

## Multi-Layer Learning Machines and Smart Sensor Applications

Magdi S. Mahmoud<sup>1\*</sup>, Ayman Al-Nasser<sup>1</sup>, Mutaz M. Hamdan<sup>1</sup>

<sup>1</sup>Systems Engineering Department, King Fahd University of Petroleum & Minerals, P. O. Box 5067, Dhahran 31261, Saudi Arabia.

msmahmoud@kfupm.edu.sa

\*Corresponding Author: Magdi S. Mahmoud.

Received: 17 March 2021

Accepted: 29 May 2021

Published: 07 June 2021

### ABSTRACT

Data-driven smart sensors are being widely used in industrial applications to estimate and evaluate the quality of critical variables. By using physical devices, most of the critical variables are being measured with difficulties. When process data is discarded, The quality variables sampling rate of majority of the smart sensors have been developed on labeled and labeled number of samples, small and large. The prediction accuracy enhancement quality will limit the large loss of information being generated from a measuring device. However, utilizing all available process data contained information, is a major and common issue of data-driven smart sensor.

In this article, a new Multi-Layer Learning Machine (MLLM) approach is recommended for the applications of smart sensors Using the Extreme Learning Machine as a foundation (ELM). Initially, a semi-supervised auto-encoders deep network structure is being used as an extraction for unsupervised feature with the reference to all process samples. At that point, extreme learning machine is used for regression with the quality variable added. In the meantime, the manifold regularization technique is presented for MLLM. The new strategy can deeply separate, and information is extracted from the data contains, and get more data from the unlabeled samples. The proposed MLLM procedure is implemented in a debutanizer column application to assess the C4 contents. Finally, the simulation result verify that our approach enhances both of the expectation and prediction accuracy in comparison with the available methods.

**Keywords:** Smart sensors, Deep learning, Extreme learning machine, Multi-layer ELM, Debutanizer Column.

### I. INTRODUCTION

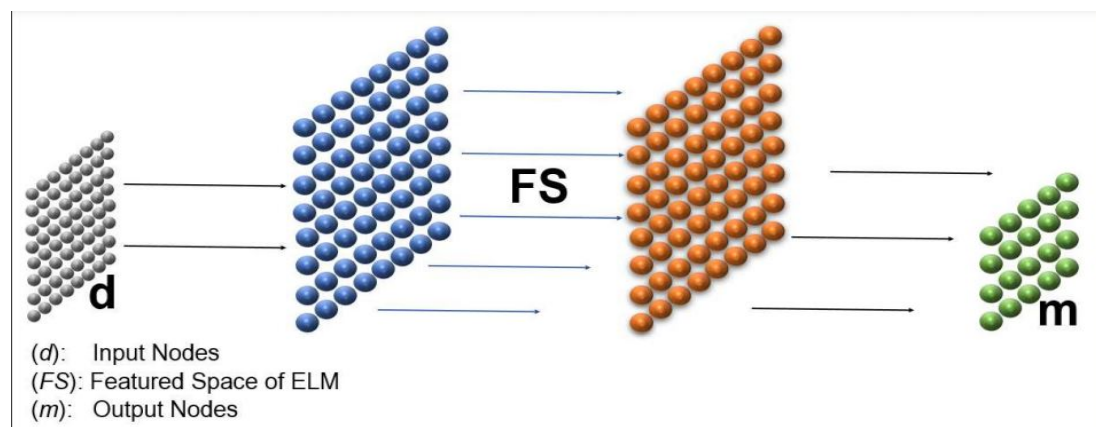
In the majority of the processes, the absence of sensors or their low-availability for each equipment will be affecting the online and continuous checking of item quality. This issue prompts intermittent creation of low-quality produced items bringing about refusing of the final produced item and resulting in loss of the industry. Part or the process models are Smart sensors which are being utilized for quality variables persistent continuous and online observing. Recently, The smart sensors have been

developed to be utilized for quality monitoring and checking in various processes for each industry, for example, polymer [1], maturation, bio-processes [2], size decrease [3-5], and rotating furnace to give some examples [6].

With regards to oil treatment facility (Refineries) and petrochemical industry, smart sensing methods was designed for predicting of various quality factors. A de-butanizer column usage is mainly for the separation of LPG and light gases comprising of the most part butane ( $C_4$ ), from the overhead distillate originating from the refining process (as well as splitting) unit. Quality control of the produced item in a de-butanizer column is a troublesome issue in view of absence of a real time observing system due to the multivariate nature and non-linearity of the process.

A de-butanizer column model based on back propagation neural network (BPNN) method was proposed back by Fortuna et al. for anticipating the base item chemical structure [7]. A report about partial least square (PLS) was partially generated for online soft sensor modeling by Song [8]. Ge et al. [9], implemented PLS and least square support-vector regression (LSSVR) for modeling soft sensor and support-vector regression (SVR).

