Quest Journals
Journal of Software Engineering and Simulation
Volume 10 ~ Issue 6 (2024) pp: 29-39
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

An Artificial Neural Network Model for Small and Medium-Size Data Analysis UsingLevenberg-Marquardt Optimization Algorithm

TIJANI, M. A.[1]NWIABU, N. D[2]& BENNETT, E. O.[3]
Department of Computer Science, Faculty of Sciences, Rivers State University, Nkpolu-Oroworukwo, Rivers State, Nigeria

Abstract
Artificial Neaural Network (ANN) is a self-adaptive and self-learning functions suitable for complex nonlinear information processing. Disadvantages of ANN include its "black box" nature, proneness to overfitting, and greater computational burden which include speed and use of lots of data. Many methods such as the error backpropagation (EBP) algorithm and the Gauss–Newton (GN) algorithm have been developed for neural-networks training to overcome the challenges. The EBP algorithm is still widely used but it is an inefficient algorithm because of its slow convergence. The slow convergence of EBP is improved by the Gauss–Newton algorithm using second-order derivatives of the error function to evaluate the curvature of the error surface to find proper step sizes for each direction and converge very fast. This research work uses the Levenberg–Marquardt (LM) algorithm to train an ANN model. The LM method blends the steepest descent method (EBP) and the Gauss–Newton algorithm. It inherits the speed advantage of the Gauss–Newton algorithm and the stability of the steepest descent (GN) method. It's more robust than the Gauss–Newton algorithm, because in many cases it can converge well even if the error surface is much more complex than the quadratic situation. Additionally, the Levenberg–Marquardt algorithm is suitable for training small and medium-sized problems which is not recommended when using EBP and GN. In this work, a small-sized chemical data was used to train an ANN model using the Levenberg–Marquardt algorithm without extrapolation and the prediction accuracy was 98%.

Keywords: *Artificial Neural Network, Levenberg-Marquardt Optimization, Algorithm, Small Medium-Size*

R*eceived 14 June, 2024;* ***R****evised 25 June, 2024;* ***A****ccepted 28 June, 2024 © The author(s) 2024.*
P*ublished with open access at* ***www.questjournals.org***

I. Introduction

Artificial Neural Networks (ANNs) mimic and abstract the neuronal network of the human brain, create a mathematical model, and interpret data by changing the connections between a large number of model nodes (Gao et al., 2019).(Yang & Huang, 2002). It works best in complicated nonlinear information processing systems and has self-adaptive and self-learning features. In a neural network, every node has a state variable. Each node has a threshold θj and a nonlinear transformation function f. The nodes i and j are linked by the connection weight coefficient wji. Artificial neural networks may be said to have originated with the McCulloch and Pitts (1943) neuron biology model (M-P model) and the Hebb rule (1949) by Hebb. Theoretically, artificial neural networks are based on the seminal works of Hopfield (1988), Kauffman (2000), LeCun (2015a, 2015b), and Hinton (2006a, 2006b). They are essential tools in computer science and are widely used in the modeling of brain and gene networks. The H-H equation, a well-known nonlinear dynamic differential equation, was developed by Hodgkin and Huxley in 1952.

The nonlinear processes that take place in the neuron membrane, such as various stability issues, self-excited oscillation, and chaos, may be explained by this equation. The first artificial neural network with learning capabilities, the Perceptron model, was published by Rosenblatt in 1958. Widrow in 1960 introduced the adaptive (Adaline) linear component model, which is a continuous self-adapt to the linear cell neural network model. In summary, artificial neural networks have reached the commercial application stage thanks to the combined efforts of several scientists. Through learning, an artificial neural network may approximate any nonlinear mapping, and as it is not constrained by the nonlinear model, it can be used to identify and model nonlinear systems (Qi &Zheng, 2021). Artificial neural networks may be classified as feedforward networks or

feedback networks based on the distinct roles they play (Zhang, 2021). The feedforward neural network's primary goals are recognition and learning.
It is crucial to remember that the Levenberg-Marquardt (LM) method offers a numerical solution for the nonlinear function minimization issue. Both its convergence and its speed are steady. The Levenberg-Marquardt technique works well for training small- and medium-sized issues in artificial neural networks. Numerous more techniques have previously been developed for training neural networks. Neural networks are often trained using the steepest descent approach, commonly referred to as the error backpropagation (EBP) algorithm. Even though the EBP algorithm is still often employed today, its sluggish convergence has earned it the reputation of being an inefficient method. The Gauss–Newton technique has the potential to significantly enhance the steepest descent method's sluggish convergence. The Gauss–Newton algorithm can find appropriate step sizes for each direction and converge quickly by using second-order derivatives of the error function to evaluate the curvature of the error surface. In particular, if the error function has a quadratic surface, it can converge directly in the first iteration.

However, this gain only occurs provided the error function's quadratic approximation is appropriate; in other cases, the Gauss–Newton technique would be mostly divergent. The Gauss–Newton algorithm and the steepest descent technique (EBP) are combined to create the Levenberg–Marquardt algorithm. It inherits the steepest descent method's stability and the Gauss–Newton algorithm's speed advantage. Because it may often converge effectively even in situations where the error surface is much more complicated than in a quadratic scenario, it is more resilient than the Gauss–Newton technique. Some laboratory experimental data are small or medium-sized. Training artificial neural networks with small-sized data using the error backpropagation (EBP) algorithm or the Gauss–Newton algorithm produces less accurate results. It is against this background that, this paper seek to answer the following questions. Is feed-forward neural network effective in representing complex systems? To what extent has modified Levenberg-Marquardt optimization algorithm enhance the ANN model? And if Python programming language will promote Small and Medium-Size Data Analysis?

