SSRG International Journal of Chemical Engineering Research
ISSN: 2394 – 5370 /doi:10.14445/23945370/IJCER-V8I1P103

Volume 8 Issue 1, 15-21, Jan-April 2021

Predicting Water Content Outcomes in Natural Gas Dehydration Systems Using Artificial Intelligence

Olayemi Kehinde Miracle and Oduola Mujeeb Koyejo*

Chemical Engineering Department, University of Port Harcourt, Nigeria

Received Date: 05 March 2021
Revised Date: 10 April 2021
Accepted Date: 25 April 2021

Abstract *- This paper offers a comprehensive evaluation aimed at predicting the water content outcome of natural gas systems using artificial intelligence. In this study, an artificial intelligence model- a three layer artificial neural network model- has been developed using past gas dehydration process data to predict the water content outcomes in natural gas dehydration systems. The water content outcomes are Class 0 which represent data points that meet the water content specification of 7 lb/MMscf or Class 1, which do not. The input features of the model are temperature of the reboiler in ºF, stripping gas flow rate in scf/gal triethylene glycol (TEG), number of equilibrium stages in the contactor, and TEG circulation rate in gal TEG/lb H_2O. An exploratory data analysis was carried out on the training data and the optimum process parameters found are TEG recirculation rates between 3.2 and 3.8 gal TEG/lb H_20, reboiler temperatures between 380ºF and 400ºF, stripping rate of 0 – 3 scf/gal TEG, and two and three equilibrium stage contactors. The model was evaluated against test data and experimental data from literature and F_1 scores of 0.969 and 0.987 were obtained respectively. This showed that the model was able to predict correctly the expected water content outcomes of new gas dehydration data points.*

Keywords — *Natural gas dehydration, water content, artificial intelligence, artificial neural network, modelling.*

I. INTRODUCTION

Natural gas is still an important source of fossil fuel in the energy industry today. Natural gas is obtained from oil and gas reservoirs and is normally saturated with water due to the formation water contained in the reservoir. For the natural gas to be saleable, the water in it must be dehydrated and the water content known. The water content of natural gas has been evaluated by several methods such as experimental phase equilibrium studies, empirical charts, mathematical correlations, and thermodynamic models. Physical experimental phase equilibrium studies of the TEG–water systems have been restricted by low-temperature conditions and boundless dilution regions. They are not as exact and dependable over specific temperatures and have poor extrapolation at these endless dilution regions [1]. Empirical correlation and charts show simplicity of calculations and relative ease of use respectively [2]. The McKetta-Wehe empirical graph is appropriate for sweet natural gas at low temperatures and pressure [3]. However, as temperature, pressure, and acid gas content in the natural gas increases, the charts gets inclined to error as the saturated water content in pure acid gas is higher than the saturated water content in sweet natural gas combinations. The mathematical correlation method also requires a few tuning boundaries and coefficients. A good number of them also stem from empirical charts that are already prone to readability errors. Thermodynamic models require long and troublesome calculations and need a few parameters to be obtained from tedious EOS calculations to get the activity coefficient of water. The models additionally require extraordinary attention to track down a suitable relationship for fitting experimental information and periodically, these strategies have lower precision at anticipating water content at low temperatures [2].

However, with the huge amount of data being produced because of the industrialization and digitization of numerous areas in the industry and with the development of super-fast PCs, it has been possible to take this information and see patterns that give answers to engineering issues. Artificial intelligence algorithms like artificial neural network (ANN) take in data and make internal models by pattern matching with the input and output and updating the weights of the network simultaneously. Innovative research in learning cycles, for example, the back propagation algorithm has made it possible to tackle numerous non-linear complex issues precisely. Dreary, costly, and moderate trials needed for research are being supplanted with artificial intelligent models that gain from historical data. The benefits of ANN over empirical model are that ANN can consistently optimize its weights that fit the model when given more data point, henceforth prompting a more vigorous model [4]. Artificial intelligent models are black box models, which makes them simple to utilize and require negligible numerical correlation intricacy. However, the researcher equipped with broad and relevant information as found in the rich data being inputted into the algorithm makes the artificial intelligence model successful for predictive capacities.

Artificial neural networks model complex non-linear problems; they do so by learning the frequently complex dynamic behaviour of a physical system. Learning is done by adjusting the network's internal parameters or weights

typically in such a way to minimize the squared error between the network's outputs and the desired outputs [5]. All the weights are optimized simultaneously using optimization algorithms like gradient descent or Adam [6].

