Neural Network Modelling of High Pressure CO_2 Corrosion in Pipeline Steels

Muhammad Hashim Abbas [a], Rosemary Norman [a], Alasdair Charles [b]

[a] Marine, Offshore and Technology Group, School of Engineering, Armstrong Building, Newcastle University, Newcastle upon Tyne, NE1 7RU, United Kingdom

[b] Bedson Building, Newcastle University, Newcastle upon Tyne, NE1 7RU, United Kingdom

ABSTRACT

The effect of carbon dioxide (CO_2) corrosion on pipelines is of great relevance to the petroleum as well as the Carbon Capture and Storage (CCS) industries. CO_2 corrosion is responsible for lost production as it brings about the gradual degradation of pipe internals with time. The cost of general corrosion is said to be between 3to 5% of an industrialised nation's gross domestic product (Schmitt et al., 2009; Popoola et al., 2013). In the U.S., the cost of corrosion in the production and manufacturing sector was $34.4 billion in 2014, with the oil and gas industry accounting for more than half (Abbas, 2016).

The use of neural networks (NN) as an analytic tool for corrosion data has been established however the aim of this paper is to characterise selected Matlab transfer and training functions, and assess their degree of suitability for CO_2 corrosion rate prediction. Assessments of the training functions include the evaluation of the correlation coefficient (R^2-value) and determination of a cumulative absolute error to indicate the level of precision and the extent of model accuracy. A NN model is developed for predicting CO_2 corrosion at high partial pressures by considering the results of the various tests and analyses on the given Matlab functions. The results showed that the model is reliable with all test results falling within the 95% confidence limits. Leave-One-Out Cross-Validation (LOOCV) was implemented as a means for carrying out an additional assessment on model performance as well as for model selection from possible alternatives.

Key Words: carbon dioxide (CO_2) corrosion, matlab, neural network, prediction model, high CO_2 partial pressure

1. Introduction

Evidence from various analysed meteorological data suggests that the composition of the earth's atmosphere is changing and this is due to elevated concentrations of greenhouse gases such as carbon dioxide and methane, induced mostly by the anthropogenic combustion of carbonaceous fossil fuels (Tans, 2009). CO_2 is the most significant greenhouse gas given that its annual emissions have risen by almost 80% between 1970 and the 2000s (Rao and Rubin, 2002; IPCC, 2014; IPCC, 2007). As of 2010, it represented approximately 77% of total global emissions, rising up to the 400ppm benchmark and surpassing this value in May 2013 (IPCC, 2014; Dlugokencky et. al., 2014).

The global percentage share of CO_2 emissions by sector indicates that electricity (power) and heat generation is the chief emitter (~40%). Also, power plants are heavily reliant on coal hence carbon capture and storage (CCS) technologies will be most effectively deployed in this sector with the potential to significantly reduce CO_2 emissions by serving as a low-cost option (IEA, 2010; IPCC, 2014; World Energy Council, 2007).

In CCS deployment, CO_2 is captured at its source and transported via pipelines at elevated pressures in the dense phase region, as seen in the CO_2 phase diagram of Fig. 1. As a result of the very high pressures, there is a risk of pipeline corrosion (IEA GHG, 2010). This risk thus highlights the need for reliable corrosion behaviour prediction in order to control corrosivity (Cottis et al., 1999). Neural networks (NN) have been utilised for the analysis of a variety of corrosion data (Cottis et al., 1999; Owen et. al., 2000). NNs use machine-learning algorithms to carry out non-parametric nonlinear regression of modelling data and as such offer more benefits in prediction than conventional polynomial and nonlinear regression techniques such as the ability to readily adapt to unknown functional forms (Owen et. al., 2000; Beale et. al., 2014). It also serves as a better method for knowledge acquisition (Radonja and Stankovic, 2002; Radonja, 2001). Another advantage of neural networks is that they are particularly useful prediction tools even in scenarios where modelling data is sparse as is currently the case with high pressure CO_2 corrosion (Radonja and Stankovic, 2002; Choi and Nesic, 2009; Choi and Nesic, 2011; Choi et. al., 2010).

In this paper, input data for the NN model was obtained from CO_2 corrosion studies of various sources in open literature (Choi and Nesic, 2009; Hesjevik et. al., 2003; Zhang et. al., 2012; Cui et. al., 2006). This input data was gathered and divided into separate training and testing datasets for the purpose of model development. The log-sigmoid (logsig), hyperbolic tangent sigmoid (tansig) transfer functions and a set of six training functions

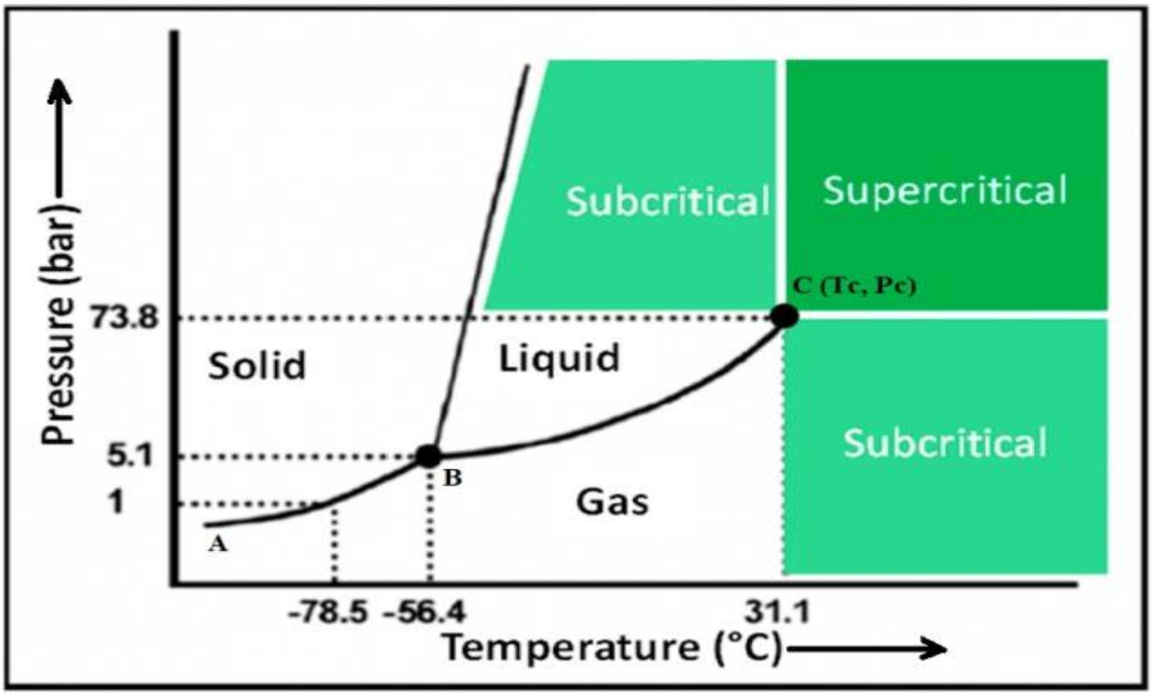

Fig. 1. *Pressure-Temperature phase diagram for pure CO_2 (Laboureur et al., 2015)*

were used in a series of tests (Beale et. al., 2014; Vogl et. al., 1988). The correlation coefficient and the sum of absolute error were calculated for each run to determine model accuracy (Draper and Smith, 1998; Sharma and Venugopalan, 2014). These results were then used as a basis for running initial tests to show general trends of performance of the given training functions and as a means for fine-tuning (pruning) the model parameters such as neuronal configurations best suited for high pressure CO_2 corrosion rate prediction.

2. Modelling

2.1 Training and testing data for NN Modelling

Data from multiple sources in open literature was used in the neural network modelling of high pressure CO_2 corrosion (see Appendix tables A1 and A2 for raw data). All data sources determined corrosion rates experimentally by weight loss using autoclaves. For the (Hesjevik et. al., 2003) study, a Hastelloy C-276 (UNS N10276) nickel-alloy was used and for the (Choi and Nesic, 2009) study, an X65 carbon steel sample was used.

For the (Zhang et. al., 2012) study, several samples of steel were used, including a martensitic carbon steel, a pipeline X65steel as well as three chromium-containing corrosion-resistant alloys (CRA).

For modelling purposes, only carbon steel corrosion rate results were used in order to maintain consistency as corrosion rate measurements for CRA would affect the derived model. For the (Cui et. al., 2006) study, samples of P110, N80 and J55 carbon steels were used.

In total, there are 22 data-points and these were divided into training and testing sets with 16 and 6 data-points for each set respectively. The bar chart of Fig. 2 shows the distribution of the given data-points from each source.

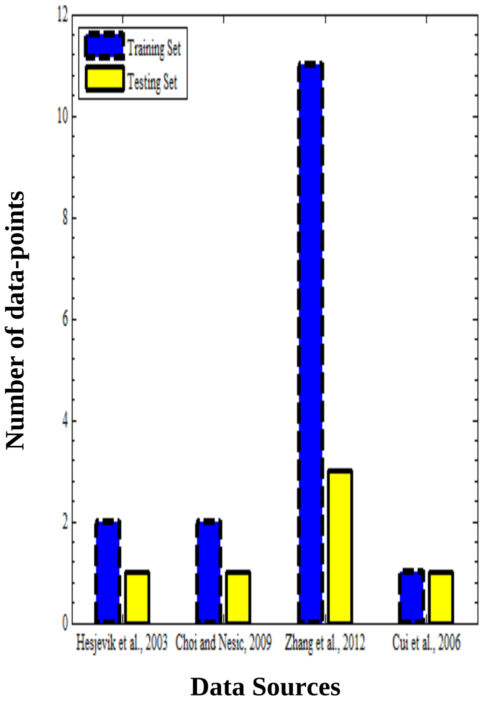

Fig. 2. *Bar chart showing the distribution of data-points from each source*

Overall the number of data-points from Zhang et. al. (Zhang et. al., 2012) exceeds those of the other sources (63% share) due to the fact that the experimental corrosion rate results for this source were carried out for the widest range of temperatures (50-130°C) and pressures (9.5-23.3MPa).