The de-butanizer column model for predicting a desired process variable with non-linear semi-supervised principal component regression (PCR). A large portion of the data-driven smart sensors announced in the discipline of oil processing plants or various industries to date rely on the randomization of the construction of the Practice set from full data. The designation of Practice set for model turn of events [10], how an acceptable designation of the training data set can bring about a notable improvement in the expectation and output prediction of the model, is a major problem in the development of data-driven smart sensor application.



**Figure 1: Kernels, Output, Input Nodes and Features of Extreme Learning Machines.**

A novel approach with single hidden layer feed-forward neural system (SLFN) named extreme learning machine (ELM) was presented [5]. In the ELM, the weighted inputs, and hidden layer biases are shown with randomized generation method, and the weighted outputs are gotten by implementing a regularized least square method as appeared in FIGURE 1. Additionally, it has been demonstrated that SLFNs with randomized production of hidden neurons and weighted output keep up its general estimation capacity [11]. It likewise was demonstrated that ELM is significantly more proficient in comparison with ANN and BPNN in the training of the network [6]. As of late, ELM has pulled in increasingly more consideration and has additionally been effectively used with industrial applications that includes modeling of smart sensors [12].

As of late, deep learning (DL) became a major subject in process modeling. And with a fully improved prediction accuracy, smart sensor models using deep NN were enhanced. The DL methods are derived from a network architecture with a single hidden layer. Relatively, DL with multi-layer architecture demonstrates a re-representation with excellent efficiency as compared to the shallow network. The DL strategy, such as Deep Belief Network (DBN) [13], and Denoising Auto-Encoders (DAE) [14], has attained remarkable success in developing smart sensors overall performance. In any event, the BP algorithm usually trains these DL techniques and all the hidden parameters for the entire system framework should be finely tuned on different occasions.

Recently, the application of DBNs for smart sensors (SS) design and evidence was given by real-world cases study of how these structures can help in: learning more complex models, reducing computational complexity, efficiently use un-labelled data, extracting high-level features from data and exploiting them in SS design [15]. The presentation covered a wide range of machine learning algorithms for intelligent data analysis and applications [16]. Variety of machine learning methods were briefly discussed which can be used to solve a variety of real-world problems above. ML and DL models are presented and analyzed in terms of efficiency and effectiveness in smart grid applications [17].

For the development of multiple linear regression, major component regression, and back-propagation for the debutanizer column via ELM models, the Practice set was used. The execution of the obtained models was assessed by the use of validation data set in the simulation. The ELM models developed are capable of estimating with almost 95 percent precision the standard of the product as shown in the acquired results. It is discovered that the performance of MLLM models presented here is stronger than the performance of ELM regression and classical models of support vector regression.

For this work, for the de-butanizer process, data-driven smart sensors were used for the online continuous tracking of butane level in the base item of the de-butanizer column. The collected data set contains data for six input sources of information and one process output and by using the maximum intra-separation Kennard-Stone rule, the complete process data set was divided to training and testing sets. Different from the literature, the proposed methodology improves the fundamental ELM and MLLM's prediction performance by combining an unsupervised extraction feature and a semi-supervised learning machine, meaning that the methodology proposed acquires complete and extensive details from the data of the process. As a result, installing a smart sensor at the product's outlet line improves the efficiency of continuous product quality monitoring and control.

The following is a list of the paper's contributions;

1. A new Multi-Layer Extreme Learning Machine (MLLM) method is proposed for the applications of smart sensors based on the Extreme Learning Machine (ELM). different from the literature, the MLLM in our approach is designed to utilize the hidden layers and to provide better response and process output.
2. The simulation of the practice set was used to construct multiple linear regression.
3. The main component regression and back propagation through ELM models were implemented for enhancing the final products of the de-butanizer column.
4. The execution of the models produced was assessed by simulating them with a data set of tests.

- the overall execution of the models of the MLLM revealed in this article is seen as a supercilious to the execution of the models of the ELM regression and the support vector regression.

The following is how the rest of this article is organized: A background on Extreme Learning Machine (ELM) will be provided in Section 2. The model development procedure is discussed in Section 3. In Section 4, The Multi-layer learning machines (MLLM) scheme for smart sensors is described. A case study discussing the implementation of the MLLM on debutanizer column application presented in Section 5. Finally, in Section 6, concluding remarks are given. See TABLE 1.