Artificial intelligence in chemical engineering was first seen in expert systems. Expert systems showed a lot of promise to artificial intelligence in chemical engineering in the 80's. Expert problem-solving typically involves large amounts of specialized knowledge, called domain knowledge, often in the form of rules of thumb, called heuristics, typically learned and refined over years of problem-solving experience. The amount of knowledge manipulated is often vast, and the expert system rapidly narrows down the search by recognizing patterns and by using the appropriate heuristics [7].

The works of Banares-Alcantara et al. [8], [9] showed the early researches done on expert systems on physical property predictions, catalyst selection & malfunction diagnosis respectively. Chemical engineering researchers in the 80's adapted AI theories into chemical engineering problems, mostly process systems engineering problems, notable of which is Basila et al.'s work on chemical reactor control [10] and Takagi et al [11]. Very early works of Kramer [12] also used expert systems and Hoskins et al. [13] and several others used artificial network to model fault diagnosis in chemical plants [14], [15] – [17].

Early neural networks research such as Thompson and Kramer [18] and Bakshi and Stephanopoulos [19] showed the usefulness of artificial neural network in solving non-linear problems in chemical engineering. Elgibaly & Elkamel [4] developed several artificial neural networks (ANN) for the prediction of hydrate formation pressure (output) of various pure gases, gas mixtures and various inhibitors using hydrate phase equilibrium data. Gas gravity and temperature were inputs in one model while varying gas compositions, inhibitors and temperature were inputs in the other models.

The ANN gravity models were compared with empirical correlations based on gas gravity and achieved better accuracy while the ANN composition models were compared with statistical thermodynamic models. Katare et al. [20] did early work on intelligent system for reaction kinetic modelling and catalyst. Efforts were made to work on hybrid system researches, and these have been reported by Sundaram et al. [21]. AI can also be seen to predict the rate of corrosion damage in carbon steel pipes [22]. With the increase in automation and integrated process controls and increased competitive nature of the manufacturing industry, the use of artificial intelligence to solve some chemical engineering problems is a wise decision.

Architecturally, a neural network consists of nodes or neurons or perceptrons in a layer that have weights attached to them and are connected from the input layer through the hidden layers to the output layer through an activation function. The accuracy of model representation is dependent directly on the architecture of the neural network [2]. Neural networks have always existed in the 80's but funding and research decreased due to the high computation cost and handicapped technology [7].

A simple neural network takes in multiple inputs in a layer called input layer and creates an internal or hidden layer using a non-linear activation function, a , like sigmoid function, $\sigma(x) = \frac{1}{1+e^{-x}}$, of the input layer values and weights. Values from this hidden layer now map out to a last layer called output layer where the activated values in the hidden layer serve as its own input values. This process is called forward propagation. Having too little neurons in the hidden layer will lead to a model with low accuracy while having too many neurons will lead to over complication of the model, overfitting, poor generalization, and more computation time.

When the final hypothesis is obtained, the cost function for fitting the weights of the neural network is calculated. The cost function is minimized by doing the computation of the error values using the back propagation method that is, the gradient of the error values are calculated from the back of the neural network structure [23]. The gradient of the error values of the output layer is calculated before the gradient of the error values of the hidden layers.

LeCun et al. [24] have indicated the back propagation of errors procedure to compute the gradient of an objective function with respect to the weights of a multilayer stack of modules is a practical application of the chain rule for derivatives. The gradients of the input layer weights are obtained from the gradients of the output layer weights backwardly. The back propagation algorithm is a very effective method to adjust the weights of a neural network till the calculated hypothesis matches the desired output value.

Due to the fact that the sigmoid function always maps its inputs to either a 0 or 1 output, neural networks are very useful for complex classification type problems where it is either something is or is not. As with logistic regression, when a neural network's hypothesis gives a value of 0.9 for example, it means that the hypothesis has a 90% chance of belonging to output class 1. Many applications of deep learning use feed forward neural network.