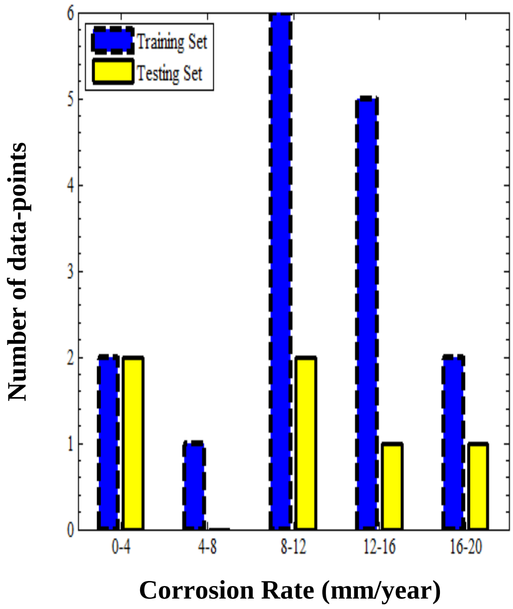

Fig. 3. *Bar chart showing the distribution of data-points for the grouped magnitudes of Corrosion Rate*

For the other sources, corrosion rate tests were carried out by maintaining a constant temperature whilst varying pressures or maintaining a constant pressure while varying temperatures as is the case with (Choi and Nesic, 2009) and (Cui et. al., 2006) respectively. For the study by (Hesjevik et. al., 2003), tests were focused on measurement of corrosion rates for temperatures less than 30°C. Experimental corrosion rate measurements were grouped into classes of 0-4, 4-8,..., 16-20mm/year. Figure 3 shows the distribution of corrosion rates in these classes.

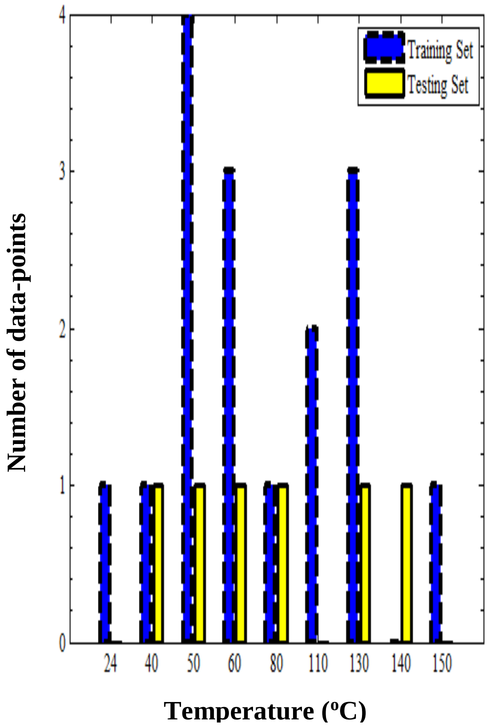

Fig. 4. *Bar chart showing the distribution of data-points for the recorded Temperatures*

From Fig. 3, the number of data-points for the mid-corrosion rate magnitude (8-12mm/year) is greater than those for end-point corrosion rate groups (0-4 and 16-20mm/year). The bar chart in Fig. 4 shows the distribution of data-points for the recorded experimental temperatures. There are more data-points in the mid-temperature readings (50 and 60°C) than for end-point temperature readings (24 and 150°C), as the former relates more closely to likely operating temperatures of pipelines than the latter. Figure 5 shows the corrosion rate-temperature profile for the training dataset. The median corrosion rate value was plotted for data-points with identical temperatures (see Appendix, Table A1). A polynomial curve fit through the points depicts the classic peak observed for CO_2 corrosion rate as a function of temperature (De Waard and Lotz, 1993). It is noted that the range of corrosion rates in the d-temperatures (50-80°C) is ~11mm/year, highlighting that the greatest variation in the magnitudes of corrosion occurs in these temperatures.

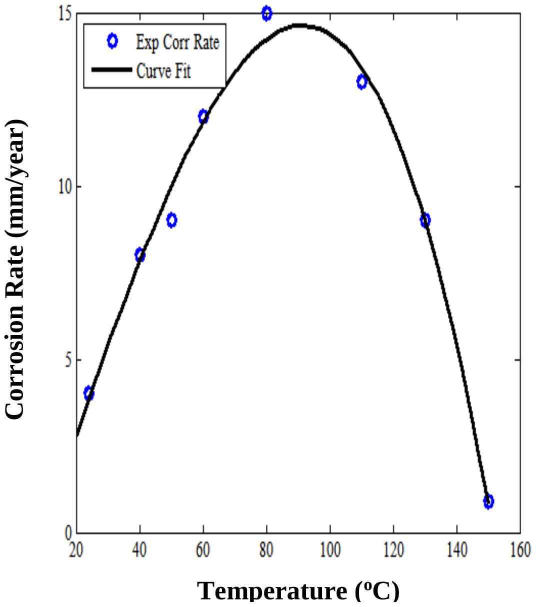

Fig. ***5.*** *Corrosion Rate against Temperature plot for Training dataset*

2.2 Effect of other variables on Corrosion Rate

In the previous section, the corrosion rate-temperature profile was discussed. In this section, the effects of all variables on CO_2 corrosivity will be discussed in terms of the statistical analyses carried out on the dataset.

Principal component analysis (PCA) was carried out on the dataset and is a mathematical procedure which transforms potentially correlated data into an orthogonal system of linearly uncorrelated principal components (Suryanarayana and Mistry, 2016). A weighted PCA is carried out in such a way that the first 2 or 3 principal components account for much of the variability within the dataset. (Jackson, 1991; Jolliffe, 2002), as shown in Fig. 6.

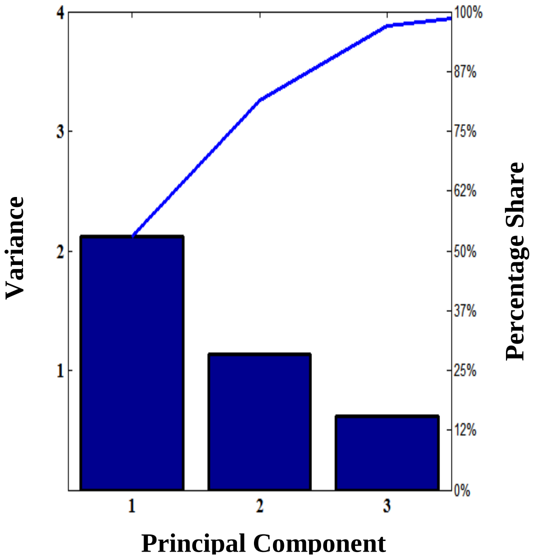

Fig. 6. *Pareto Chart showing the spread of the data (variance) among each of the principal components*

The cumulative percentages indicate how well the principal components explain variation within the data. The cumulative share for principal components 1 and 2 is 81.5% which is sufficient to explain variation in the dataset.

In addition to performing PCA on the data, it is also necessary to determine the extent of the influence of each variable to CO_2 corrosion rate. This is summarised graphically in the biplot shown in Fig. 7.

partial pressures can induce high corrosion rates in pipelines (Zhang et. al., 2012).

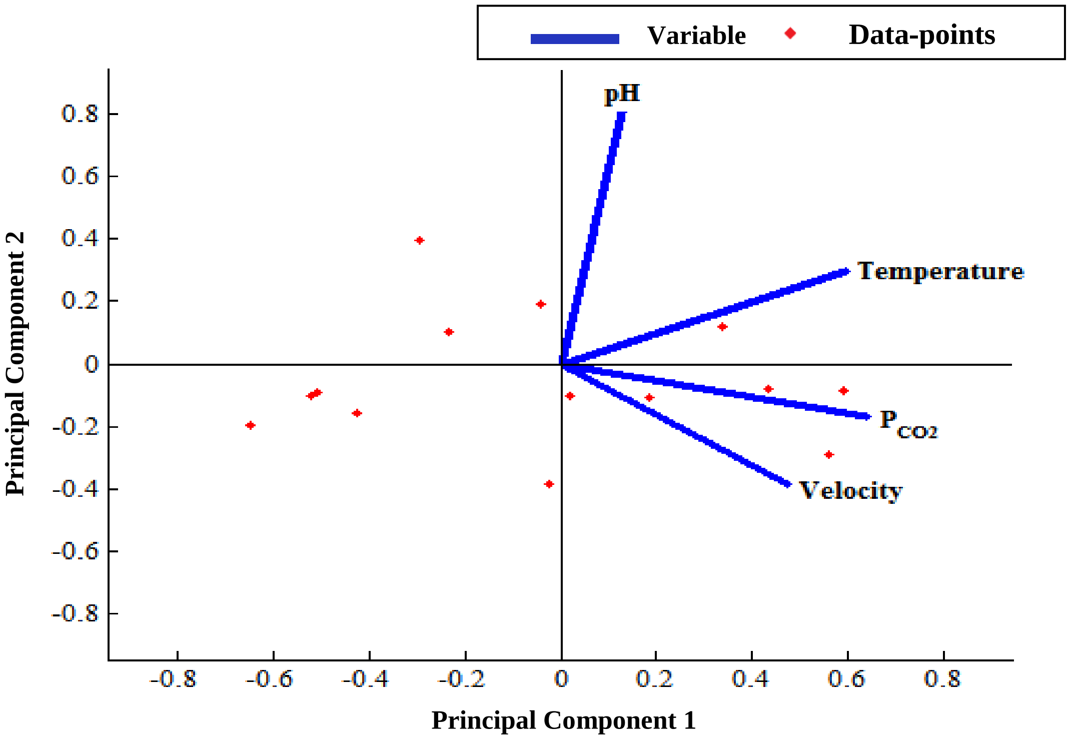

Fig.7. *Two-dimensional biplot showing the orientation of the variables in the principal component axes*

The biplot shows that the CO_2 partial pressure (P_{CO_2}) variable is the most dominant variable in terms of accounting for the variation within the dataset followed by the temperature (*T*), velocity (*U*) and pH respectively. This is because the direction of these three variables shows they are within close proximity to the Principal component 1 (PC1) axis. The length of each variable indicates their relative contribution to each principal component. It is also true that because the flowrate velocity and pH are less significant, owing to their smaller contributions to the variance of the dataset, they have a smaller impact on the statistical range or spread of the corrosion rate. Elevated

Thus, the P_{CO_2} variable combined with mid-temperature range values result in the highest corrosion rates in the dataset. The fluid flow velocity values for this dataset were binary, i.e. 0m/s or 4m/s. High flow velocities have a decreased effect on corrosivity for this data, and this may be explained by the formation of thicker scales associated with high CO_2 corrosion being resistant to flow erosion (Zhang et. al., 2012). It may also be the case that some sort of localised corrosion may be taking place at particular sites along the steel surface, which is however, not contributing to the overall uniform corrosion being investigated. For the effect of pH, an increase in pH leads to a drop in the corrosion rate, which is because the higher the pH, the lesser its influence on CO_2 corrosion as the acidity of the aqueous medium decreases.