II. Literature Review

The Multilayer Perceptrons Theory was propounded by Miller *et al.*(2011). Among the most popular ANN algorithms is multilayer perceptrons (MLP). Since this technology is made up of several neurons stacked in various layers, the term "multilayer" is employed. Every link between the input and one or more hidden layers—or between two hidden layers—is like a synapse, which is its biological equivalent, and the input data is altered by a predetermined weight. Accordingly, an input layer, two hidden layers, and an output layer make up a three-layer feed-forward network (Miller et al., 2011). Because the data only travels in a forward manner, MLP is also known as feed-forward neural networks. Stated otherwise, a layer's output is only used as an input by the layer after it. Supervised learning is a crucial feature of feed-forward networks (Smits et al., 1994). The training stage is the most important duty in the MLP approach. The goal of the training or learning stage is to minimize the squared errors of prediction (experimental x estimated data) by searching for a set of weight values.

The slowest phase is this one, and a minimum worldwide accomplishment is not guaranteed. Although there are a number of learning methods for MLP, including Levenberg-Marquardt,quasi-Newton, conjugate gradient descent, and others, the back-propagation approach is the most often used. This approach modifies the weight of layer connections based on the output layer (prediction) error values. As a result, Marini (2009) states that this method guarantees minimal (local or global) convergence. Selecting the best architecture is the primary MLP difficulty. The number of layers and the number of hidden unities in each layer have a significant impact on the MLP learning's speed and performance (Widrow& Lehr, 1990). The complexity of the issue that has to be addressed increases in direct proportion to the number of layers in an MLP algorithm. The intricacy of the neural network's ability to recognize patterns increases with the number of hidden layers**.**

Kohonen Neural Network Theory

Zupan et al. proposed the Kohonen neural network theory in 1997. The idea included the use of an unsupervised neural network to carry out a non-linear mapping of a high-dimensionality data space onto a low-dimensional one, often one that is bidimensional. The output 2D-layer neurons' distances and proximities are used to visualize the output data. Stated differently, the SOM method is used to cluster and extend the data set while preserving its original topology. Only the closest neighbors are linked to the output neurons of the SOM. An output neuron represents a comparable pattern, which is represented by the neighborhood. An output neuron's neighborhood is often described as square or hexagonal, meaning that each neuron has four or six closest neighbors, depending on the definition (Zupan et al., 1997). The SOM model is used in the combinatorial creation of cannabis drugs (Honório et al., 2010) and purinergic receptor antagonists (Schneider and Nettekoven, 2003) using square and hexagonal neurons, respectively. The learning method used in the SOM approach makes it a competitive neural network. Due to competitive learning, only the output layer neuron is chosen above the other input neurons if its weight most closely resembles the input pattern. Ultimately, the neighborhood's learning rate is lowered in proportion to the winning output neuron's distance (Kohonen, 2001).

A more sophisticated chemical process, consisting of five typical components (reactor, condenser, gas-liquid separator, desorption tower, and circulating compressor), was simulated by Downs and Vogel (1993) using BP neural network (BPNN). Four reactions occur in total, yielding two products and one byproduct. A 4-dimensional vector is used to represent the 15 known fault numbers, to which disturbances are added. Remove 500 data sets to train the neural network; after that, remove 100 data sets for testing. BPNN demonstrates a high degree of fault identification capacity by training using the Levenberg-Marquardt back-propagation method. Gu (2008) discovered that the process connection in the numerical simulation examples is static and linear. Gu (2008) employed BP neural network and RBF neural network to identify and diagnose problems. The BP network has a positive impact on fault finding.

Hu et al. (2006) simulated the catalytic distillation tower startup process using artificial neural networks. The L-M technique is the artificial neural network's learning algorithm used in this investigation. The initial feed liquid's composition and total moles in the tower are the input parameters; the catalyst distillation's start time is the output parameter. As the amount of training data grows, the artificial neural network's prediction performance rises and its prediction error steadily falls.

Yang et al. (2007) simulated and predicted the ultrasonic-enhanced supercritical flavonoid extraction process from Toonasinensis leaves using a three-layer BP network model. There are minimal inaccuracies in the data obtained when discussing the impacts of several parameters on the extraction rate of total flavonoids, including extraction temperature, pressure, fluid flow, quantity of entrainer, extraction time, and ultrasonic power.

In order to regulate the heat transfer mechanism, Vasičkaninová et al. (2011) used a neural network predictive control (NNPC) structure. The findings demonstrate the efficiency and superiority of NNPC by demonstrating that the heat exchanger's NNPC uses less heating medium when compared to the traditional PID (Proportional-Integral-Derivative) control.

Osuolale et al. (2016) suggested modeling and optimizing distillation tower energy efficiency using neural networks and the second law of thermodynamics. A bootstrap aggregated neural network may improve the model's precision and dependability. The binary systems of benzene-toluene and methanol-water are distilled using Aspen HYSYS, which eventually results in utility consumption reductions of 8.2% and 28.2%, respectively.