The following works have used one form of artificial intelligence model to estimate water content in natural gas. Amir & Dominique [2] developed an artificial neural network using a three layer feed-forward neural network and a modified Levenberg-Marquardt algorithm for estimating the water content of natural gases in equilibrium with liquid water and gas hydrates at low temperatures using experimental data. The output, logarithm of the water content was a function of the inputs, inverse of temperature and pressure. The architecture of the model was one input layer with two neurons, one hidden layer with six neurons and one output layer with one neuron. An average absolute deviation value of 4.4% was obtained from the experimental and predicted values when compared with independent experimental data and the results of past predictive approaches.

Ahmadi & Bahadori [25] used intelligent systems in artificial neural network least square support vector machine (LSSVM) coupled with genetic algorithm to estimate the water dew point(output vector) of a natural gas stream at different TEG concentrations and contactor temperature (input vectors). The R2 value and mean square error of the model was calculated.

Ghiasi et al. [26] developed two mathematical-based models to estimate the water content of natural gas dried by solid calcium chloride dehydration units. The first was a simple empirical correlation where water content was a function of temperature and pressure. The second was a three layer neural network with temperature and pressure as the input. Both models were compared with reported literature and were quite accurate with average absolute deviation of less than 0.2%.

Tatar et al. [27] used radial basis function neural network, LSSVM to predict the water content in a dehydration system using solid dehydrator CaCl2 for both fresh charging and before recharging conditions. The input parameters were temperature and pressure of the dehydrator and MATLAB was used for the computation. The developed model was compared with previously proposed intelligent models and classic correlations.

II. METHODOLOGY

This part focuses on the methodology employed in building a model that predicts the water content outcomes in natural gas dehydration systems using artificial intelligence. In the dehydration of natural gas, sales gas suppliers have to meet up with a water content specification of 7 pounds of water per million standard cubic feet of gas (7 lb/MMscf) or lower. When there is any deviation from this, the gas is tagged off-spec and a lower monetary value is placed on it. Therefore, the two water content outcomes are gas that meets the water content specification and gas that do not meet the water content specification.

Exploratory Data Analysis (EDA)

An exploratory data analysis was carried out on the training data used in building the artificial neural network model. The EDA carried out show the effects of varying the dehydration process parameters on the dehydration process parameters and water content outcome and invariably the expected water content outcomes of new dehydration process parameters. It also shows the optimum value for each dehydration process parameters to ensure the water content specification is met. The EDA was carried out by making a visualisation plot of all the input features, and their relationship with each other and the output, water content outcome.

Source and Nature of Data

The source of the training data used in this work is quantitative secondary data from the work of Hernandez-Valencia et al., [28]. They designed and optimized a typical TEG dehydration unit with PROSIM software by varying the following process parameters; temperature of the reboiler in ºF, stripping gas flow rate in scf/gal TEG, number of equilibrium stages in the contactor, and TEG circulation rate in gal TEG/lb H_2O. The residual water content of the sales gas was calculated for each variation. The base set of dehydration process parameters used to get the data in their work is suitable for this work and is shown in Table 1. The data from the work of Hernandez-Valencia et al., [28] was comprehensive enough to serve as training data for this study.

Table 1: Base Dehydration Unit Process Parameters [28]

Inlet gas temperature	90ºF
Inlet gas pressure	500 psia
Inlet gas composition:	
Methane	85.1 mol %
Ethane	8.5 mol %
Propane	3.8 mol%
n-Butane	1.9 mol %
n-Pentane	0.7 mol%
Lean glycol temperature	90ºF
Rich glycol flash pressure	65 psia
Regenerator pressure	1 atm
Equilibrium trays in contactor	2

Data Manipulation

This work is a classification type problem and classification type problems require the output training set data be labelled either 0 or 1. Using the water content specification limit of 7 lb/MMscf, the discrete water content values in the work of Hernandez-Valencia et al. [28] were turned into categorical data and classified as either 0 or 1. Class 1 represents data points that do not meet the water content specification(water content numbers greater than 7 lb /MMscf) and Class 0 represents data points that meet the water content specification (water content values of 7 lb /MMscf and below). After data manipulation, the discrete water content values became qualitative as they are now represented as either 0 or 1. Table 2 shows the new characteristics of the data. 2020 data points were obtained from Hernandez-Valencia et al. [28], of which 985 data points are class 1 and 1035 data points are class 0. Table 3 shows a sample of the 2020 data collected.