2.3 Design of the neural network model

Fig. 8 shows the schematic that was used to develop the procedural operation of the neural network (NN) model.

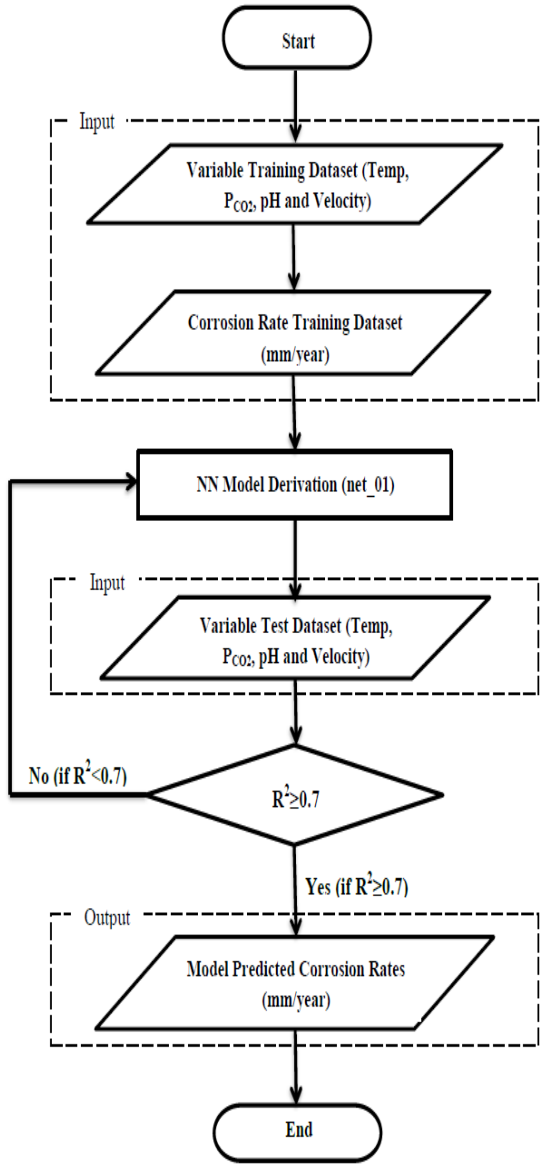
Start
Input
Variable Training Dataset (Temp, P_{CO_2}, pH and Velocity)
Corrosion Rate Training Dataset (mm/year)
NN Model Derivation (net_01)
Input
Variable Test Dataset (Temp, P_{CO_2}, pH and Velocity)
No (if R^2<0.7)
R^2≥0.7
Yes (if R^2≥0.7)
Output
Model Predicted Corrosion Rates (mm/year)
End

Fig. 8. *Flowchart showing the general sequence of steps in the development of the Neural_Network Model (net_01)*

There are two sets of input to the NN model. For the first input set (shown as long-dashed lines around the first 2 steps in Fig. 8), there is the variable training dataset as well as the corrosion rate training dataset. The first set of input to the NN comprises the temperature, CO_2 partial pressure (P_{CO_2}), pH and flow rate velocity. The second set of input to the NN is the corresponding corrosion rate data, as shown in Fig. 9, and is the target (see Appendix, Table A1). The training of these variables and the corresponding corrosion rates together constitute the first part of training.

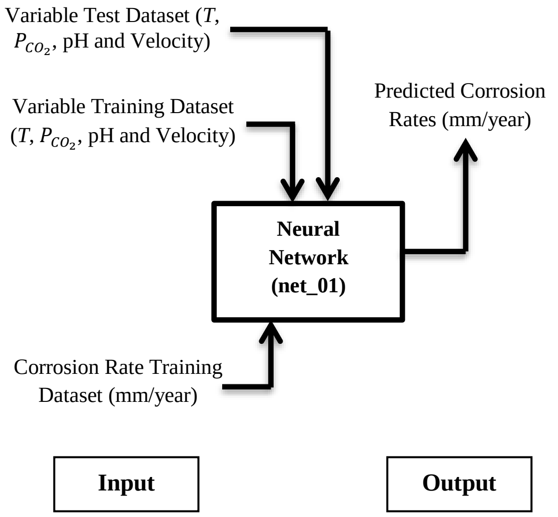
Variable Test Dataset (T, P_{CO_2}, pH and Velocity)
Variable Training Dataset (T, P_{CO_2}, pH and Velocity)
Predicted Corrosion Rates (mm/year)
Neural Network (net_01)
Corrosion Rate Training Dataset (mm/year)
Input
Output

Fig. 9. *Block diagram of the Neural Network Model (net_01)*

These corrosion rate data are direct laboratory measurements from corrosion rate experiments based on the conditions in the aforementioned variable training datasets: (Choi and Nesic, 2009; Hesjevik et. al., 2003; Zhang et. al., 2012; Cui et. al., 2006). The NN model, net_01, is shown in Fig. 8 as a process step in the flowchart sequence because the model is developed and defined by the two sets of input preceding it. The model is then used for corrosion rate prediction. The variable test dataset is contained within the second input set and is used for testing (see Appendix, Table A2). It contains parameters identical to those in the variable training dataset. The model reads this data and based on learned knowledge from the first input, yields output. These output results are then compared against actual corrosion rates from the original data in order to compute the correlation coefficient (R^2-value). This statistic is a common criterion for goodness of fit for regression models and indicates how well model predictions match up against raw corrosion rate input data, and is shown in Fig. 8 as the decision step (Draper and Smith, 1998), (Abyaneh, 2014). The condition, $R^2 \geq 0.3$, was chosen in the initial testing phase in order to characterise the selected training functions to be used: a low-level screening process to rule out undesirable training functions early. For fine-tuning and obtaining the derived NN model result, however $R^2 \geq 0.8$ was used, for deriving a highly accurate model. The actual corrosion rates are a direct consequence of the parametric conditions contained in the variable test dataset, measured from experimental corrosion rate plots. The errors are calculated by subtracting the actual (experimental) CO_2 corrosion rates from model predicted CO_2 corrosion rates. Absolute values of these errors are then evaluated and then results are summed up to obtain the 'Sum Total of Absolute Error'. Equations for these expressions are shown as follows:

$$\begin{aligned} &Corrosion\ Rate\ Error = \\ &Model\ Predicted\ Value - Actual\ Value \quad (1) \\ &\qquad = y_i - x_i \end{aligned}$$

For any given plotted point i:

$$\begin{aligned} Absolute\ Error &= |Corrosion\ Rate\ Error| \quad (2) \\ &= |y_i - x_i| \end{aligned}$$

For a given plotted point i:

$$\begin{aligned} &Sum\ Total\ of\ Absolute\ Error \\ &\qquad = \sum (Absolute\ Error) \quad (3) \\ &\qquad = \sum_{i=1}^{n} |y_i - x_i| \end{aligned}$$

Where x_i and y_i are the i^{th} observation and model predicted values respectively. (Draper and Smith, 1998; Abyaneh, 2014).

2.4 NN training – determination of model parameters

In order to determine the model parameters such as size of the network, the number of layers (network configuration) and the type of training function(s) to use, simulation runs were performed using Matlab 2012a version on a Windows 10-based computer.

Given that in any given neural network model, the transfer function converts input data to a given output, trial runs to determine the derived NN model, for the purpose of this study, are based on the type of transfer function.

Hence simulation tests were carried out on two main groups – the logsig and tansig transfer function groups. The reason for their selection as the head of their respective groups is that their respective algorithms allow them to accept data of any magnitude (negative to positive infinity) whilst returning an output in the range 0 to 1 for the former and -1 to 1 for the latter, thus making them the most suitable to use as a starting transfer function in the network (Beale et. al., 2014). Training algorithms are a set of instruction code that governs how the NN will be optimised such as the setting of weights and biases, for instance (Beale et. al., 2014). There are three types of training algorithms and these are: the gradient descent, conjugate gradient and quasi-Newton algorithms (Sharma and Venugopalan, 2014).

The gradient descent algorithms evaluate function gradients at the initial estimate value and take steps proportional to the negative direction of the gradient in order to find a local minimum. In conjugate gradient methods, the local minimum is located by searching along conjugate directions. This implies that a faster convergence is achieved than with the gradient-descent method. The quasi-Newton algorithms usually give faster optimisation than conjugate methods however a greater amount of memory is required (Sharma and Venugopalan, 2014).

Training functions are network functions that dictate a universal algorithm which sets weights and biases in any given network to optimise performance. Matlab training functions can be classified based on the type of training algorithm they use – all Matlab training functions thus fall into one of the three given training algorithms (Sharma and Venugopalan, 2014).

For this study, two training functions were selected from each of the training algorithm groups to make a total of six functions. The definitions of each of these functions are shown in Table 1.

Table 1. Definitions of the Training Functions Used (Beale et al., 2014), (Sharma and Venugopalan, 2014; Vogl et al., 1988)

Training Function	Training Algorithm Group	(Definition) Updates network weights according to:
Traingdm	Gradient Descent	Gradient descent with momentum
Trainrp	Gradient Descent	Resilient backpropagation (Rprop)
Trainscg	Conjugate Gradient	Scaled conjugate gradient method
Traincgf	Conjugate Gradient	Conjugate gradient backpropagation with Fletcher-Reeves updates
Trainbfg	Quasi-Newton	Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method
Trainlm	Quasi-Newton	Levenberg-Marquardt optimisation

The testing phase of the NN model development involved the use of two transfer functions, each with several training algorithms.