**Table 1: Nomenclature.**

BPNN	Back Propagation Neural Network
MAE	Mean Absolute Error
MLR	Multiple Linear Regression
PC	Principal Component
PCA	Principal component analysis
PCR	Principal Component Regression
$R$	Correlation Coefficient
RMSE	Root Mean Squared Error

*Notations:* We denote  $IN$  as the  $N$  dimension identity matrix in this paper.  $L$  is the graph Laplacian matrix,  $L = D - U$ ,  $U = [\mu_i, j]$  is the similarity matrix,  $D$  is the diagonal matrix with the elements  $D_{ii} = \sum j^{l+u} \mu_i, j$ ,  $J(x)$  is the hidden layer of the ELM output matrix,  $\Psi_i$  is the hidden layer of the MLLM output matrix, and  $W_i$  is the weight matrix of the hidden layer between  $i$  and  $i$ .

## II. BACKGROUND ON EXTREME LEARNING MACHINE

Extreme Learning Machine (ELM) is a practice algorithm that proficiently decides the parameters of SLFN [5]. In the ELM model training phase, we are assuming that  $N$  training sample(s) with input variables  $X \in \mathbb{R}^{N \times m}$  and output variable  $Y \in \mathbb{R}^{N \times r}$  are gathered, where  $m$  and  $r$  represent the dimensions. So, ELM can retrieve the approximation function which will be used to evaluate the estimation of the output  $Y$ . See FIGURE 2.

The training of the ELM comprises of two phases as follows:

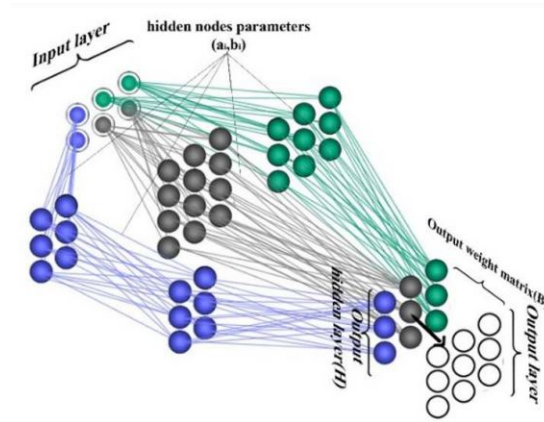
- The randomized mapping stage: It is to build the hidden-layer using a certain number of created and randomized mapping neurons such that the following sigmoid function is used to conduct the mapping [18]:

$$J(x; \theta) = (1 + e^{-(\omega^T x + b)})^{-1} \quad (1)$$

where  $\theta = \{\omega, b\}$  are the mapping function parameters.

- The parameters solving stage: The solution of a regularized least square problem is used to acquire:

- (a) Parameters of the hidden mapping functions which will be generated randomly.
- (b) The remaining parameters.
- (c) The weighted output.



**Figure 2: Multilayers perceptron-based ELM.**

In this article, we characterize the hidden-layer output vector as  $J(x) \in \mathbb{R}^{(N \times n)}$ , where the quantity of hidden-neurons is  $n$ . the assumed output function of the  $i$ th hidden-node for a single hidden-layer of ELM is  $\psi_i(x) = (a_i, b_i, x)$ , with the parameters of the  $i$ th hidden-node denoted by  $a_i$  and  $b_i$ . The ELM output function for SLFNs consists of  $L$  hidden-nodes will be:

$$f_L(x) = \sum_{i=1}^L \beta_i \psi_i(x) \quad (2)$$

where  $\beta_i$  is the weighted output of the  $i$ th hidden-node.

$$\psi(x) = [F(\psi_1(x), \dots, \psi_L(x))] \quad (3)$$

is the hidden layer output mapping of ELM. The hidden layer output matrix  $\Psi$  of ELM for  $N$  practice samples and the practice data target matrix  $T$  are represented by:

$$\Psi = \begin{bmatrix} \psi(x_1) \\ \psi(x_N) \end{bmatrix} = \begin{bmatrix} F(a_1, b_1, x_1) & \dots & F(a_L, b_L, x_1) \\ \vdots & \vdots & \vdots \\ F(a_1, b_1, x_N) & \dots & F(a_L, b_L, x_N) \end{bmatrix} \quad (4)$$

and

$$T = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix} \quad (5)$$

By and large, ELM is a sort of regularization neural systems with tuneless hidden-layer mappings (This means it is shaped by either randomized hidden-nodes, kernels or different implementations). The ELM objective function is:

$$\text{Minimize: } \|\beta\|_p^{\sigma_1} + C \|\Psi\beta - T\|_q^{\sigma_2} \quad (6)$$

Where  $\sigma_1 > 0, \sigma_2 > 0, p, q = 0, \frac{1}{2}, 1, 2, \dots, +\infty$ .

For regression, classification, compression, sparse coding, clustering, and characteristic learning, the combinations of  $\sigma_1, \sigma_2, p$ , and  $q$  can be utilised and result in various learning algorithms.

However, a easy ELM practice algorithm can be as a particular case learning a type of single hidden-layer NN:

$$\hat{Y} = W_2 \sigma(W_1 x) \quad (7)$$

where  $W_1$  is the hidden-layer input matrix weights,  $\sigma$  is the activate function, and  $W_2$  is hidden-layer output matrix weights.