An artificial neural network (ANN) model was presented by Longo et al. (2020) to forecast the refrigerant's boiling heat transfer coefficient in a brazed plate heat exchanger (BPHE). The average error (MAPE) of the anticipated value is 4.8%, and the model considers the impact of plate shape, operating circumstances, and refrigerant properties. The ANN model has superior prediction power when compared to the majority of the most sophisticated analysis and calculation tools for internal boiling of BPHE that are accessible in the public literature. Model Predictive Control (MPC) provides a better operational efficiency when compared to classical PID control.

Shin et al. (2020) replaced the current linearization model with an artificial neural network (ANN) model and utilized Aspen HYSYS to simulate the de-propanizer. They take into account every possible operational situation in order to provide a substantial volume of dynamic simulation data, which is then used to train and evaluate artificial neural networks. The findings demonstrate that MPC has more sensitive control performance than PID control by requiring shorter rising and settling times.

Zhu et al. (2020) optimized the burdock solid fermented Ganodermalucidum polysaccharide extraction method using a novel adaptive genetic neural network. An adaptable genetic neural network was built using MATLAB R2016B, and it was trained using test data taken from Ganodermalucidum polysaccharide. The experimental value of the polysaccharide content was fitted using the adaptive evolutionary neural network technique, with the beginning interval of the ownership value threshold and the variable accuracy set to 10-4. The findings demonstrate that the adaptive genetic neural network algorithm has better prediction and optimization skills than the regression analysis approach, and that the 13 sets of predicted values of the genetic neural network model are nearly entirely matched with the experimental ones.

A dependable model known as ANN-ELM, developed by Manssouri et al. (2021), is based on the artificial neural network type known as ELM (Extreme Learning Machine) and is capable of differentiating between normal and abnormal patterns. It is sprayed with toluene/methylcyclohexane to the distillation column. The tower top temperature is the result of all pertinent inputs, which include heating power, preheating power, reflux rate, feedrate, pressure drop, and preheating temperature. The outcomes of training and testing the ELM model on a database of 1000 samples demonstrate the model's excellent prediction accuracy. In the test phase, a low RMSE value (RMSE = 0.0168) was observed when the hidden layer's number of neurons was thirty. The ANN-ELM predictive model may be utilized for online detection and diagnostics and is best suited for normal mode modeling of the changeable operating point of the automated continuous distillation column.
Based on BP neural network, Kang et al. (2021) improved the process of polyvinylidene fluoride/polypropylene

gradient composite filter medium. MATLAB is used to build a feed-forward neural network with the goal value being fiber membrane filtration resistance. The algorithm learns, trains, and tests on sample data until it becomes proficient in computation. Assume three input units: voltage, receiving distance, and injection speed. Configure four hidden units. The output unit is filtering resistance. To construct a network object for pre-feedback training, use the new function. After 400 training cycles, configure the network's parameters and provide feedback on the outcome. Each layer's output value is determined using the Sigmoid excitation function. To get the fitting curve, set aside 70% of the data for training, 15% for correction, and 15% for testing. The voltage of 30 kV, the receiving distance of 16.8 cm, and the flow rate of 1.6 mL/h are the optimal process parameters. The resistance prediction value is 81.25 Pa, indicating strong prediction accuracy, while the BP neural network's relative error is 1.99%.

Su et al. (2021) optimized the light guide's injection molding procedure using a radial basis neural network model. The best Latin hypercube sampling technique is chosen to get the sample, using the light guidance strip of a car front combination lamp as an example. A radial basis (RBF) neural network model was built using five parameters (melt temperature, mold temperature, holding duration, holding pressure, and cooling time) chosen for the input layer and two parameters (minimum volume shrinkage rate and minimum sink mark index) chosen for the output layer. A set of parameters for the ideal injection molding process are obtained using the Insight optimization module, and the actual simulation results are essentially identical with the anticipated ones, which enhances the molding quality.

Zhang et al. (2021) assessed the in vitro antioxidant activity of Tibetan tea polysaccharide (TTP) and optimized its extraction procedure using a general artificial neural network (GA-ANN). The response surface method (RSM) and genetic algorithm-artificial neural network (GA-ANN) were used to optimize the extraction process, with the liquid-to-material ratio, extraction temperature, and extraction duration serving as input factors and the TTP extraction rate as the output parameter. Du et al. (2018) used an improved evolutionary algorithm and an artificial neural network model to predict and optimize the best protoplast manufacturing method for Paecilomycestenuipes. The ideal preparation method discovered via the aforementioned optimization was followed in the creation of the Paecilomycestenuipes protoplasts. In five parallel tests, the average generation of protoplasts was 4.4×107 cells/mL, and the error with the ANN model's anticipated value was 0.23%.

III. System Design

The approach used in this case is the Object-Oriented Design Method (OODM). This method combines the functionality of Object-Oriented Design and Top-Down Design Approach and involves breaking down the entire system into subsystems and modules. The OODM is useful for investigating the structure and usage of the system and utilizes a formal and efficient process for designing and planning the data system. To develop the software, Objected-Oriented Design Approach (OODA) and Top-Down plan are used, with the key elements being to find the objects, organize the objects, and describe how they interact. A use case diagram is drawn out for the system using the Unified Modeling Language. An interaction between a user and the system is represented graphically in a use case diagram.