3. Results

The variation of the magnitudes of R^2-coefficients of the training functions with number of neurons for the logsig transfer function is shown in Fig. 10. The maximum number of neurons tested was set at 20 to avoid overtraining. It must be noted that the simulation tests discussed in this section are carried out with respect to the total number of neurons in all hidden layers. Neuron configuration is discussed in Section 4. There is a general decrease in the magnitude of the R^2-coefficients with increasing number of neurons for all training functions except for the Trainscg. This fall in the magnitude of the correlation coefficient with increasing number of neurons indicates that model performance is decreasing despite the increasing network size. This may be due to the presence of a greater number of redundant nodes or synaptic weights in the NN or that training is possibly diverging, resulting in predicted values being very different from expected results (Haykin, 1999). Other reasons may be that the training dataset is small or that the inherent characteristic of the data is such that modelling with an increasing number of neurons beyond 5, results in a fall of the R^2-value.A similar trend was observed in an environmental research study (Abyaneh, 2014).

For the tansig transfer function, Fig. 12 is the variation of the magnitudes of R^2-coefficients against number of neurons.

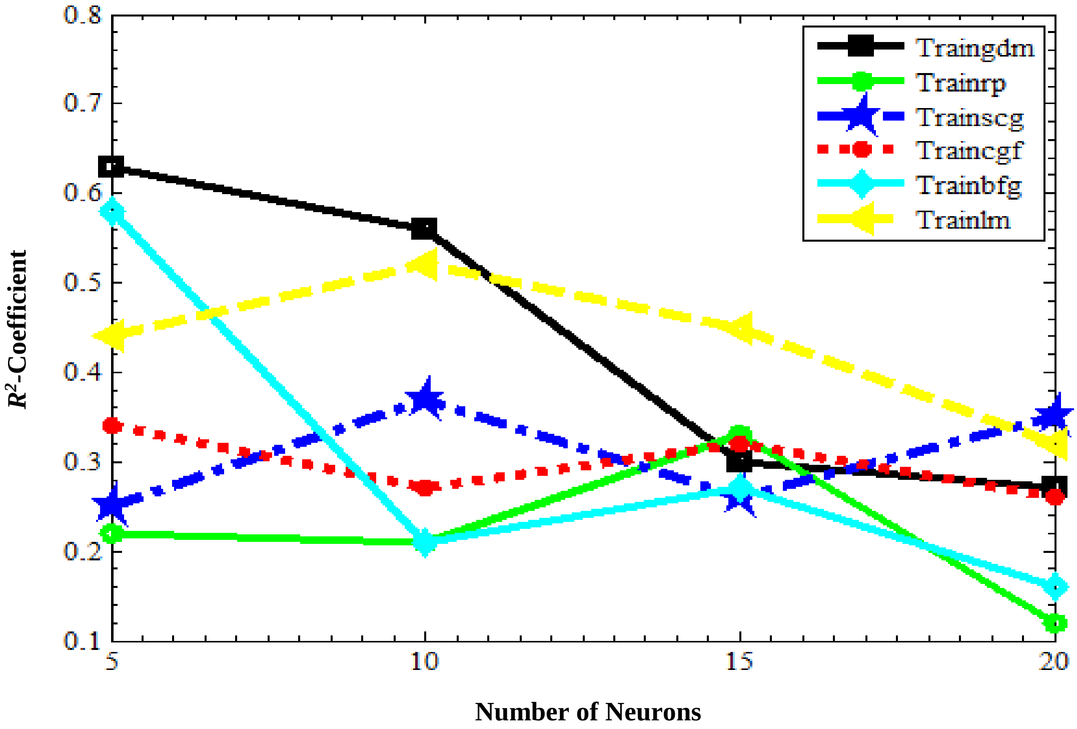

Fig. 10. Line Plots of R^2-values of Training Functions against Total Number of Neurons in Hidden layers for the Logsig Transfer Function

There is a general dip in the magnitudes of the R^2-coefficients as the number of neurons increases from 5 to 20. Again, given that the training dataset is small, there is greater certainty that as the number of neurons increases in the NN, the number of excess weights also increases and there is a tendency for their presence to reduce model accuracy (Haykin, 1999).The only exceptions are the quasi-Newton algorithms, trainbfg and trainlm. The quasi-Newton algorithms are generally known to give better optimisation results than the others, particularly for network sizes that are not large (Beale et. al., 2014; Sharma and Venugopalan, 2014).

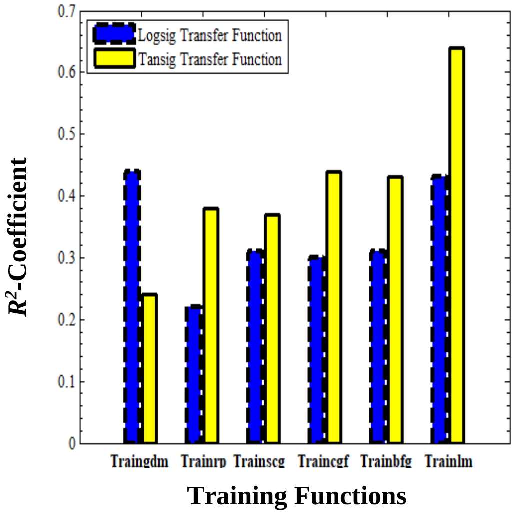

***Fig. 11.** Bar Charts Showing the Means of the R^2-values of Each Training Function for Both Logsig and Tansig Transfer Functions*

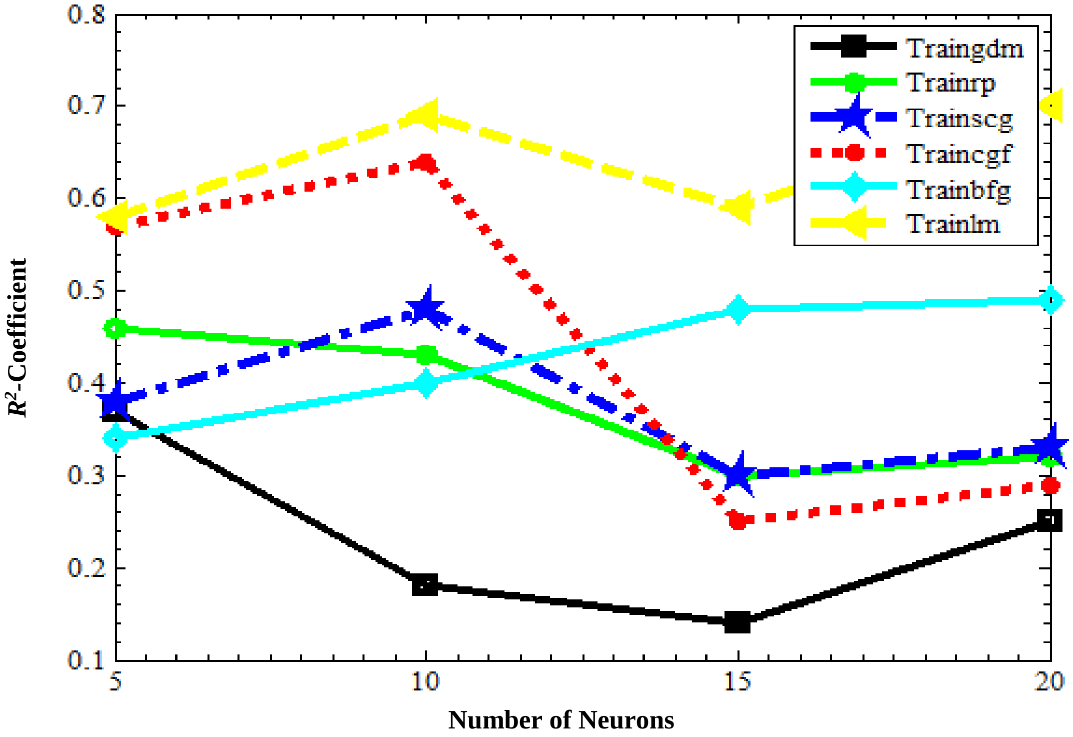

Fig. 12. *Line Plots of R^2-values of Training Functions against Total Number of Neurons in Hidden layers for the Tansig Transfer Function*

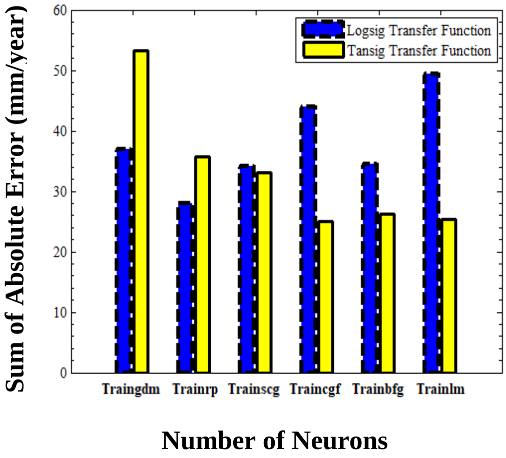

Fig. 13. *Bar Charts Showing the Sums of Absolute Corrosion Rate Error of Each Training Function for Both Logsig and Tansig Transfer Functions*

Figure 11 shows the performances of each training function for both logsig and tansig transfer functions in terms of the average R^2-coefficient. The trainlm function again outperforms all the others. The traincgf and trainbfg rank second and third respectively on the R^2-value performance test. It is also worth noting that the training functions mostly perform better with the tansig transfer function than the alternative logsig transfer function, the only exception being the traingdm function. This may be due to the effect of the unique combination of the input data, the logsig transfer function and the algorithm in the traingdm training function culminating in a good performance level. Overall, a steady, high magnitude of R^2-coefficient is maintained by the trainlm function with an increasing number of neurons. This indicates its suitability for use in developing the derived NN model. The training functions performances were also assessed in terms of the sum of their absolute errors. Fig. 13 is the bar chart variation of the sum of absolute errors of the training functions against number of neurons. It is seen that the traingdm has the least favourable performance in terms of the mean absolute corrosion rate errors obtained while the trainrp and trainbfg have the smallest errors for both logsig and tansig transfer functions as a combination. On a singular basis, the trainlm has the least error when the tansig function is in use, followed closely by the traincgf function. In summary, the best training functions appear to be the trainlm and traincgf. It must be borne in mind that for all modelling tests conducted and discussed so far, the condition applied is that $R^2 \geq 0.3$. The only exception is the test for the sum of absolute square errors, where, $R^2 \geq 0.7$ was applied.