From (1), with the output weights  $\gamma \in \mathbb{R}^{n \times r}$ , the network output can be shown as follows:

$$\hat{y}_i = J(x_i)\gamma, i = 1, 2, \dots, N \quad (8)$$

During the phase of parameter solving, the weighted output is obtained by minimization the below un-constrained optimization problem:

$$\min_{\gamma \in \mathbb{R}^{n \times r}} \Gamma_{ELM} = \frac{1}{2} \|\gamma\|^2 + \frac{C}{2} \|Y - J\gamma\|^2 \quad (9)$$

where  $J = [J(x_1)^T, \dots, J(x_n)^T]^T \in \mathbb{R}^{N \times n}$ . The 1st item in (9) is the term of regularization in opposition to over-fitting, and the second one is the vector of error, where  $C$  is called a coefficient of penalty. We can obtain the below equation by setting the gradient of  $\Gamma_{ELM}$  to zero with respect to  $\gamma$

$$\nabla \Gamma_{ELM} = \gamma + CJ^T(Y - J\gamma) = 0 \quad (10)$$

Usually, in the situation where the hidden matrix  $J$  has full column rank and the practice samples number is extra than the hidden-neurons number, equation (11) will be:

$$\gamma^* = \left(J^T J + \frac{I_n}{C}\right)^{-1} J^T Y \quad (11)$$

where  $I_n$  is the identity matrix of dimension  $n$ . Despite what might be expected, if  $J$  is loaded with row rank, which implies higher hidden-neurons in comparison with the training sample(s), then the outcome of  $\gamma^*$  will be as below:

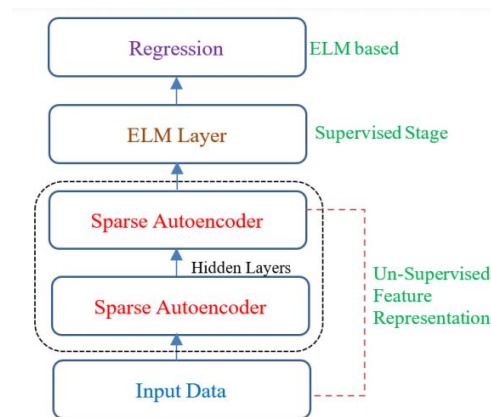
$$\gamma^* = J^T \left(J^T J + \frac{I_N}{C}\right)^{-1} Y \quad (12)$$

where  $I_N$  is the identity matrix of dimension  $N$ .

Considering this, a various leveled (Multi-Layer) ELM (MLLM) was presented [19], which has indicated a much effective performance in comparison with the conventional DL techniques in the arrangement field. The MLLM has two primary segments:

1. **The un-supervised option** is a feature with element extraction where the ELM-based auto-encoder (ELM-AE) is used and each layer in the stack architecture might be considered as a sub-module.
2. **The supervised option**, is a feature for classification where the deep profound level features are initially split using randomized matrix, and afterward the first ELM is implemented to ultimate choice's creation or prediction. The irregular element mapping theory of ELM is not needed for the whole framework system, which lead to a faster preparing speed than the conventional BP-based profound NNs.

In any case, the technique of MLLM is yet to be used as desired for modeling of smart sensor in the process industry. In practicality, input variables being utilized to model smart sensor, are those quick sampling process variables such as flow rate, temperature, and weights, where the output variables are key factors which are hard to gain because of lower testing and rate of sampling, cost, time consuming and entangled chemical analysis,... etc [20]. See FIGURE 3.



**Figure 3: Multi-Layer extreme learning machine (H-ELM) architecture.**

Samples of data that have, solely input part called unlabeled data while labeled data consist of output and input parts. Subsequently, in practicality, the data obtained from the real process industry is typically partly named or labeled. In addition, the wide variety of unlabeled data is typically much greater than that of labeled data, and it is not yet sufficient to build a smart sensor model in simple terms, depending on the samples labeled. But the traditional approaches used to model smart sensor are supervised algorithms that use only labeled Procedure data samples with their acceptable quality variable, where the enormous number of data samples that are unlabelled is ignored.

Referring to semi-supervised algorithm, it presumes that the dataset which is input, in the input area can join the unlabeled and labeled data in the learning procedure in addition to that it is following same structure. Plus, the mainly used semi-supervised technique is called manifold-regularization, which is assuming that the contingent probabilities of two comparative samples

ought to be comparable also [21]. It tends to be handily installed in a conventional technique to improve a supervised strategy into a semi-supervised structure. The regularization of manifold structure is used in few algorithms to frame the semi-supervised learning machine and increased unrivaled performance [22].