Method of Data Collection

Secondary Data collection: we adopted Secondary Data Collection because it allows the collection of data without the owner’s approval. The data was already analyzed, made available by the owner on books, magazines, newspaper etc.

The data was collected online from Google Data set Search which represents data collected from series of experimental works on the assessment of the content of sulfur in hydrogen sulfide. The tables below contain some sulfur content assessments in the dataset used for model training.

Data processing: is a step in the data mining and data analysis process that takes new data and transforms it into a format that can be understood and analyzed by computers and machine learning. Feature selection help in reducing the inputs for processing and analysis, or of finding the most meaningful inputs. Data acquisition helps in collecting data from relevant sources before it can be stored, cleaned, preprocessed and used for further mechanisms.

Table 1: Sample dataset of Sulfur content in hydrogen sulfide (Mohammadi&Richon, 2008)

T(K)	P(Mpa)	Experimental value (mass%)
316.3	7.028	1.24
316.3	10.473	1.37
316.3	17.363	1.47
316.3	24.253	1,54
315.3	31.143	1.63
338.7	7.028	1.94
338.7	19.478	2.16
338.7	17.363	2.70

338.7	24.253	3.07
338.7	31.143	3.20
336.5	10.473	2.68
336.5	17.363	4.44
336.5	24.368	5.84
336.5	31.143	6.92
374.8	10.473	2.08
374.8	17.363	4.27
374.8	24.353	5.82
374.8	31.143	7.23
388.2	17.363	3.98
374.8	24.253	5./85
374.8	31.143	7.11

Analysis of the System

The proposed system is an intelligent system that predicts sulfur content in a composition of hydrogen sulfide at certain temperature and pressure using Artificial Neural Network (ANN). To avoid sulfur deposition problems, accurate knowledge of sulfur + sour/acid gas phase behaviour is important. Vapor and liquid phases of sulfur are composed of different molecules. Sulfur may exist as a number of polymeric species ranging up to S_8 in the gas and combines with other gases to produce polysulfides or sulfanes. The amount of each molecule depends both on pressure and temperature. Under atmospheric pressure and at temperatures greater than the fusion temperature, liquid sulfur is composed of about 99% of S_8 and traces of lighter molecules.

Artificial Neural Network/Levenberg–MarquardtModel

An Artificial neural network is a useful approach for correlating a limited quantity of experimental data with required variables in some specific cases where temporarily lack of fundamental understanding of the necessary relationship between the variables under examination, or the analytical or empirical correlation is possibly not conventional to be actually applied

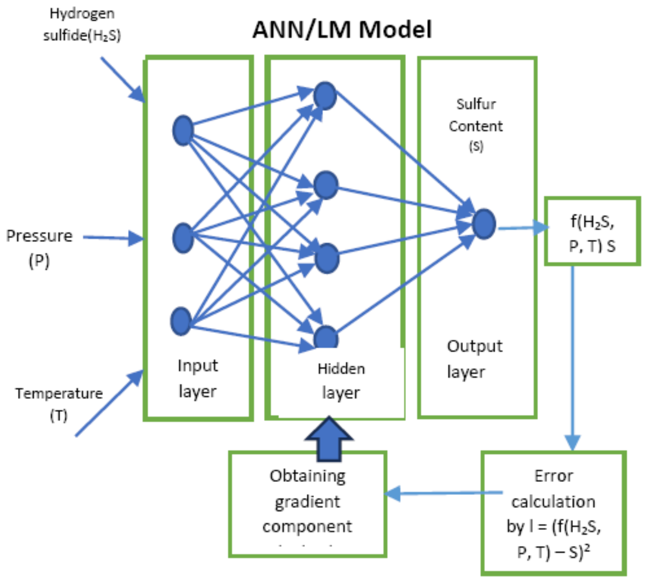

Figure1: Artificial Neural Network/Levenberg–Marquardt Model

An artificial neural network consists of large numbers of computational units called neurons, connected to each other by means of direct weighted communication links as shown in Figure 1. The input layer of the network receives all the input data and introduces scaled data to the network. The data from the input neurons are propagated through the network via weighted interconnections. Every *i* neuron in a *k* layer is connected to every neuron in adjacent layers. The *i* neuron within the hidden *k* layer performs the following tasks: summation of the arriving weighted inputs (input vector Ii = [$\mathrm{I}_{i,I,....}\mathrm{I}_{i,}\mathrm{N}_{k\text{-}I}$]) and propagations of the resulting summation through a non-linear activation function f to the adjacent neurons of the next hidden layer or to the output neuron(s). In this work, the activation function is a sigmoid function:

$$f(x)=\frac{1}{1+e^{-x}} \qquad x\in[0,1] \qquad 1$$

A bias term, *b*, is associated with each interconnection in order to introduce a supplementary degree of freedom. The expression of the weighted sum, *S*. to the i^{th} neuron in the k^{th} layer (*k*≥2) is:

$$S_{k,i} = \sum_{j=I}^{N_{K-I}} \left[\left(w_{k-1,j,i} I_{k-1,j} \right) + b_{k,i} \right] \qquad 2$$

Where *w* is the weight parameter between each neuron-neuron interconnection. Using this simple feed-forward networks with non-linear sigmoid activation functions, the output, 0, of the i neuron within the hidden *k* layer is therefore:

$$O_{k,i} = \frac{1}{1+e-\left(\sum_{j=1}^{N_{k-1}} \left[\left(w_{k-1,j,i}\, I_{k-i,j} \right) + b_{k,i} \right] \right)} = \frac{1}{1+e^{-s_{k,i}}} \qquad 3$$