4. Discussion

4.1 Discussion of the Developed NN Model Specifications

The two functions used in this stage are the traincgf and trainlm functions. The best performances were obtained when the tansig transfer function was applied. In the initial testing stages, it was discovered that very few neurons were required to achieve desirable results as shown in Figures 10 and 11. In particular, the NN performances were relatively high for neuron numbers of 5 to 15. The number of hidden layers to be used was determined by testing five (5) different neuron configurations in multiples of 5 as shown in Table 2. Fig. 14 shows the comparative line plots for both traincgf and trainlm functions.

Table 2. *Summary of the Results Obtained for the Tested Neuron Configurations*

Neural Network Property	**Value/Function**
Number of hidden neurons and configuration	10 neurons, configuration= [5 5]
Number of hidden layers	2
Training function	Levenberg-Marquardt (Trainlm)
Transfer functions	Tansig-Tansig

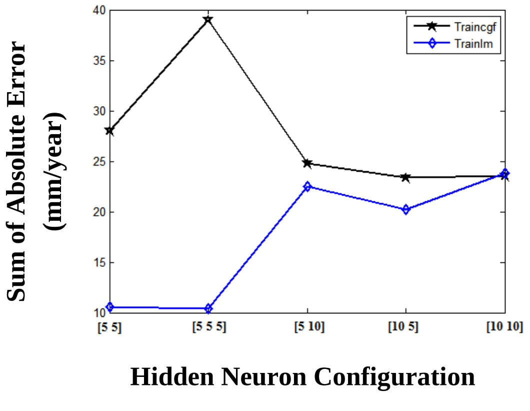

Fig. 14. *Line Plots of Sum of Absolute Corrosion Rate Error against Neuron configuration for the Traincgf and Trainlm Functions*

From Fig. 14, the trainlm function consistently has the least absolute corrosion rate errors implying that it is a better training function for developing the derived NN model. It is also noted that NNs with one hidden layer often encounter difficulties with approximation and model-fitting due to global interaction between neurons. On the other hand, NNs with two hidden layers have no such issues, as the first layer extracts local features within the data while the second layer extracts the global features (Haykin, 1999).

Table 3. *Summary of Neural Network Properties*

Hidden Neuron Configuration	**Total Number of Hidden Neurons**	**Sum of Absolute Error (mm/year)**	**R^2-value**
[5 5]	10	10.61	0.91
[5 5 5]	15	10.38	0.88
[5 10]	15	22.46	0.68
[10 5]	15	20.19	0.76
[10 10]	20	23.82	0.80

It should be noted that the overall NN comprises 4 input neurons, corresponding to the input variables (see Section 2.3, Figure 8), the hidden neuron configuration just discussed, as well as one output neuron, i.e. (4 [5 5] 1). Also, from Table 3, the [5 5] hidden neuron layer configuration offers the highest R^2-value. This value coincides with the smallest corrosion rate error, as seen in Fig. 14. This corrosion rate error is equivalent to 10.61/6 ≈ 1.77mm/year per plotted data-point on the NN model corrosion rate against experimental corrosion rate plot shown in Fig. 15. The plots in Fig. 15 and Fig. 16 show the performance of the 10-neuron, [5 5] configuration, 2-hidden layer neural network model, when tested against an unseen dataset (see Appendix, Table A2). The magnitude of the R^2-value and the closeness of all data-points to the 95% confidence intervals in Fig. 15 indicate the model is accurate. The plot of experimental and predicted corrosion rate against pH is shown in Fig. 17. There is a decrease in corrosion rates as pH increases and this follows the trend for CO_2 corrosion rate vs pH for low pressure CO_2 systems (Dugstad et. al., 1994). In particular, the trend more closely resembles the high ferrous ion concentration ($[Fe^{2+}]$) curve for corrosion rate vs pH, which results because supersaturation happens more readily. It is noteworthy that supersaturation is more likely to occur for high pressure corrosion systems (Zhang et. al., 2012). The reasoning behind the trend is that lower pH-environments bring about a more corrosive medium while higher pH-environments are less severe in corrosivity. The curve is an exponential equation with an R^2-coefficient of 0.7, for the NN predicted values. This indicates that there is a good correlation between pH and the NN-predicted corrosion rate and as such the former can be used to predict the latter fairly accurately, using the following equation:

$$Corr\ Rate = (6.92 \times 10^4) \times e^{(-2.72 \times pH)} \quad (4)$$

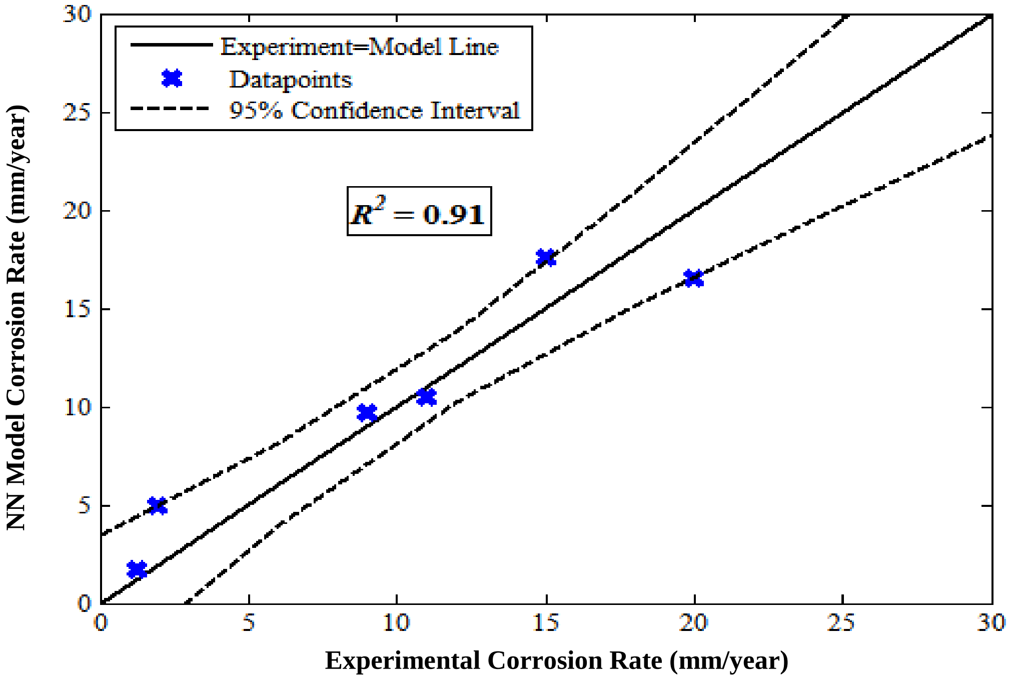

Fig. 15. *NN Model Corrosion Rate against Experimental Corrosion Rate plot*

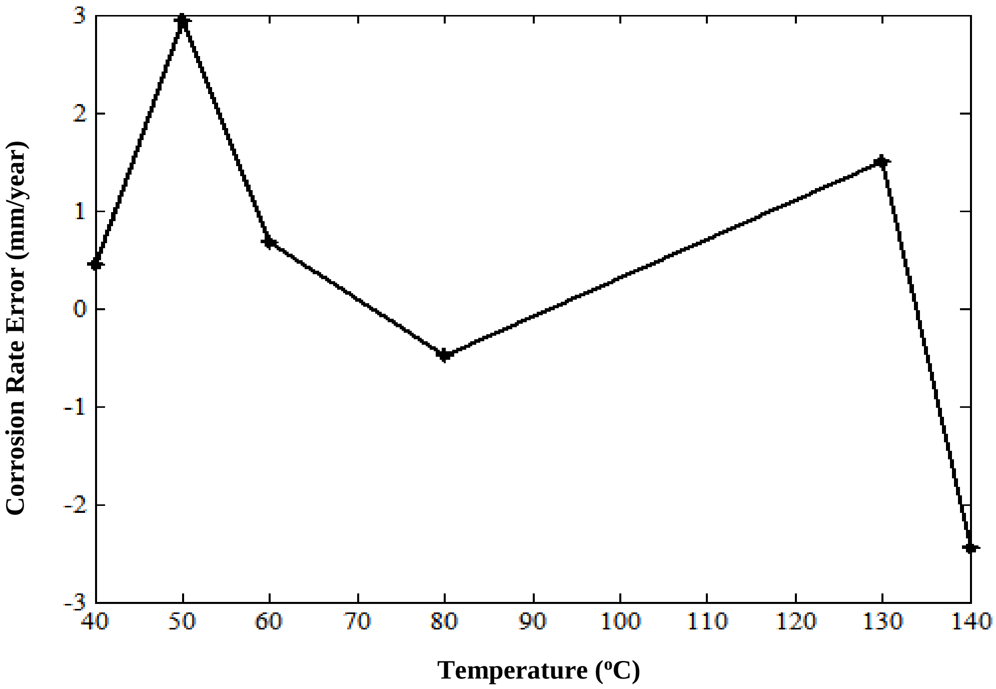

Fig. 16. *Error Line Plot – Error is the difference between Experimental and Model Corrosion Rate*

Figure 18 shows the corrosion rate-temperature profile for the test dataset with corresponding NN model predictions. The best fit line is a polynomial function with an R^2-coefficient of 0.55. A higher R^2-value can be attained however this results in a distortion of the overall shape and is badly conditioned. There is an increase in corrosion rates as temperatures increase from 20°C to 80°C owing to temperatures accelerating the chemical and electrochemical corrosion reactions. The precipitation rate though is said to increase as temperatures increase hence protective layers form on the metal surface leading to a reduction in corrosion rate (Johnson and Tomson, 1991; Yin et. al., 2009). NN predictions are close to the test data results shown in Fig. 18, appearing mostly as pairs of data-points. There is a slight degree of inaccuracy in model predictions for the mid-temperature region of 50-80°C and this is attributed to greater variation in the training and test set corrosion rates. For the corrosion rate against P_{CO_2}, as shown in Fig. 19, the predicted NN results predict the experimental data to within 3.5mm/year. The reason a curve fit was not used is because for low pressure there is a near-linear relationship for CO_2 partial pressures less than 1 MPa (Dugstad et. al., 1994), while for high pressures, such as this one, the relationship is non-linear.