A semi-supervised MLLM based DL model shown in this work, is used for smart sensor performance enhancement. The set of gathered process data will be utilized for modeling. The algorithm used here for the modeling consists of two major components: the semi-supervised regression which will be used for prediction and unsupervised feature extraction.

To start with, the AEs structure of deep network system is revised to meet the actual condition for un-supervised extraction feature including all process samples inputs. At that point, at a significance system level, the first ELM layer is used to relapse by attaching the quality variable (the smart sensor output variable).

In the ELM semi-supervised learning algorithm, the regularization of manifold technique is inserted. Consequently, the modeling method is not just with deep of the data that it is containing through the deep system network however taking benefit out of the extra un-labeled processes sample data too. The proposed semi-supervised MLLM model adequacy and achievability for smart sensor improvements are shown in an industrial actual case.

In this study, by re-designing an inferential debutanizer sensing system, We make an effort to tackle the problem of trouble progressively observing of the product quality . Data driven smart sensors are sophisticated of the debutanizer column bottom product for butane content prediction.

The output and input data set of the debutanizer cycle was derived from real application data and was brought into everyone's Knowledge by Fortuna et al. as an example. [23]. [24] and [25], have previously used this benchmark data set to advance least square support-vector regression and main component regression models for the debut column. The developed models' overall performance, was calculated through simulation with the data-set after validation. Effects from simulation results, correlation coefficient (R), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and parameter values were determined and calculated for statistical model assesment. The findings clarify the better execution of the model of neural network of the Levenberg-Marquardt algorithm-trained back propagation described in this manuscript in compare with the models of support vector regression.

### **III. MODEL DEVELOPMENT USING SEMI-SUPERVISED MLLM**

#### **3.1. Multi-Layer Learning Machine**

Deep NN such as stack auto-encoder has a common use for mining data features to achieve high generalization efficiency [24]. MLLM is a Hierarchical Extreme Learning Machine network with several layers of ELM-AE vector parameter learning [26]. For a superior prediction output, MLLM combines ELM's learning efficiency and AE's deep structure.

The ELM-AE is essentially based on the ELM principle, which can effectively guarantee the ability to approximate continuous functions by random mapping. In a multilayer network, the AE functions as a function extractor as shown in FIGURE. 4. The encoded outputs are used to minimize reconstruction errors for an approximation of the original inputs.



Mathematically, via the  $\psi = g((A \times x) + b)$  mapping, the AE is used for mapping the input data  $x$  to the hidden layer  $\psi$ , with  $A$ ,  $b$ , and  $g(\cdot)$  represent the input weight matrix, the bias vector, and the activation function, respectively. Then,  $\psi$  is then mapped to the reconstructed  $\hat{x} = g((A' \times \psi) + b')$  input vector using the parameters  $A'$  and  $b'$ .  $A$  is generated randomly in the ELM-AE and the mapped output  $\hat{x}$  is considered as  $x$ . Finally, the output weight is determined as  $A'$ . It is noted here that fine-tuning of parameters is not needed once the AE is initialized.

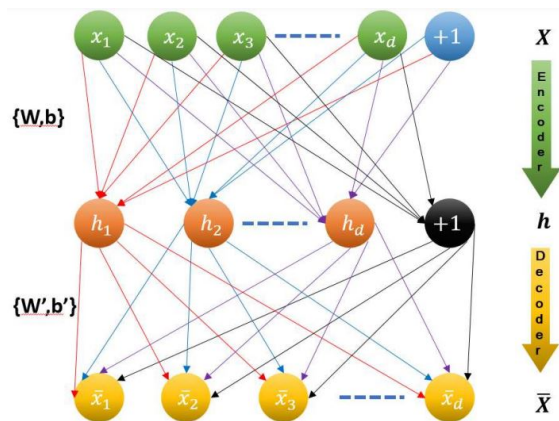


Figure 4: Autoencoder Structure.

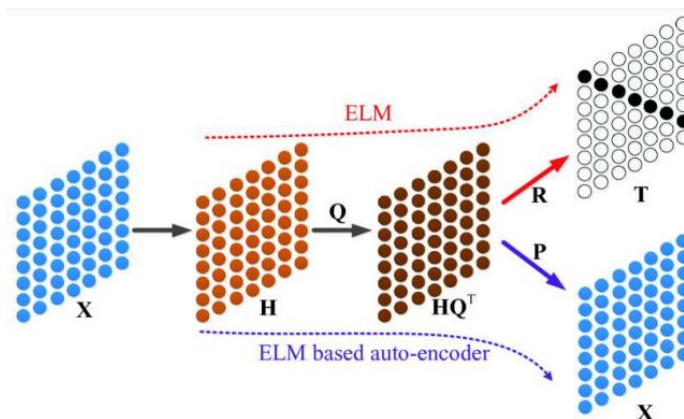


Figure 5: Separated ELM-based autoencoder [27].

the procedure of learning MLLM model from the practice samples  $X$  and  $Y$  is represented in FIGURE 5. A completely integrated multilayer network is compressed here with  $h$  hidden layers. As mentioned in section II, MLLM network consist of two parts: the extraction of unsupervised features with multilayer ELM-AEs and the ELM regression with supervision.