Input Features and Output of the Model

The features (input) chosen to be used in building this model are temperature of the reboiler in ºF, stripping gas flow rate in scf/gal TEG, number of equilibrium stages in the contactor, and TEG circulation rate in gal TEG/lb H_2O. These four input features were chosen because they are the critical parameters important for the optimization of a gas dehydration system in order to get the most optimal water content outcome [29]. The target (output) is the water content outcome usually in lb H20/MMscf but will be represented as 0 or 1 because this is a binary classification problem. These features were chosen because in order to design an optimal dehydration system,

these are the most important variables that affect the water dehydration efficiency [28]. These input features and output, water content outcome make up the training data.

Data Preparation

Using the Pandas library in Python programming language, the data set in Table 3 and the URL stated below, which contains 2020 data points, was vectorized and the input features and output were specified. The entire input features, TEG recirculation rate, number of equilibrium stages, temperature of the reboiler, and stripping gas flow rate, are represented as X and the labelled output, water content outcome is represented as y in this model. The data was split randomly into three parts using the 60/20/20 ratio. Each part was set apart for training, validating, and testing the model respectively. All the input features of the training data were standardized by subtracting the mean of each input feature from each input feature's data point and dividing by the standard deviation. Standardization of input features is important for the model to run faster. The standardized data was fed into the model algorithm.

Life Cycle of the ANN Model Building

The following steps were employed in building the ANN model.

Model Definition

The model architecture chosen for this artificial neural network (Fig. 1) consists of three layers, these are the input layer, hidden layer and output layer. One hidden layer is chosen because it is sufficient for a large majority of chemical engineering problems [26]. Using Python programming language, the Sequential model in the TensorFlow library is the choice of model as this creates a stack of layers that map the input layer to the output layer linearly. The input layer will contain four nodes which represents the four standardized input features of the model. The input layer of the model is a 4 x 2020 matrix, representing the four input features and 2020 data points. The hidden layer will contain seven nodes, which contain the hidden layer activation values. Seven was chosen in accordance with a rule of thumb on the size of hidden layer which says the hidden layer should never be more than twice as large as the input layer [30].

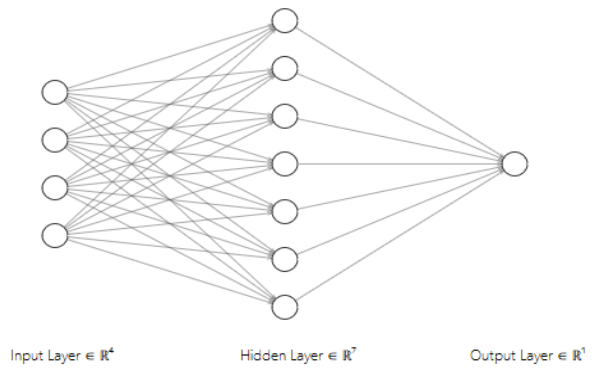

Figure 1. Model architecture

The hidden layer is a 7 x 2020 matrix, representing the seven hidden layer nodes and 2020 data points. The output layer has one node, the target variable water content outcome, represented as 0 or 1 as this is a classification problem. The activation function used for the hidden layer is rectified linear unit because it is computationally efficient (Equation 1 and 2). The sigmoid activation transfer function is used for the output layer which computes the final hypothesis (Equation 3 and 4). The output layer is a 1 x 2020 matrix, representing the output layer single node and 2020 data points. As discussed in chapter 2, since the activation function for the final output of the model, a_{Oi} , is a sigmoid function, the output layer single node a_{Oi} which is the prediction of the model outputs values between 0 and 1. And for any value of a_{Oi} that is obtained, it means the probability of the water content outcome to belong to class 1 is a_{Oi}. A decision boundary of 0.5 is employed, so for any a_{Oi} value less than 0.5, it means the water content outcome of that set of dehydration parameters is class 0. The model contains 35 weights; 28 (7x4 matrix) weights (θ_1 to θ_{28}) between the input layer and the hidden layer and seven (1x7 matrix) weights ($\theta_{29}$to θ_{35}) between the hidden layer and the output layer. The weights of the model are initialized randomly with the He initialization method [31], [32].