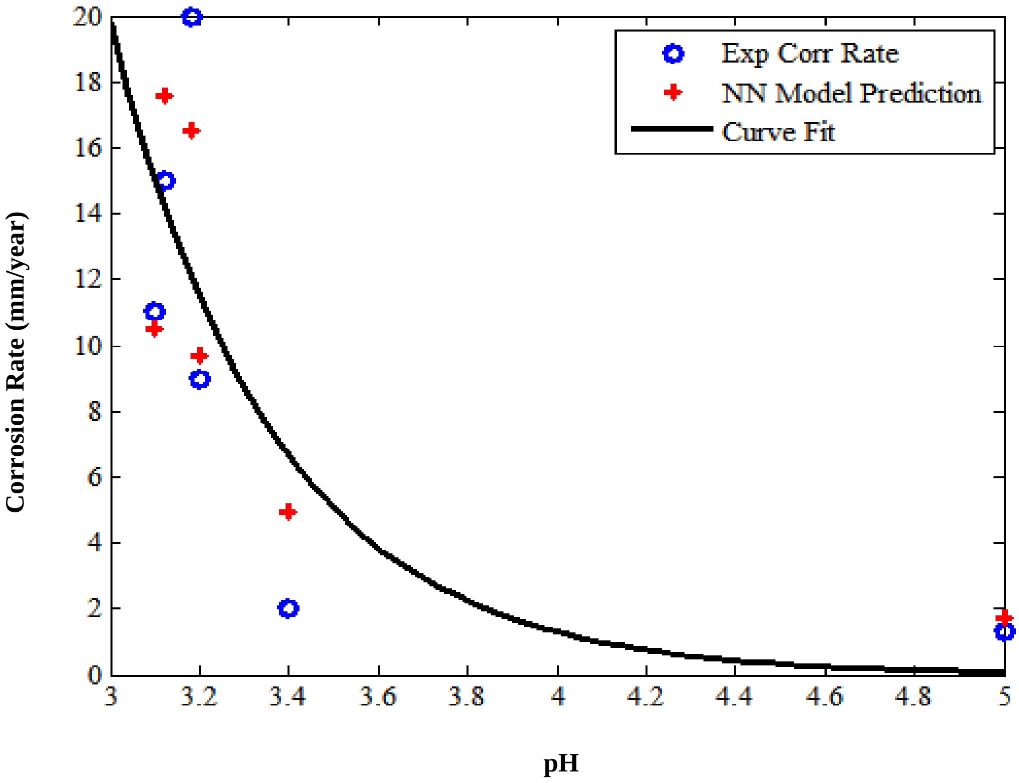

Fig. 17. *Corrosion Rate against pH for the testing dataset with NN Model predictions*

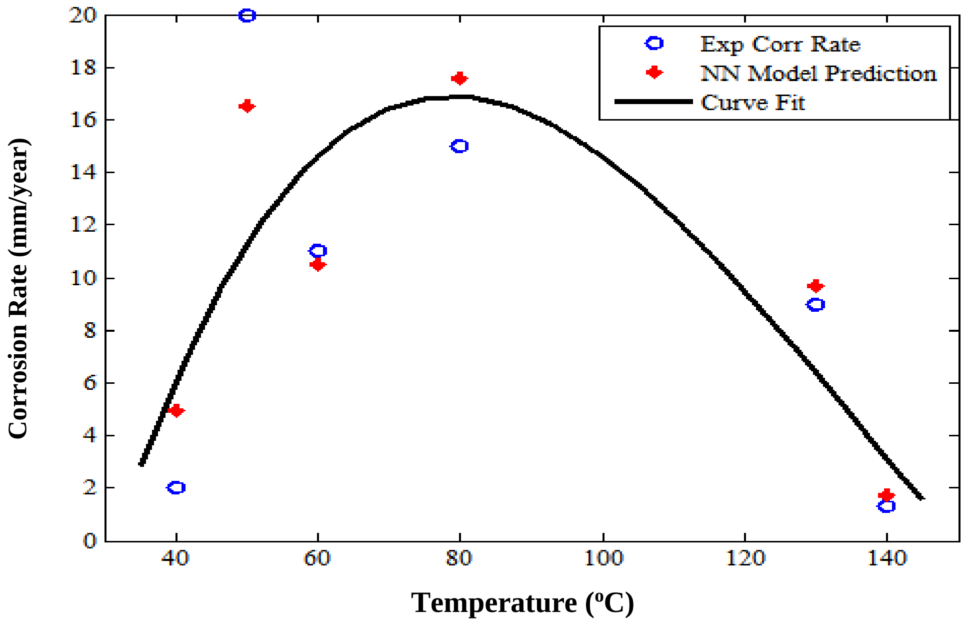

Fig. 18. Corrosion Rate against Temperature plot for the testing dataset with NN Model predictions

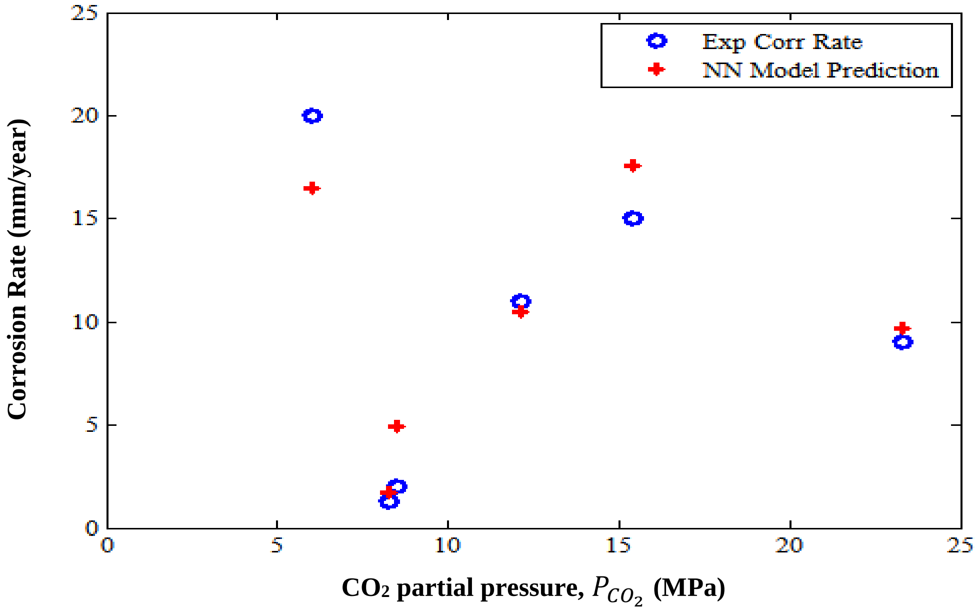

Fig. 19. *Corrosion Rate against CO_2 partial pressure (P_{CO_2}) for the testing dataset with NN Model predictions*

4.2 Discussion of Cross-Validation on the NN Training Data

The method described in this section is carried out independently of the model derived in the previous section, Section 4.1. The NN model being derived here, with the aid of cross-validation shares some properties with the previous model and these are shown in Table 2. The accuracy of the previous model has been highlighted therefore the newly derived model is loosely based on it. As the size of the dataset used in the development of the NN model in the previous section is small, cross-validation then serves as a means for assessing the NN model performance against potential new test data.

In *k*-fold cross-validation, the dataset is partitioned randomly into *k* smaller subsets or folds of roughly equal sized parts; one of these folds becomes the designated testing set and the remaining *k*-1 folds are subsequently used to train the algorithm (Hastie et al., 2017; Kohavi, 1995). In addition, this technique is useful when modelling with scarce data (Hastie et al., 2017). In particular, the leave-one-out cross validation (LOOCV) technique was implemented in this paper. This is a special case of the *k*-fold technique in which the dataset assigned for training is partitioned into the total number of data-points, $k=N$ (Hastie et al., 2017; James et al., 2013). LOOCV was performed on the 16 data-points of the training dataset and fed to the NN model (see Appendix, Table A1). In the whole process, a single randomly selected validation set is used as a basis for the evaluation of the absolute error, in a rotation estimation manner (Kohavi, 1995). Fig. 20 shows the line plot of absolute errors against the number of model simulations. The absolute error is calculated using the formula shown in equation 2. In general, the line plot undulates about the mean absolute error, MAE, (dashed line), which is approximately 4.5 mm/year. Ten models, i.e. Models 1, 2, 4, 5, 6, 7, 11, 13, 14 and 15 have absolute errors that are less than the MAE, and the final model was selected from these.

Models 4, 7 and 11 are the strongest performing models however, in this instance; Model 13 was selected, with an absolute corrosion rate error of ~2.66mm/year. Though the aforementioned models perform better, Model 13 was selected as it will be expected to generalise better to new data. The strong-performing models are more likely to exhibit data overfitting than the selected model as NN models have a tendency for overfitting (Pasini, 2015). Fig. 21 shows the model prediction for the unseen test set (see Appendix, Table A2). The selected model over-predicts for low corrosion rates and is more accurate for the larger corrosion rate values. A prediction interval is included to indicate an estimate of the region in which other high pressure CO_2 corrosion data is likely to fall. There is a high degree of certainty in the selected model that data-fitting is accurate, indicated by the four points within the 95% confidence interval. The 95% prediction interval covers all points except one. The location of this point, (2.0, 12.1) mm/year re-emphasises the fact that the model is less accurate for low corrosion rate values. In addition, the root mean square (RMSE) and R^2-values were evaluated as statistical metrics in assessing model performance. The RMSE metric has the same scale as the parameter of interest, is a measure of model accuracy, and is a popular choice for the comparison of prediction errors for different models on the same dataset (Hyndman and Koehler, 2006). The computed generalisation errors are RMSE=5.45mm/year and R^2=0.61.