For the use with the neural network, the input data, *X*, were normalized and centred:

$$X_i{}^I = 0.1 + 0.8\,\frac{X_i{}^I - X_{i,min}}{X_{i,max} - X_{i,min}} \qquad 4$$

Where X*i* is the i^{th} values of the input data fed to the input neuron *i*, and $X_{i,min}$ is the minimal value of the input data fed to the same *i* neuron and inversely $X_{i,max}$ is the maximal value. To achieve a better stability and to have output of the same order of magnitude, the following scaling rule was applied to sulfur content before normalisation:

$$X_{Network} = In\left(X_{exp}\right) \qquad 5$$

During the training, input variables are fed to the network and the difference between the experimental outputs and the calculated outputs is used as a criterion for adjustment of network's synaptic weights. All synaptic weights and biases are first initialised randomly. The network is then trained, its synaptic weights are adjusted by an optimisation algorithm, until it correctly emulates the input/output mapping, by minimizing the average root mean square error. The optimisation method chosen in this work is the Levenberg-Marquardt algorithm (Levenberg, 1944; Marquardt, I963).

Use Case Diagram

Use case diagram is described as a software requirement for a new software program underdeveloped, it specifies the expected behaviour (what) and not the exact method of making it happen (how). Use case once specified, can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective and effective communication of the system behaviour in the user's terms by specifying all externally visible system behaviour.

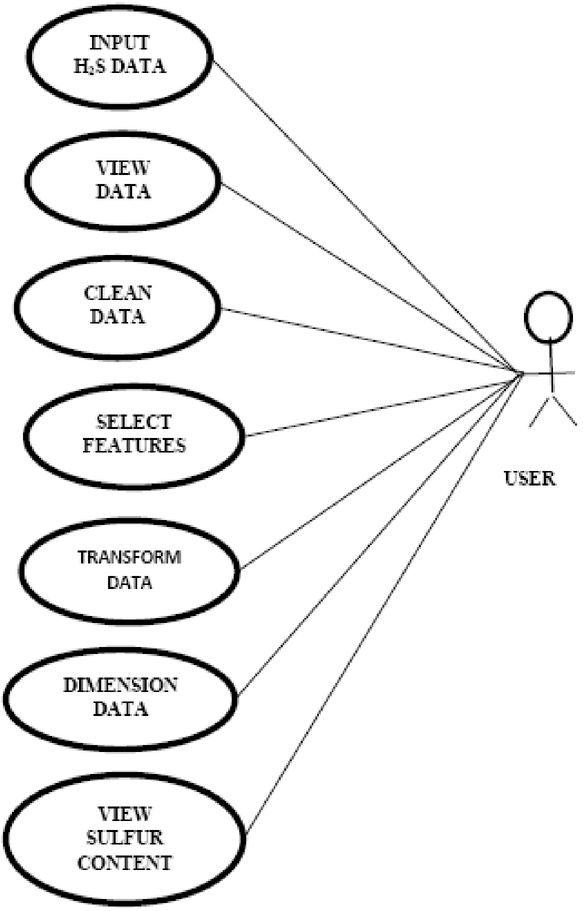

Figure 2: Use Case Diagram

The functionality and requirements of every system is based on its use case diagram. The use case diagram of the proposed system shows the interaction between the actors which are chemists and the system. A detailed description and function of the use case diagram is shown below for clarification.

Table 2: Use Case description and its functionality

Actor	**Use case**	**Description**
User	Input data	The user import dataset into the model for prediction
	Validate results	The user will validate the result by comparing the prediction made by the model to ascertain if the prediction is correct or not.
	View data	
	View result	He user views the result that was predicted
	Collect Dataset	Data are collected continuously while residents perform their routines
		Data wrangling and processing
	Clean data	Identifying and correcting mistakes or errors in the data.
	Select Features	Identify these input variables that are most relevant to the task.
	Transform data	Changing the scale or distribution of variables
		Features Engineering: Deriving new variables from available data
	Dimension data	Dimensionality reduction: Creating compact projections of the data
	Train the Model	The system train the model with fully connected networks. 21 input dimension. 2 hidden layer and 28 output.

Class Diagram

A class diagram describes the structure of a system by showing the system's classes, their attributes operations (or models) and the relationships among objects.