Table 2: Training Data Process Parameters [13]

Process parameters	Data range	Data type
Reboiler temperature (°F)	360, 380,400	Ordinal categorical data
Number of equilibrium stages	1, 2, 3	Ordinal categorical data
Stripping rate (scf/gal TEG)	0, 1,3,6	Ordinal categorical data
TEG recirculation rate (gal TEG/lb H_2O)	0.6 to 8.3	Numerical continuous data
Output		
Water content outcomes	0,1	Ordinal categorical data

Table 3: Sample of Training Data Collected [13]

TEG recirculation rate	Equilibrium stages	Reboiler temperature	Stripping rate	Water content outcome
0.600	3	400	0	1
1.200	3	400	0	1
1.400	3	400	0	1
1.413	3	400	0	1
6.118	2	400	6	0
2.422	3	400	0	0
1.452	3	400	0	0
1.465	3	400	0	0
1.478	3	400	0	0
3.700	2	400	0	0

3.714	2	400	0	0
3.729	2	400	0	0
3.743	2	400	0	0
3.757	2	400	0	0
6.571	1	360	0	1
6.707	1	360	0	1
6.843	1	360	0	1
6.979	1	360	0	1
7.114	1	360	0	1
1.700	1	380	0	1
2.000	2	400	3	0
4.886	1	380	0	1
5.007	1	380	0	1
3.057	1	400	0	1
1.771	2	400	1	1
3.071	1	400	0	1

$$z_{Hi} = \theta_{Hi} x_{Ii} \quad (1)$$

$$f(z_{Hi}) = a(z_{Hi}) = max(0, z_{Hi}) = a_{Hi} \quad (2)$$

$$z_{Oi} = \theta_{Oi} a_{Hi} \quad (3)$$

$$f(z_{Oi}) = g(z_{Oi}) = \frac{1}{1 + e^{-z_{Oi}}} = a_{Oi} \quad (4)$$

Where:

x= input features

θ= weights of the model

z= weights and input summation

a= activation value

i= 1 to n, where n = number of nodes in the layer the i is attached to

I= Input layer

H= Hidden layer

O= Output layer

a_{Oi}=final output of the model

Model Compilation

The model was compiled by specifying the loss function, the optimization algorithm for updating the weights of the model, and the metrics for evaluating the model. The loss function will be calculated using the labeled output y of the training data and the final hypothesis/output of the model a_{Oi} with the binary cross-entropy/log loss function as the output has only two classes. The derivatives of the cost function will be calculated using back propagation. The weights of the model will be updated with Adam optimizer until the difference between the labeled output y of the training data and the final hypothesis/output of the model a_{Oi} is very low. Adam optimizer was chosen because it is the most efficient optimizer for solving deep learning problems [6]. The metric chosen is accuracy.

Model Fitting

The model was then fitted with the training data using epochs of 150 and a batch size of 32. An epoch of 150 means the model went through the entire training data 150 times, with batch sizes of 32 data points per batch. Fitting the model produced outputs, model loss and model accuracy, for the training and validation data as specified in the model compilation cycle. Figure 1 shows the code used for the model definition, compilation, and fitting. The full dataset and code can be found in this URL https://github.com/miracleolayemi/Prediction-of-water-content-outcomes-in-gas-dehydration-systems-using-AI.

Part of the code is given below:

```
from pandas import read_csv
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder,
StandardScaler
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
from sklearn.metrics import confusion_matrix, f1_score,
recall_score,precision_score
np.random.seed(42)
%matplotlib inline
```

```
#Predicting the test set values
y_predict=model.predict_classes(X_test)

#Making the confusion_matrix
cm=confusion_matrix(y_test,y_predict)
print('CM:',cm)
pscore=precision_score(y_test,y_predict)
print('PS:',pscore)
rscore=recall_score(y_test,y_predict)
print('RS:',rscore)
fscore=f1_score(y_test,y_predict)
print('F1S:',fscore)
```

```
#Libraries imported
import numpy as np
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
np.random.seed(42)

#Model definition
n_features=X_train.shape[1]
model=Sequential()
model.add(Dense(7,activation='relu',kernel_initializer='he_normal',input_shape=(n_features,)))
model.add(Dense(1,activation='sigmoid'))

#Model compilation
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])

#Model fitting
history= model.fit(X_train,y_train,epochs=150,batch_size=32, validation_split=0.2, verbose=2)
```

Figure 2. Code used for the model definition, compilation, and fitting

III. RESULTS AND DISCUSSION

Optimum Value for Dehydration Process Parameters

From the EDA, the optimum value for each dehydration process parameters to ensure the water content specification is met has therefore been seen to be TEG recirculation rates between 3.2 and 3.8 gal TEG/lb H20 and above, reboiler temperatures between 380ºF and 400ºF, and two and three equilibrium stage contactors only. Specifically, the optimum values of input features for number of equilibrium stage, reboiler temperature and TEG recirculation rates respectively are (2,400,3.6) or (3, 380,3.8) or (3,400,3.2).