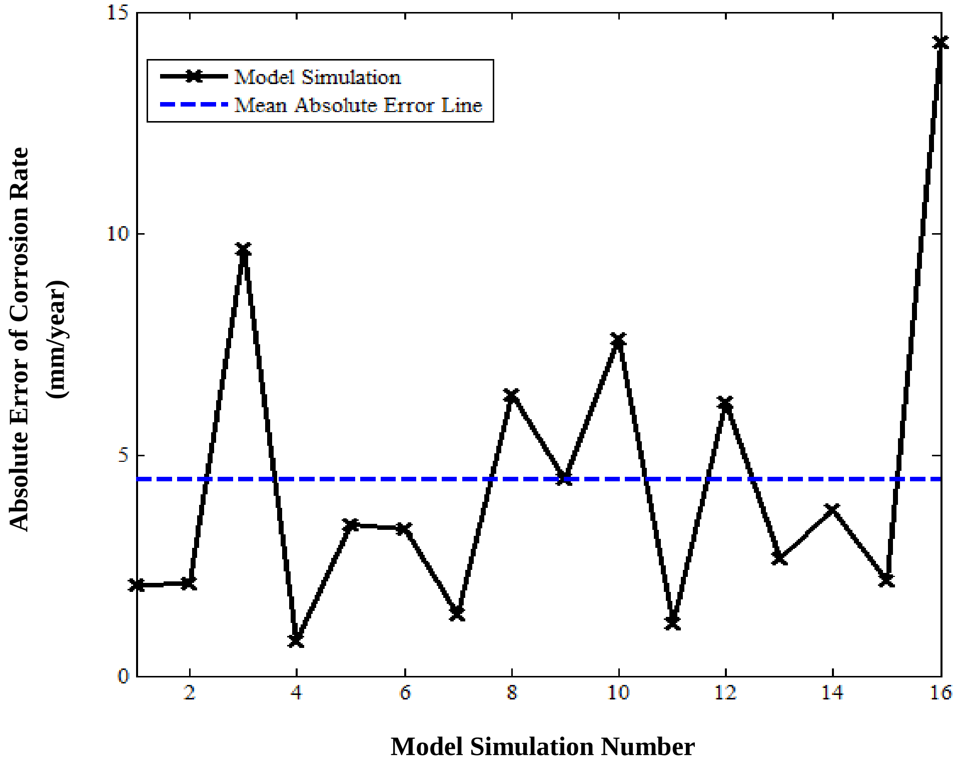

***Fig.20**. Line plot of Absolute Error in Corrosion Rate against LOOCV Model Simulation Number*

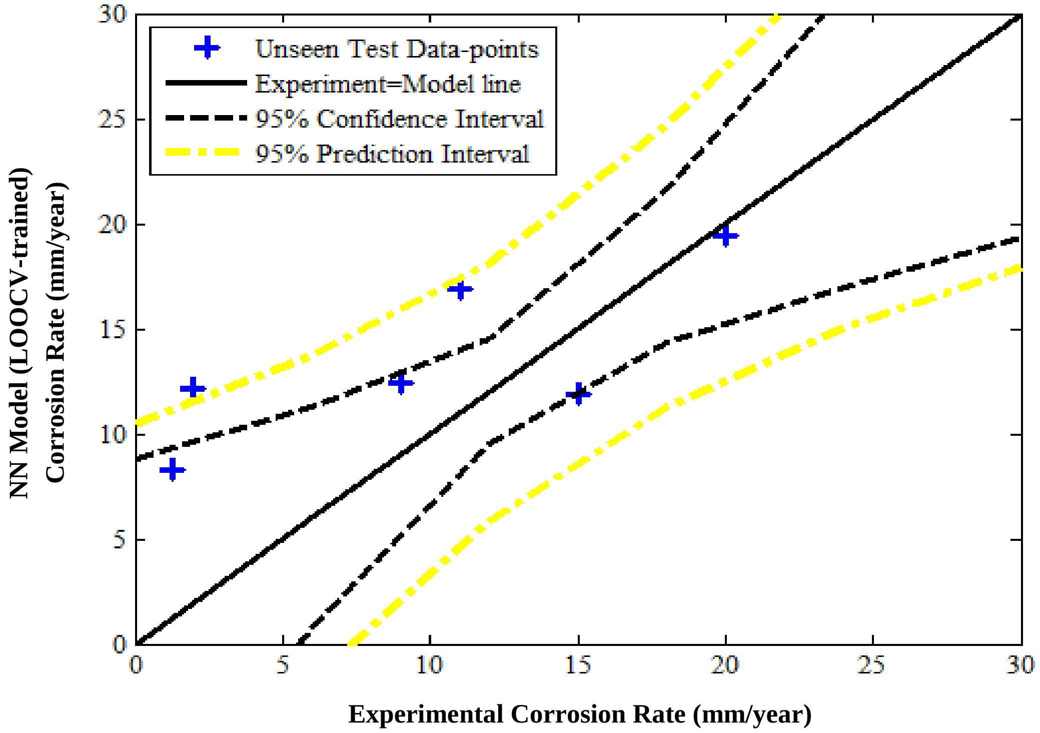

Fig. 21. *NN Model (LOOCV-trained) Corrosion Rate against Experimental Corrosion Rate for the testing dataset*

5. Conclusion

The tansig transfer function consistently outperformed the logsig transfer function for most training functions with the exception of the traingdm and trainrp training functions, in terms of the absolute corrosion rate errors. As these two algorithms are gradient descent functions, their under-performance relative to the conjugate gradient and quasi-Newton algorithm function alternatives is to do with the technique of finding and locating local minima while carrying out nonlinear optimisation.

Also, in terms of the magnitude of the R^2-coefficent, the tansig transfer function offered better results for all training functions except the traingdm. Again, the quasi-Newton algorithm functions, trainbfg and trainlm are known to offer better optimisation results.

The results from PCA on the data indicate that the CO_2 partial pressure and temperature contribute more greatly to the variance in the dataset than the velocity and pH. Given that the dataset was gathered from static laboratory tests and the flow velocities were mostly 0m/s, it is therefore understandable that this variable has a small effect on the variance. For the pH, the greater the acidity of the medium, the higher the corrosive risk, particularly for high pressure scenarios, however, the results from PCA, suggest uniform corrosivity which is the focus of this paper, is not greatly affected by the low pH. There may be some underlying localised corrosion taking place or that the effect of scaling somewhat diminishes the effect of this variable, when compared with the effects of partial pressure and temperature. In any case, an exponential relationship was derived between the pH and NN predicted corrosion rate. Prediction from the equation was fairly accurate with an R^2-coeffcient of 0.7.

As is usually the case with CO_2 corrosion, the mid-temperature experimental values (50-80ºC) induce the greatest corrosion rates due to the Arrhenius theory - the fact that higher temperatures speed up corrosion rates by providing greater amounts of kinetic energy to reacting species on a molecular level. However the limiting factor preventing a continuous induced elevation of CO_2 corrosion rates beyond levels of the mid-temperature values to higher levels of the very high end-point temperatures (110-150ºC) is the consequent formation of iron carbonate ($FeCO_3$). Although the [Fe^{2+}] increases in solution with higher temperatures, solubility decreases, thereby resulting in scale formation which coats the metal surface and brings about a decrease in corrosion rates.

This phenomenon reflects on both training and testing datasets with mid-point temperatures coinciding with very high corrosion rates. Also, in these datasets, 50% of the data-points lie in the 50-80ºC range, which represents a statistical range of ~11mm/year in the training dataset (8.1-19.0 mm/year, see Appendix, Table A1) and this represents a large difference between the corrosion rates within the temperature range between the minimum and maximum corrosion rates. Hence, there is a greater variance (spread) in this range.

NNs tend to predict less accurately when trained with data of considerable variation. This is depicted in the model vs experimental plot where the high corrosion rate points are on opposite bounds of the 95% confidence limits while other data-points are not as widely-spaced apart on the plot.

The R^2-coefficient for the derived NN model is 0.91 and provides a good fit for the CO_2 corrosion test set. All test points are within the bounds of the 95% confidence limits indicating a high degree of accuracy. The error line plot indicates that predicted results are within ±3mm/year of the CO_2 corrosion test set. The NN model was trained with a small dataset, and though care was taken to ensure the test dataset was unseen, this poses a limitation.

To compensate for the use of a small dataset for training and testing, LOOCV was carried out as a means for assessing model performance and selection. Both RMSE and R^2-values were evaluated as statistical metrics to summarise the selected model, though the former is known to be sensitive to the presence of outliers. As NN models have a tendency for overfitting, the number of hidden neurons being analogous to the degree of the polynomial in a polynomial regression, it is necessary to select a model that achieves some degree of generalisation. The 95% prediction interval was included to show where data from other sources are likely to lie when compared to the results discussed in this paper.

It is also worth noting that the training and testing data come from multiple sources conducted under various experimental conditions and in different environments. These factors introduce uncertainty to modelling and as such impose some limitations on results.

6. References

Abbas, M. H. (2016). *Modelling CO_2 Corrosion of pipeline steels*, PhD Thesis, University of Newcastle upon Tyne, School of Marine Science and Technology, Newcastle upon Tyne. Chapter 1, Pages 2, 3.

Abyaneh, Z. (2014). *Evaluation of Multivariate Linear Regression and Artificial Neural Networks in Prediction of Water Quality Parameters*, Journal of Environmental Health Science & Engineering, Volume 12, Issue 40.

Beale, M. H., Hagan, M. T., DeMuth H. B., *Neural Network Toolbox User's Guide R2014b*. The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098. 2014, Pages 2-4, 2-5, 2-17, 2-18, 2-19.

Choi, Y-S., Nesic, S. (2009). *Corrosion Behavior of Carbon Steel in Supercritical CO_2-Water Environments,* NACE International, Corrosion 2009, Paper 09256, Atlanta, Georgia, USA.