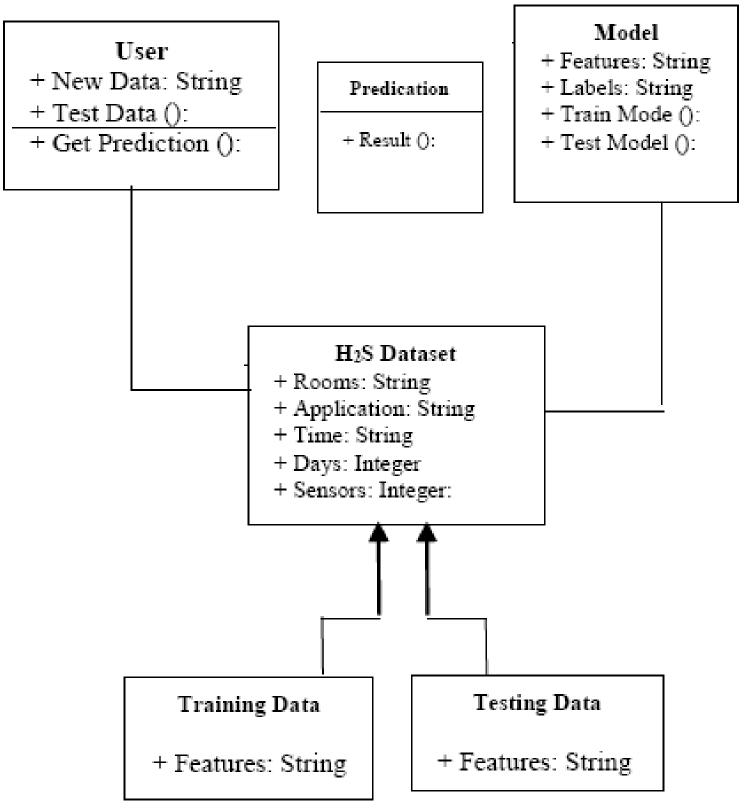

Figure 3: Class Diagram

In the class diagram above, there are 6 classes. The user has new data of hydrogen sulfide which will be tested for prediction, the model is now trained and tested using a dataset. The trained dataset contains all the lectures and labels which will enable the model to be trained and tested for a predictable result thereafter the content of sulfur in hydrogen sulfide compound will display.

IV. Results

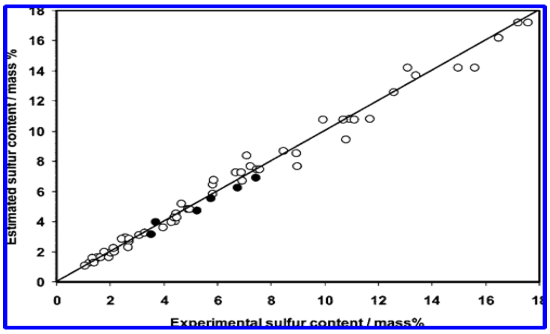

Figure 4: Estimated sulfur content of hydrogen sulfide (mass %) versus corresponding experimental value (mass %): (a) data used for training (and testing); (b) data used for validation.

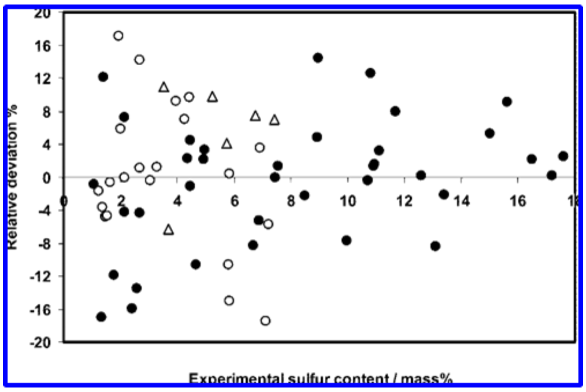

Figure 5: Relative deviation between experimental value (mass%) of sulfur content of hydrogen sulfide (mass %) versus ANN predicted value for sulfur content of hydrogen sulfide (mass%).

AD: Absolute deviation = (experimental value - predicted value)/experimental value. *D*: Relative deviation = (experimental value - predicted value)/exp experimental value.

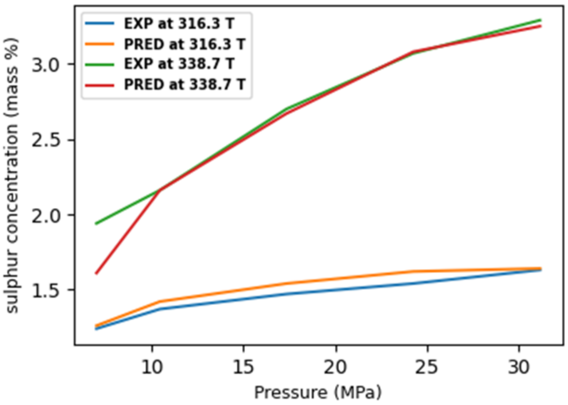

Figure 6:Experimental/Predicted Sulfur Content of Hydrogen Sulfide at 316.3K & 338.7K

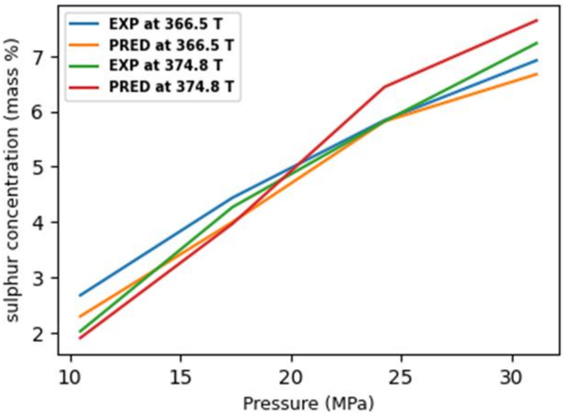

Figure 7:Experimental/Predicted Sulfur Content of Hydrogen Sulfide at 366.5K & 374.8K