Accuracy

Figure 3 shows the model's training and validation data accuracy per epoch. It can be observed that the accuracy increases for both training and validation datasets as the model learns from the data. It can also be seen that the training accuracy is higher than the validation accuracy, showing that the model is not overfitting. The ANN's model final training and validation data accuracy were 97.3% and 96.3% respectively. This shows that the model got 97.3% of its water content outcome prediction right for the training data and 96.3% for the validation data. Also, the accuracy of the model when evaluated with the test data is 96.8%. This shows that the model got 96.8% of its water content outcome prediction right. This means that for every new prediction for an unseen dehydration data set, the model is poised to get the prediction of the water content outcome right 96.8% of the time.

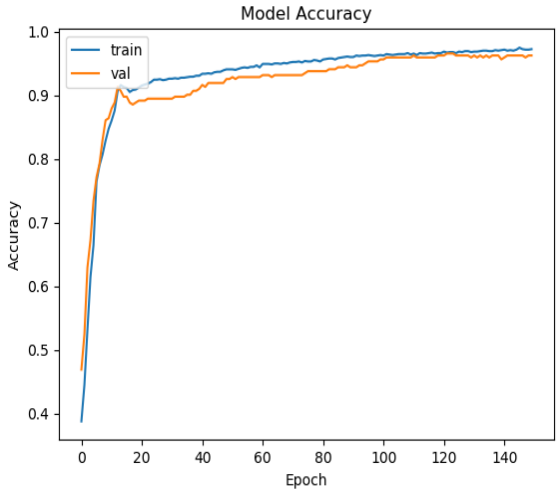

Figure 3. Model accuracy for training and validation dataset

Loss

Figure 4 shows the model's training and validation loss per epoch. It can be observed that the loss reduces as the model learns from the data. It can also be seen that the training loss is lower than the validation loss, showing that the model is not overfitting. The ANN's model final training and validation data loss was 0.084 and 0.097 respectively. Also, the loss of the model when evaluated with the test data set is 0.076. The loss function represents how far apart the predicted water content outcome is from the water content outcome. The loss is close to zero, and it shows the model performed well.

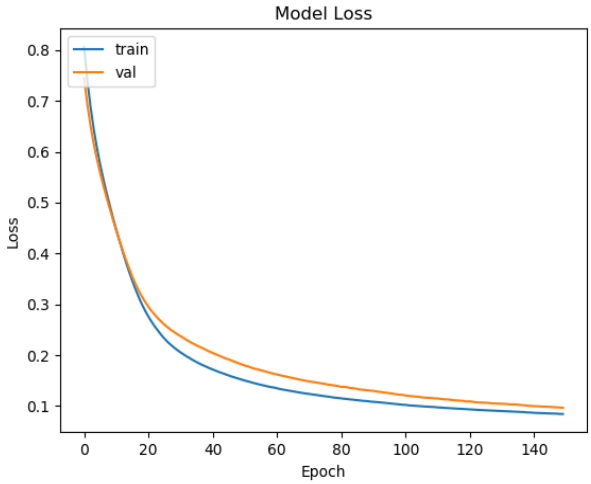

Figure 4. Model loss for training and validation dataset

F_1 Score

The F1 score of the model when evaluated with test data was obtained. The F1 score of the model is 0.969. This is the most important metric for the model. This puts the precision and recall score into consideration and a F1 score close to 1 is very efficient.

Evaluation with Experimental Data Result

The ANN model was evaluated with dehydration process data from Arubi & Duru [14]. Table 4 shows the summary of the ANN result when evaluated with the test data and experimental data. It can be seen that the accuracy and F1 score are high showing good generalization capability of the model.