Choi, Y-S., Nesic, S. (2011). *Determining the Corrosive Potential of CO_2 Transport Pipeline in High pCO2-Water Environments*, International Journal of Greenhouse Gas Control, Volume 5, Issue 4, pp. 788-797.

Choi, Y-S., Nesic, S., Young, D. (2010). *Effect of Impurities on the Corrosion Behavior of CO_2 Transmission Pipeline Steel in Supercritical CO_2-Water Environments*, Journal of Environmental Science and Technology, Volume 44, No. 23, pp. 9233-9238.

Cottis, R. A., Qing, L., Owen, G., Gartland, S. J., Helliwell, I. A., Turega, M. (1999). *Neural Network methods for Corrosion Data Reduction*, Materials and Design: 1999, Vol. 20, No. 4, pp. 169-178.

Cui, Z. D., Wu, S. L., Zhu, S. L., Yang, X. J. (2006), *Study on Corrosion Properties of Pipelines in Simulated Produced Water with Supercritical CO_2*, Journal of Applied Surface Science, Volume 252, Issue 6, pp. 2368-2374.

De Waard, C., Lotz, U. (1993). *Prediction of CO_2 corrosion of carbon steel*, NACE International, Corrosion1993, Paper No. 69 Houston, Texas, USA.

Dlugokencky, E. J., Hall, B. D., Montzka, S. A., Dutton, G., Muhle, J., and Elkins, J. W. (2014): [Global climate] Long-lived greenhouse gases [in *State of the Climate in 2013*]. Bulletin of the American Meteorological Society, Volume 95, No.7, S33–S34.

Draper, N. R.; Smith, H. (1998). *Applied Regression Analysis*. Wiley-Interscience, Chapter 1, pp.15-20, 40-44.

Dugstad, A., Lunde, L. and Nesic, S. (1994). *Control of Internal Corrosion in Multi-phase Oil and Gas Pipelines*, Prevention of Pipeline Corrosion Conference, Pipe Line Industry and Pipes & Pipelines International, Houston, Texas, USA.

Hastie, T., Tibshirani, R., Friedman, J. (2017). *The Elements of Statistical Learning, Data Mining, Inference and Prediction*, Springer Series in Statistics, Second Edition, Chapter 7, pp. 241-245.

Haykin, S. (1999). *Neural Networks – A Comprehensive Foundation*, Pearson Education Ltd., Second Edition, Chapter 4, pp. 234-244.

Hesjevik, S. M., Olsen, S., Seiersten, M. (2003). *Corrosion at High CO_2 Pressure*, StatOil and Institute for Energy Technology in Norway, NACE International, Corrosion 2003, Paper 03345, San Diego, California, USA.

Hyndman, R. J., Koehler, A. B. (2006). *Another Look at Measures of Forecast Accuracy*, International Journal of Forecasting, Volume 22, Issue 4, pp. 679-688.

International Energy Agency (IEA), 2010: World Energy Outlook 2010, pp.418.

International Energy Agency Greenhouse Gas (IEA GHG), CO_2 pipeline Infrastructure: An analysis of global challenges and opportunities report, Conference /2010, pp. 12, 23-29 (Cambridge, UK, 2010).

IPCC, 2007: *Climate Change 2007: Synthesis Report. Contribution of Working Groups I, II and III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change* [Core Writing Team, Pachauri, R.K and Reisinger, A. (eds.)]. IPCC, Geneva, Switzerland, 104, pp. 17, 60.

IPCC, 2014: *Climate Change 2014: Synthesis Report*. *Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change* [Core Writing Team, R.K. Pachauri and L.A. Meyer (eds.)]. IPCC, Geneva, Switzerland, 151, pp. 46, 47.

Jackson, J. E., *A User's Guide to Principal Components*, John Wiley and Sons, 1991, p. 592.

James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*, [Casella, G., Fienberg, S., Olkin, I. (eds.)], Springer Texts in Statistics, Springer Science+Business Media, LLC, New York, Chapter 5, pp. 178-184.

Johnson, M. L., & Tomson, M. B. (1991). *Ferrous Carbonate Precipitation Kinetics And its Impact on CO_2 Corrosion*, NACE International, Corrosion 1991, Paper No. 268, Houston, Texas, USA.

Jolliffe, I. T., *Principal Component Analysis*, 2nd edition, Springer, Springer Series in Statistics, 2002, p.112, 113, 382-385.

Kohavi, R. (1995). *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*, Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI), Volume 2, Montreal, Quebec, Canada, Morgan Kaufmann Publishers Inc., San Francisco, California, USA, pp. 1137-1143.

Laboureur, L., Ollero, M., Touboul, D. (2015). *Lipidomics by Supercritical Fluid Chromatography*, International Journal of Molecular Sciences, Volume 16, Issue 6, pp.13868-13884.
doi: 10.3390/ijms160613868

Owen, G., Cottis, R. A., Turega, M. (2000). *Prediction of the Corrosion Rate of Steel in Seawater Using Neural Network Methods*, Corrosion 2000, Paper

00489, NACE International, Orlando, Florida, 2000.

Pasini, A. (2015). *Artificial Neural Networks for Small Dataset Analysis*, Journal of Thoracic Disease, Volume 7, Issue 5, pp. 953 -960.

Popoola, L. T., Grema, A. S., Latinwo, G. K., Gutti, B., Balogun, A. S. (2013). *Corrosion Problems during Oil and Gas Production and its Mitigation*, International Journal of Industrial Chemistry, Volume 4, Issue 1, Article 35, Springer, Springer Berlin Heidelberg, https://doi.org/10.1186/2228-5547-4-35

Radonja, J. Pero. (2001). *Neural networks based model of a highly nonlinear process*, Proceedings of ZX Telecommunications forum, TELFORZOOI, 20-22. November, 2001, pp. 435-438, Belgrade.

Radonja, P., Stankovic, S. (2002). *Neural Network Models Based on Small Data Sets*, 2002 6th Seminar on Neural Network Applications in Electrical Engineering, NEUREL 2002, pp. 101-106.

Rao, A., Rubin, E. (2002). *A Technical, Economic and Environmental Assessment of Amine-based CO_2 Capture Technology for Power Plant Greenhouse Gas Control*, Environmental Science & Technology, Volume 36, pp. 4467-4475.

Schmitt, G., Schütze, M., Hays, G. F., Burns, W., Han, E-H, Poubaix, A. and Jacobson, G. (2009). *Global Needs for Knowledge Dissemination, Research and Development in Materials Deterioration and Corrosion Control*, The World Corrosion Organization (WCO), Page 5.

Sharma, B. and Venugopalan, K. (2014). *Comparison of Neural Network Training Functions for Haematoma Classification in Brain CT Images*, Journal of Computer Engineering (JCE), Volume 16, Issue 1, pp. 31-35.

Suryanarayana, T. M. V., Mistry, P., B. (2016). *Principal Component Regression for Crop Yield Estimation*, 1st Edition, SpringerBriefs in Applied Sciences and Technology, Springer Nature Publishing, Springer Science+Business Media Singapore Plc Ltd., Springer, Chapter 2, pages 17-25.

Tans, P. (2009). *An Accounting of the Observed Increase in Oceanic and Atmospheric CO_2 and an Outlook for the*

Future. *Oceanography,* Volume 22, No.4, pp. 26–35.

Vogl, T. P., J.K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon. (1988). *Accelerating the Convergence of the Backpropagation Method*, Biological Cybernetics, Volume 59, pp. 257-263.

World Energy Council, Carbon Capture and Storage: A WEC “Interim Balance”, 2007, pp.17.

Yin, Z. F., Feng, Y. R., Zhao, W. Z., Bai, Z. Q., and Lin, G. F. (2009). *Effect of Temperature on CO_2 Corrosion of Carbon Steel*, Journal of Surface and Interface Analysis, Volume 41, Issue 6, pp. 517-523.

Zhang, Y., Gao, K., Schmitt, G. and Hausler, R. H. (2012). *Modeling Steel Corrosion under Supercritical CO_2 Conditions*, Journal of Materials and Corrosion, Volume 64, Issue 6, pp.478-485. DOI: 10.1002/maco.201106382

7. Appendix

Table A1. *Training Dataset*

T (ºC)	P_{CO_2} (MPa)	U (m/s)	pH	Corr. Rate (mm/year)	Source
24.00	3.50	0.00	3.10	4.00	Hesjevik et al., 2003
40.00	5.80	0.00	3.40	8.00	Hesjevik et al., 2003
50.00	4.00	0.00	3.25	18.00	Choi and Nesic, 2009
50.00	8.00	0.00	3.14	19.00	Choi and Nesic, 2009
50.00	9.50	4.00	5.10	9.00	Zhang et al., 2012
50.00	9.50	0.00	5.10	8.10	Zhang et al., 2012
60.00	12.10	4.00	3.10	15.00	Zhang et al., 2012
60.00	12.10	4.00	4.08	12.00	Zhang et al., 2012
60.00	12.10	0.00	4.08	10.50	Zhang et al., 2012
80.00	15.40	4.00	4.00	15.00	Zhang et al., 2012
110.00	20.30	4.00	4.00	13.00	Zhang et al., 2012
110.00	20.30	4.00	4.00	13.00	Zhang et al., 2012
130.00	23.30	4.00	3.20	9.00	Zhang et al., 2012
130.00	23.30	4.00	3.90	9.00	Zhang et al., 2012
130.00	23.30	0.00	3.90	9.00	Zhang et al., 2012
150.00	8.27	0.00	6.00	0.90	Cui et al., 2006

Table A2. *Unseen (Final Testing) Dataset*

T (ºC)	P_{CO_2} (MPa)	U (m/s)	pH	Corr. Rate (mm/year)	Source
40.00	8.50	0.00	3.40	2.00	Hesjevik et al., 2003
50.00	6.00	0.00	3.18	20.00	Choi and Nesic, 2009
60.00	12.10	0.00	3.10	11.00	Zhang et al., 2012
80.00	15.40	4.00	3.12	15.00	Zhang et al., 2012
130.00	23.30	0.00	3.20	9.00	Zhang et al., 2012
140.00	8.27	0.00	5.00	1.30	Cui et al., 2006