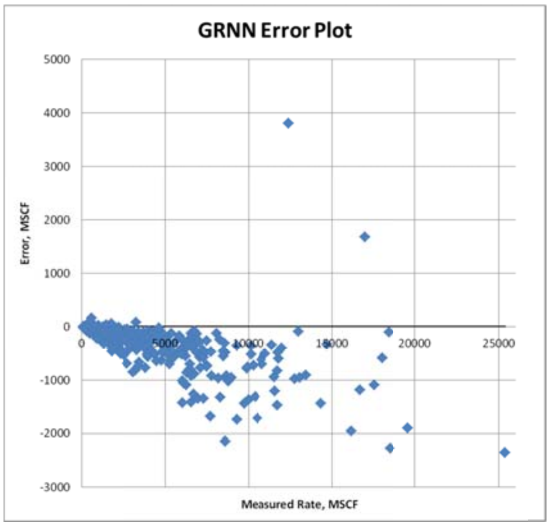

Figure 13: The Estimation Error using the GRNN AI Method and the Original Data Set

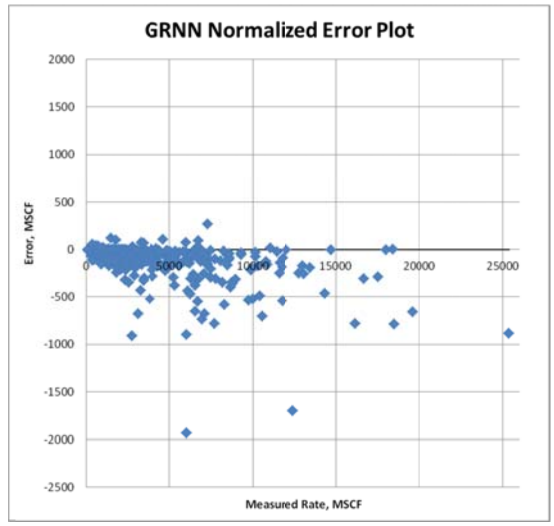

Figure 14: The Estimation Error using the GRNN AI Method and the Normalized Data Set

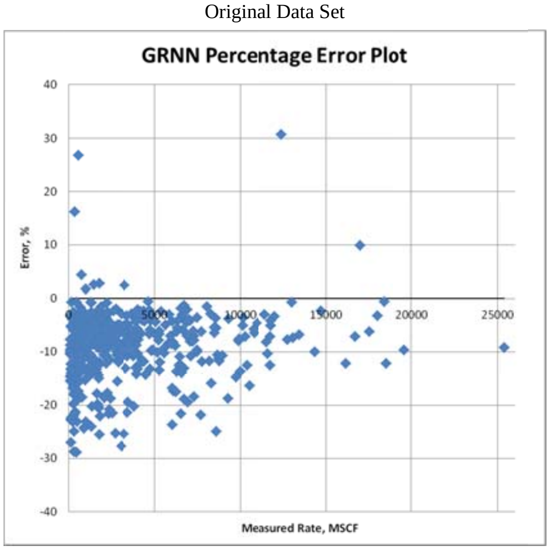

Figure 15: The Estimation Percentage Error using the GRNN AI Method and the Original Data Set

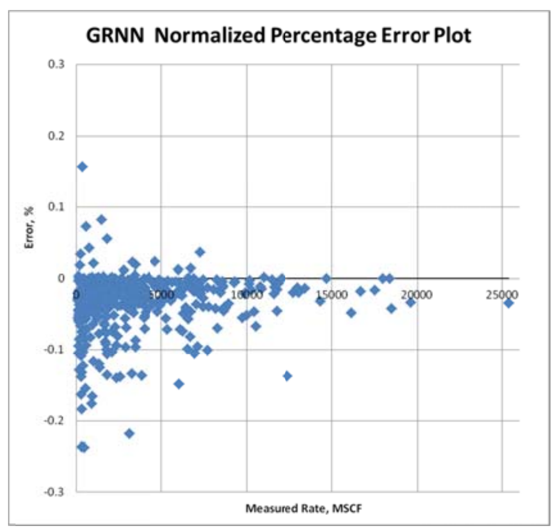

Figure 16: The Estimation Percentage Error using the GRNN AI Method and the Normalized Data Set

Conclusions and Reccommendations

- GRNN and SVM methods show promising results for estimating production rate from each layer in a multi-layered gas reservoir.
- Normalizing the input data sets has led to the improvement in the results, drastically in some cases.
- The accuracy of the model can be improved by quality checking and adding new data samples that covers wider ranges and different combinations.
- The developed model shouldn't be used outside the range of the training data. If need arises, it should be re-trained with new data points that widen the range.

References:

1. Economides, M., Hill, A. D., & Ehlig-Economides, C. (1994). Petroleum Production Systems. Cliffs, NJ: PTR Prentice Hall.
2. Juell, A., & Whitson, C. (2011). Backpressure Equation for Layered Gas Reservoirs. SPE 146066.
3. Duda, R. (1981). Knowledge-Based Expert Systems Come of Age. Byte, 238-281.
4. Mohaghegh, S. (2005). Recent Development in Application of Artificial Intelligence in Petroleum Engineering. JPT, 86-91.
5. Saputelli, L., Malki, H., Canelon, J., & Nikolaou, M. (2002). A Critical Overview of Artificial Neural Network Application in the Context of Continuous Oil Field Optimization. SPE 77703.
6. Al-Dhufairi, M. A. (2011). Prediction of Mobility Profile with Minimum Real Time Measurements using Artificial Intelligence. Dhahran, Saudi Arabia: KFUPM.
7. Beale, M. H., Hagan, M. T., & Demuth, H. B. (2010). Neural Network Toolbox 7 User's Guide. Natick, MA: The MathWorks, Inc.
8. Sherrod, P. H. (2008). DTREG Predictive Modeling Software User Manual. www.dtreg.com.