### 3.2. Semi-Supervised Mllm

Referring to FIGURE 5, MLLM could be used in classifying and regressing data, although it is a administered algorithm with only one input equal to only one output. The quality variable is sampled very low in real processes, this indicate that samples of quality are much lower than the variables of the process. Commonly, a limited number of labeled samples are built on smart sensors and several unlabeled samples are discarded. Therefore, in this section, a multi-layer learning machine (MLLM)

approach for modeling smart sensor is presented to utilize the bigger unlabeled dataset which contains a lot of unused the procedure information.

The model's prediction performance and precision could be enhanced by implementing additional unadministered learning mechanism to evaluate the sample's distribution. Manifold ordinance is an un-supervised learning constraint that is derived from the multiple learning of all available samples to achieve the distribution of data [21].

**Assumption 1** The labeled samples  $S_l = \{x_i + y_i\}_{i=1}^l$  and unlabeled samples  $S_u = \{x_i\}_{i=1}^u$  have close similar samples. So, the most similar data have the same distribution aiming to minimize  $\hat{\Phi}$  which described as follows:

$$\hat{\Phi} = \frac{1}{2} \sum_{i,j} \mu_{ij} \| \hat{y}_i + \hat{y}_j \|^2 \quad (13)$$

where  $\hat{y}_i$  and  $\hat{y}_j$  denote the predicted outputs with respect to the samples  $x_i$  and  $x_j$  and  $\mu_{ij}$  is the similarity between  $x_i$  and  $x_j$ , constructed using a Gaussian function as follows:

$$\mu_{ij} = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} \quad (14)$$

Now, (13) will transform into the following equation:

$$\hat{\Phi} = Tr(\hat{Y}^T L \hat{Y}) \quad (15)$$

with  $L = D - U$  denotes the graph Laplacian matrix [21] such that the similarity matrix  $U = [\mu_{ij}]$  and  $D$  is a diagonal matrix with the elements  $D_{ii} = \sum_j^{l+u} \mu_{ij}$ .

The regularized item (15) has to be considered in (9) to better the guess performance of the semi-administered learning algorithm. So, it can be formulated as:

$$\begin{aligned} \Gamma_{SS-ELM} &= \min_{\gamma \in \mathbb{R}^{n \times r}} \frac{1}{2} \|\gamma\|^2 + \frac{c}{2} \|e_i\|^2 + \frac{\lambda}{2} Tr(\hat{Y}^T L \hat{Y}). \\ e_i &= y_i^T - J(x_i)\gamma, i = 1, \dots, l. \\ \hat{Y}_i &= J(x_i)\gamma, i = 1, \dots, l + u \end{aligned} \quad (16)$$

where  $L \in \mathbb{R}^{(l+u) \times (l+u)}$ ,  $\hat{Y}$ , and  $\lambda$  are the graph Laplacian matrix, the assumed output matrix mapped by all the inputs, and the tradeoff parameter, respectively. Moreover, one can write (16) in the following format:

$$\Gamma_{SS-ELM} = \min_{\gamma \in \mathbb{R}^{n \times r}} \frac{1}{2} \|\gamma\|^2 + \frac{c}{2} \|Y - J\gamma\|^2 + \frac{\lambda}{2} Tr(\gamma^T \Psi^T L \Psi \gamma) \quad (17)$$

where  $Y$  and  $\Psi \in L \in \mathbb{R}^{(l+u) \times n}$  are the labeled sample's genuine output and the hidden neurons with relation to all of the accessible samples' input, respectively. Also  $\Psi$  is represented as:

$$\Psi = \begin{bmatrix} J(w_1^T x_1 + b_1) & J(w_2^T x_1 + b_2) & \cdots & J(w_n^T x_1 + b_n) \\ J(w_1^T x_2 + b_1) & J(w_2^T x_2 + b_2) & \cdots & J(w_n^T x_2 + b_n) \\ \vdots & \vdots & \vdots & \vdots \\ J(w_1^T x_{l+u} + b_1) & J(w_2^T x_{l+u} + b_2) & \cdots & J(w_n^T x_{l+u} + b_n) \end{bmatrix} \quad (18)$$

The output weight is calculated by setting the gradient of (17) to 0, such that:

$$\gamma^* = I_n + C J^T J + (\lambda \Psi^T L \Psi)^{-1} C J^T Y \quad (19)$$

where  $I_n$  is an identity matrix of dimension  $n$ . If the number of labelled data is smaller than the number of hidden neurons, the output weight is built as in (11):

$$\gamma^* = \tilde{J}^T (I_{l+u} + C \tilde{J}^T \tilde{J} + \lambda \Psi^T L \Psi)^{-1} \tilde{J}^T C \tilde{Y} \quad (20)$$

where  $I_{l+u}$  is an identity matrix with dimension  $l + u$ ,  $\tilde{J}$  and  $\tilde{Y}$  are augment matrix with the last  $u$  rows equal to 0 and the first  $l$  rows equal to  $J$  and  $Y$ , respectively.