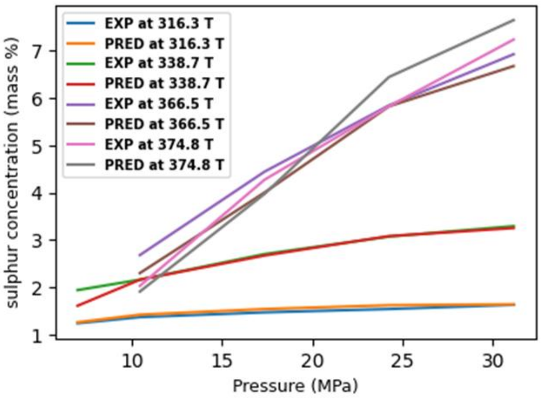

Figure 8:Experimental/Predicted Sulfur Content of Hydrogen Sulfide at 316.3K, 338.7K, 366.5K, & 374.8K

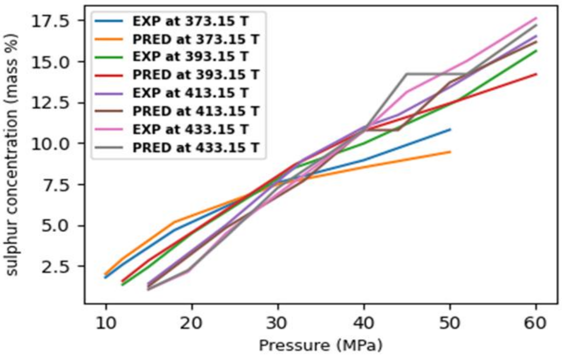

Figure 9:Experimental/Predicted Sulfur Content of Hydrogen Sulfide at 373.15K, 393.15K, 413.15K, & 433.15K

V. Discussion

Table 3.2 presents a sample of the experimental data used for developing the ANN algorithm. The table reports sulfur content of hydrogen sulfide in the temperature range of (316-433) K and pressures up to 60 MPa. An ANN algorithm with one hidden layer was devoted to compute the logarithm of sulfur content (in mass fraction) of H_2S. The input neurons are H_2S, temperature, and pressure, while the output is the sulfur content. The optimum number of neurons in the hidden layer according to both the accuracy of the fit (minimum value of the objective function) and the predictive power of the neural network is found in table 1, and figure 1 to 6, comparing the results of the ANN algorithm with the experimental data.As can be seen, acceptable agreement was achieved and the average absolute deviation among all the experimental and predicted data is 6.1%. The maximum absolute deviation (AD) among all the experimental and predicted data is less than 18%. To better present the deviations, we have compared relative deviations between estimated sulfur content of hydrogen sulfide and corresponding experimental value versus experimental value for sulfur content of hydrogen sulfide in Figure 2. As can be observed, high deviations are observed at low values of sulfur content of hydrogen sulfide. Considering the fact that such measurements, especially at high temperatures and high pressures and low sulfur contents, are indeed not easy and some experimental data may have errors, the AD up to 18% can be regarded acceptable. It should be mentioned that the ANN algorithm introduced in this study was developed using the most reliable experimental data found in the literature on sulfur content of hydrogen sulfide. By generating reliable experimental data on sulfur content of hydrogen sulfide in the future and readjusting the model parameters, more reliable predicted data would be expected.

VI. Conclusion

Artificial Neaural Network (ANN) is a self-adaptive and self-learning functions suitable for complex nonlinear information processing. We developed an ANN model but ANN works well with large datasets. The dataset for this study is small dataset. The study used the Levenberg–Marquardt (LM) algorithm to train the ANN model. The Levenberg–Marquardt (LM) algorithmprovides a numerical solution to the problem of minimizing a nonlinear function. The algorithm blends the steepest descent method (EBP) and the Gauss–Newton algorithm. It inherits the speed advantage of the Gauss–Newton algorithm and the stability of the steepest descent method. It's more robust than the Gauss–Newton algorithm, because in many cases it can converge well even if the error surface is much more complex than the quadratic situation. When a solution is far from a local minimum, the algorithm behaves like a steepest descent method: slow but guaranteed to converge. When a solution is close to a local minimum, it becomes a Gauss-Newton method and exhibits fast convergence suitable for modeling chemical data. The integrated ANN/Levenberg–Marquardt model was implemented using a small chemical data estimating sulfur content of hydrogen sulfide in a temperature range of 316-433 K and pressures up to 60 MPa. The agreements between the experimental and predicted data were generally found acceptable.

VII. Recommendation

An iterative method known as the local minimum algorithm (LM algorithm) finds the local minimum of a multivariate function that is represented as the sum of squares of many real-valued, non-linear functions. The Gauss-Newton technique and the steepest descent are combined to create LM. The algorithm operates like a steepest descent technique when a solution is distant from a local minimum: it is sluggish but will eventually converge. A solution becomes a Gauss-Newton technique and shows quick convergence as it approaches a local minimum. In chemical research, the Levenberg-Marquardt (LM) algorithm should be used as a standard method to handle data-fitting in non-linear least-squares issues.

References

[1]. Bayes, T. (1764). An Essay toward solving a problem in the doctrine of chances. Philosophical Transactions of the Royal Society of London, 53, 370-418.

[2]. Downs, J. J. & Vogel, E. F. (1993). A plant-wide industrial process control problem. Com Chem Engineering, 17(3), 245-255.

[3]. Gao, J., Chakraborty, D., Tembine, H. &Olaleye, O. (2019). Nonparallel Emotional Speech Conversion. INTERSPEECH 2019, Graz, Austria.