Result of new predictions

From the result of the EDA, data points that meet water content specifications are TEG recirculation rates between 3.2 and 3.8 gal TEG/lb H20 and above, reboiler temperatures between 380ºF and 400ºF, and two and three equilibrium stage contactors and stripping gas rate value of 3 scf/gal TEG. Therefore, using the following unseen input feature values for TEG recirculation rate, number of equilibrium stage, reboiler temperature and stripping rate respectively, (2.9, 1, 360, 0) for data points that do not meet water content specification, the model predicted the final activation output and because the final activation output was greater than the decision boundary of 0.5, the model successfully predicted class 1. Using these unseen (3.2, 2, 400, 3) input features for data points that meet the water content specification, the model predicted the final activation output and because the final activation output was less than the decision boundary of 0.5, the model predicted successfully class 0.

IV. CONCLUSIONS

The ANN model was successfully built and had an accuracy of 96.8%, loss of 0.076, and F1 score of 0.969 when evaluated with test data. The ANN model had an accuracy of 98.1%, loss of 0.088, and F1 score of 0.987 when evaluated with data from Arubi & Duru [14]. The ANN model was able to predict correctly the expected water content outcomes of new dehydration process parameters for both classes.

Table 4: Summary of ANN Result with Different Evaluation Data

Data used	Data points	Accuracy	Loss	F_1 score
Test data	404	96.8	0.076	0.969
Arubi & Duru [14]	104	98.1	0.088	0.987

REFERENCES

[1] Twu, C. H., Tassone, V., Sim, W. D., & Watanasiri, S., Advanced equation of state method for modeling TEG–water for glycol gas dehydration. Fluid Phase Equil. (2005) 228–229, 213–221. https://doi.org/10.1016/j.fluid.2004.09.031

[2] Amir, H. M., & Dominique, R., Use of artificial neural networks for estimating water content of natural gases. Ind. Eng. Chem. Res, 46(0) (2007) 1431–1438.

[3] McKetta, J. J., & Wehe, A. H., Petroleum refiner (Hydrocarbon Processing), 37(8)(1958) 153.

[4] Elgibaly, A. A., & Elkamel, A. M., A new correlation for predicting hydrate formation conditions for various gas mixtures and inhibitors. Fluid Phase Equilibria, 152(1998) 23–42. https://doi.org/10.1016/S0378-3812(98)00368-9

[5] Psichogios, D. C., & Ungar, L. H., Hybrid neural network-first principles approach to process modeling. AIChE Journal, 38(10)(1992) 1499–1511. https://doi.org/10.1002/aic.690381003

[6] Kingma, D. P., & Ba, J., Adam: A method for stochastic optimization [paper presentation]. International Conference for Learning Representations (ICLR), San Diego, USA. arXiv:1412.6980 (2015).

[7] Venkatasubramanian, V., The promise of artificial intelligence in chemical engineering: Is it here, finally? AIChE Journal, 65(2)(2018) 479–482. https://doi.org/10.1002/aic.16489

[8] Banares-Alcantara, R., Westerberg, A. W., Ko, E. I., & Rychener, M. D., Decade-A hybrid expert system for catalyst selection: Expert system consideration. Comput. Chem. Eng., 11(3)(1987) 265–277. https://doi.org/10.1016/0098-1354(87)85008-1

[9] Banares-Alcantara, R., Westerberg, A. W., Ko, E.I., & Rychener, M. D., Development of an expert system for physical property predictions. Comput. Chem. Eng., 9(2)(1985) 127–142. https://doi.org/10.1016/0098-1354(85)85003-1

[10] Basila JR, M. R., Stefanek, G., & Cinar, A., A model-object based supervisory expert system for fault tolerant chemical reactor control. Comput. Chem. Eng., 14(15)(1990) 551–560. https://doi.org/10.1016/0098-1354(90)87026-L

[11] Takagi T, Sugeno M. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics. 1985;15:116-132

[12] Kramer, M. A., Malfunction diagnosis using quantitative models with non-boolean reasoning in expert systems. AIChE Journal, 33(1)(1987) 130–140. https://doi.org/10.1002/aic.690330115

[13] Hoskins, J. C., Kaliyur. K. M., & Himmelblau, D. M., Fault diagnosis in complex chemical plants using artificial neural network. AIChE Journal, 37(1)(1991) 137–141. https://doi.org/10.1002/aic.690370112

[14] Hajjar Z, Kazemeini M, Rashidi A, Tayyebi S. Artificial intelligence techniques for modeling and optimization of the HDS process over a new graphene based catalyst. Phosphorus, Sulfur, and Silicon and the Related Elements. (2016) 191:1256-1261