Lastly, the last layer of MLLM is then converted to the semi-administered form after constructing the semi-administered ELM. Subject to FIGURE 4, when the final hidden layer's output is determined and evaluated using (7), the input ( $X$ ) of the last ELM will be taken and the consistency samples partially collected ( $Y$ ) will be integrated to generate the semi-administered ELM. Finally, via (14) or (15), The final output weight is determined, and the MLLM model is entirely formulated.

#### 4. PROCEDURE FOR MODELING SMART SENSORS USING MLLM

The procedure of dealing with smart sensors using MLLM is described in the following steps

##### MLLM SCHEME FOR SMART SENSORS

1. Identify the inputs  $X$  and  $Y$ , the output  $W_i (i = 1, \dots, \psi)$ : weights between the  $i - 1$ th and  $i$ th hidden layer,  $\gamma$  output weights;  $Y_{pre}$ : the testing sample's output prediction. And the original datasets: The penalty coefficient  $C$  and the trade-off parameter  $\lambda$ .
2. Select the variables by implementing the theoretical analysis and operator Experiences utilizing smart sensor data driven.
3. Select the training and the validation datasets, and normalize it.
4. Assign values for  $C$  and  $\lambda$ .
5. For each layer, set the number of neurons and the number of layers for the MLLM.
6. Make random selection for the input weights and the weights between the hidden neurons.

7. Run the training of the ELM-based multilayer autoencoder using both the labeled and unlabeled samples and record the weight matrices  $W_i$  in each layer.
8. For the last layer, train the Semi-supervised ELM with a quality variable and record the output weight  $\gamma$ .
9. Formulate the MLLM model for output prediction and check its performance.
10. Expect the standard variable ( $Y_{pre}$ ) for the testing of samples.
11. Find information about the accuracy of training and testing.
12. Repeat the steps again if the training and testing accuracy are not acceptable.

## 5. CASE STUDY: APPLICATION DEBUTANIZER COLUMN

For the Debutanizer column, 4,200 input and 700 output process data values were available in total. Taken from the web resource, this data set. The dataset available was divided into an equal number of training sets (utilised in the development of models) and a set of validation (used for model assessment).

There are 700 input and 100 output data values in each data subset. The ELM and MLLM algorithms were used to create the Practice set from the whole total data. For the algorithm's implementation, the needed MATLAB code was taken from a freely available toolbox [28]. This training data was then utilised to construct a semi-supervised model.

In the multiple linear regression (MLR) model, the output is stated as a linear combination of the inputs. The shape of the MLR model for the Debutanizer column is as follows:

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \beta_3x_3 + \beta_4x_4 + \beta_5x_5 + \beta_6x_6 + \beta_7x_7 \quad (21)$$

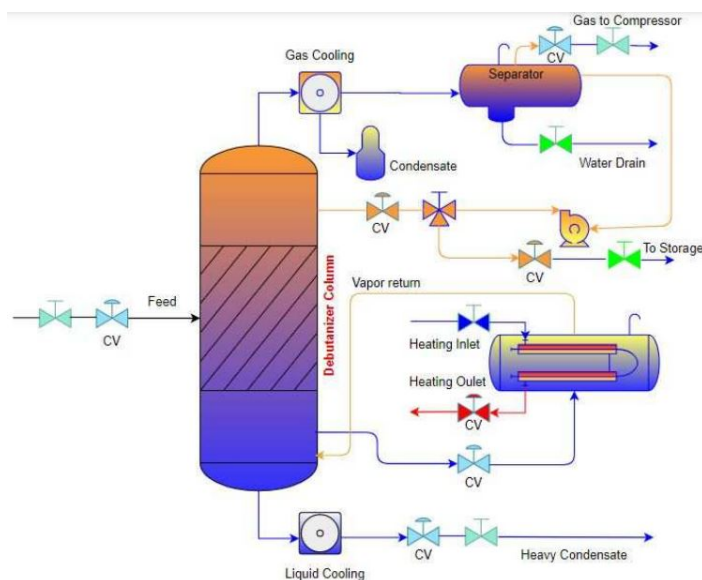
There are several old methods before machine learning where one of them is neural networks widely used and, in this work, we will show a comparison between our proposed method and the machine learning method. Also, we will show the difference in the response between ELM and MLLM for the same set of data retrieved.