[4]. Hebb, D. O. (1949). The Organization of Behavior: A Neuropsychological Theory[M]. Lawrence Erlbaum Associates, New Jersey.

[5]. Hinton, G. E., Osindero, S. &Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18(7), 1527-1554

[6]. Honório, K. M., de Lima, E. F., Quiles, M. G., Romero, R. A. F., Molfetta, F. A., & da, Silva. A. B. F. (2010). Artificial Neural Networks and the Study of the Psychoactivity of Cannabinoid Compounds. Chem. Biol. Drug. Des., 75, 632-640.

[7]. Hopfield, J. J. (1988). Artificial neural networks. IEEE Circuits and Devices Magazine, 4(5), 3-10.

[8]. Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. Proceedings of National Acad Science, 81(10), 3088–3092.

[9]. Hopfield, J. J., Feinstein, D. I. & Palmer, R. G. (1983). Unlearning' has a stabilizing effect in collective memories. Nature, 304(5922), 158-159.

[10]. Hu, H., Wu, H. X. &Xu, S. M. (2006). Predictions Catalytic Distillation Column Start-up Processes Via Artificial Neural Network. Journal of Molecular Catalysis (China), 20(4), 360-362.

[11]. Kang, L., Wang, L. &Gao, X. (2021). Process optimization of polyvinylidenefluoride/poly- propylenegradient composite filter media based on BP neural network. ActaMateriaeCompositaeSinica, 38, 9.

[12]. Kauffman, G. W. &Jurs, P. C. (2000). Prediction of inhibition of the sodium ion-proton antiporter by benzoylguanidine derivatives from molecular structure. Journal of Chemical Information and Modeling, 40(3), 753-761.

[13]. Kohonen, T. (2001). Self Organizing Maps (3 ed.), New York, Springer.

[14]. LeCun, Y. (2015). Deep learning and convolutional neural networks, HOTCHIPS.

[15]. Longo, G. A., Mancin, S. &Righetti, G. (2020). Application of an Artificial Neural Network (ANN) for predicting low-GWP refrigerant boiling heat transfer inside Brazed Plate Heat Exchangers (BPHE). International Journal of Heat and Mass Transfer, 160, 120204.

[16]. Manssouri, I., Boudebbouz, B. &Boudad, B. (2021). Using artificial neural networks of the type extreme learning machine for the modelling and prediction of the temperature in the head the column. Case of a C6H11-CH3 distillation column, Materials Today: Proceedings, 45, Part 8, 7444-7449.

[17]. Marini, F. (2009). Artificial neural networks in foodstuff analyses: Trends and perspectives. A review. Anal. Chim. Acta, 635, 121-131.

[18]. Miller, F. P., Vandome, A. F. &Mc Brewster, J. (2011). Multilayer Perceptron, Alphascript Publishing.

[19]. Osuolale, F. N. & Zhan, J. (2016). Energy efficiency optimisation for distillation column using artificial neural network models. Energy, 106, 562-578.

[20]. Qi, Y. &Zheng, Z. (2021). Neural Network Algorithm and Its Application in Supercritical Extraction Process, Asian Journal of Chemical Sciences, 9(1), 19-28.

[21]. Rosenblatt, F. (1958). The perceptron: Probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386-408.

[22]. Shin, Y., Smith, R. & Hwang, S. (2020). Temperature Prediction and Model Predictive Control (MPC) of a Distillation Column Using an Artificial Neural Network Based Model. Journal of Cleaner Production, 277, 124124.

[23]. Smits, J. R. M., Melssen, W. J., Buydens, L. M. C., &Kateman, G. (1994). Using artificial neural networks for solving chemical problems. Part I. Multi-layer feed-forward networks. Chemom. Intell. Lab., 22, 165-189.

[24]. Su, T., Huang, Y. & Ni, J. (2021). Radial Basis Neural Network -Based Light Guide Strip Injection Molding Process Optimization. Modern Plastics Processing and Applications, 33(1), 36-39.

[25]. Vasičkaninová, A., Bakošová, M., Mészáros, A. &Klemeš, J. J. (2011). Neural network predictive control of a heat exchanger. Applied Thermal Engineering, 31(13), 2094-2100.

[26]. Widrow, B., & Lehr, M. A. (1990). 30 years of Adaptive Neural Networks: Perceptron Madaline and Backpropagation. Proc. IEEE, 78, 1415-1442.

[27]. Yang, R., Qiu, T. Q. & Ding, C. M. (2007). The simulation on ultrasound-enhanced supercritical fluid extraction with the artificial neural network. Computers and Applied Chemistry, 9, 1201-1204.

[28]. Yang, Y.,& Huang, D. (2002). Research and Applications of Artificial Neural Networks.Journal of East China University of Science and Technology, (05), 551-554.

[29]. Zupan, J., Noviča, M., &Ruisánchez, I. (1997). Kohonen and counter propagation artificial neural networks in analytical chemistry. Chemom. Intell. Lab., 38, 1-23.

[30]. Zhang, B., Yao, Y. & Zhang, H. (2021). Study on Extraction Technology Optimization and Antioxidant Activity of Tibetan Tea Polysaccharide. Chemical Reagents, 43(6), 842-847.

[31]. Zhang, B., Yao, Y. & Zhang, H. (2021). Study on Extraction Technology Optimization and Antioxidant Activity of Tibetan Tea Polysaccharide. Chemical Reagents, 43(6), 842-847.