[15] Hajjar Z, Khodadadi A, Mortazavi Y, Tayyebi S, Soltanali S. Artificial intelligence modeling of DME conversion to gasoline and light olefins over modified nano ZSM-5 catalysts.Fuel. (2016)179:79-86

[16] Gibbs MS, Morgan N, Maier HR, Dandy GC, Nixon JB, Holmes M. Investigation into the relationship between chlorine decay and water distribution parameters using data driven methods. Mathematical and Computer Modelling. (2006) 44:485-498

[17] Tayyebi S, Shahrokhi M, Boozarjomehry RB. Fault diagnosis in a yeast fermentation bioreactor by genetic fuzzy system. Iranian journal of chemistry and chemical engineering.2010;29:61-72

[18] Thompson, M. L., & Kramer, M. A., Modeling chemical processes using prior knowledge and neural networks. AIChE Journal, 40(8)(1994) 1328–1340. https://doi.org/10.1002/aic.690400806

[19] Bakshi, B. R., & Stephanopoulos, G., Wave-Net: A multiresolution, hierarchical neural network with localized learning. AIChE Journal, 39(1)(1993) 57–81. https://doi.org/10.1002/aic.690390108

[20] Katare, S., Caruthers, J. M., Delgass, W. N. & Venkatasubramanian, V., An intelligent system for reaction kinetic modeling and catalyst design. Ind. Eng. Chem. Res., 43(14)(2004) 3484–3512. https://doi.org/10.1021/ie034067h

[21] Sundaram, A., Ghosh, P., Caruthers, J. M., &Venkatasubramanian, V., Design of fuel additives using neural networks and evolutionary algorithms. AIChE Journal, 47(6)(2004) 1387–1406. https: //doi.org/10.1002/aic.690470615

[22] Salaheddine, K., Ghalib, L., Si Ali, B., & Daud, W., Developing an artificial neural network model to predict carbon steel's corrosion potential in carbonated solution mixture of amines [Paper presentation].CORCON Conference 2015, Chennai, India.,(2016).

[23] Rumelhart, D. E., Hinton, G. E., & Williams, R. J., Learning representations by back-propagating errors. Nature, 323(1986) 533–536. https://doi.org/10.1038/323533a0

[24] LeCun, Y., Bengio, Y., & Hinton, H., Deep learning. Nature, 521(2015), 436–444. https://doi.org/10.1038/nature14539 (2015).

[25] Ahmadi, M. A., & Bahadori, A., Prediction performance of natural gas dehydration units for water removal efficiency using least square support vector machine. International Journal of Ambient Energy, 37(5)(2015) 486–494. https://doi.org/10.1080/01430750.2015.1004105

[26] Ghiasi, M. M., Bahadori, A., & Zendehboudi, S., Estimation of the water content of natural gas dried by solid calcium chloride dehydrator units. Fuel, 117(2014)(2013) 33–42. https://doi.org/10.1016/j.fuel.2013.09.086

[27] Tatar, A., Nasery, S., Bahadori, M., Bahadori, A., Bahadori, M., Barati-Harooni, A., & Najafi-Marghmaleki, A., Prediction of water removal rate in a natural gas dehydration system using radial basis function neural network. Petroleum Science and Technology, 34(10)(2016) 951–960. https://doi.org/10.1080/10916466.2016.1166131

[28] Hernandez-Valencia, V. N., Hlavinka, M. W., & Bullin, J. A., Design glycol units for maximum efficiency [Paper presentation]. Gas Processors Association (GPA) Annual Convention 1992, Oklahoma, USA., (1992).

[29] Arubi, I. M., & Duru, U. I., Optimizing glycol dehydration system for maximum efficiency: A case study of a gas plant in Nigeria. Society of Petroleum Engineers, SPE 113781., (2008);

[30] Berry, M. J., & Linoff, G. S., Data mining techniques: For marketing, sales, and customer support. John Wiley & Sons, Inc., (1997).

[31] He, K., Zhang, X., Ren, S. & Sun, J., Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. arXiv:1502.01852 [cs.CV]., (2015).

[32] Srinivasan Suresh., Prediction of Average Annual Daily Traffic Using Machine Learning Methods., SSRG International Journal of Computer Science and Engineering 6.11(2019) 51-54.