Referring to FIGURE 1, one can identify how MLLM works against NN and deep learning, and large ELM which is a sort of regularization of systems with tuneless hidden-layer mappings (This means it is shaped by either randomized hidden-nodes, kernels, or different implementations within the Featured space before accessing the main system model).

The intention behind proposing this solution is that Multi-Layer Learning Machine is a tool that can be utilised for any part of any chemical process and can be replicated as well in other various applications in more efficient way comparing to the conventional machine learning and NN.

A column of debutanizer is part of a refinery's several processing units. The Debutanizer column is utilised to extract the lighter fractions from the gasoline anywhere there is output of LPG and the gasoline, e.g. in the atmospheric and vacuum crude

distillation unit, cracking and coking units. The un-stabilized naphtha is the feed to the debutanizer column and the products coming out of the column are LPG as the top product and the bottom product is gasoline / stabilized naphtha. See FIGURE 6.



**Figure 6: Debutanizer Column.**

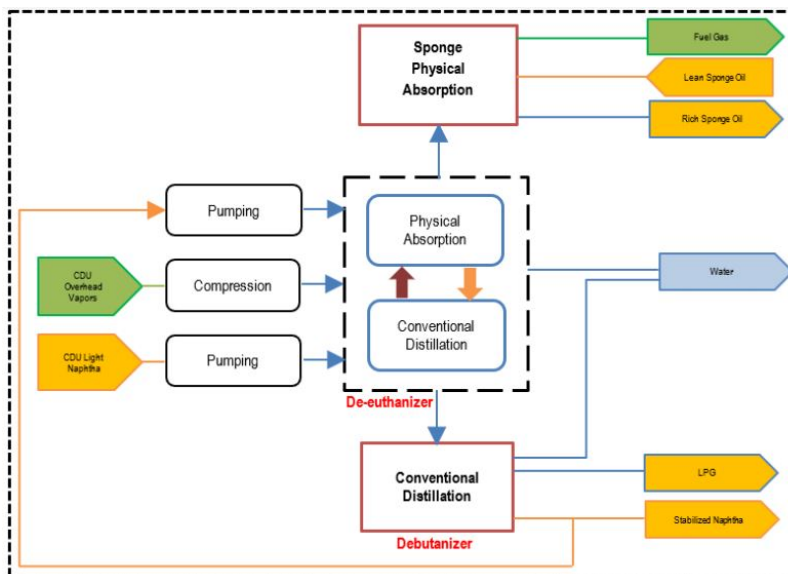
The diagram of the schematical process is shown in FIGURE 4. The butane (C4) content in the bottom product should be minimized for improved process efficiency. This includes close monitoring of the quality of C4 in the product at the bottom. For this reason, In the process, a gas chromatograph is used. However, in the bottom flow line coming from the debutanizer column, the hardware sensor (gas chromatograph) is not mounted and is instead placed in the overhead of the de-isopentane column that is positioned some distance away from the column of the debutanizer.

**Table 2: Input-Output Process Variables for the debutanizer column.**

	Variables	Description
Inputs	$x_1$	Top Temperature
	$x_2$	Top Pressure
	$x_3$	Reflux
	$x_4$	Flow to next process
	$x_5$	6th tray temperature
	$x_6$	Bottom temperature
Outputs	$y$	Butane (C4) content in the debutanizer column bottom

This causes a measurement time delay of 30-75 min [26]. A smart sensor can therefore be used to resolve the time delay issue of the hardware sensor in the bottom flow of the debutanizer column. The C4 content present in the debutanizer column bottom product is the performance to be predicted by the smart sensor model.

This output depends, as stated in the literature [7,8], on seven procedure inputs. In TABLE 2, the seven process inputs and the output quality variable to be estimated by the smart sensor are described. The positions of the sensors for the seven process inputs, the gas chromatograph used for the content measurement of C<sub>4</sub> and the smart sensor proposed are shown in FIGURE 4.



**Figure 7: Debutanizer Process Schematic Diagram.**

Before performing the model training, some parameters are established based on the previous driven smart sensor data's experienced values. The C penalty coefficient and trade-off parameter  $\lambda$  are not sensitive in a stable area. For ELM, the C value is usually set in the [1,10] interval. To measure the value of C, the interval [5] is used for MLLM. For both ELM and MLLM, the  $\lambda$  is usually set to [0.1 , 0.5]. See FIGURE 7.

The multiple parameters are established by trial and error in this paper during the experimental breaks with the assesment dataset. Usually, the hidden neuron's number for ELM is established closed to the number of input neurons. Whereas the hidden neuron's number of the AEs N<sub>ae</sub> is typically less than the MLLM input neuron's number.

Then, the hidden layer's number should be listed and described for creating the MLLM model. The models are trained on multiple hidden layer numbers respectively and used to guess the output performance of the validation dataset. In TABLE 3, the prediction results are shown. Both ELM and MLLM were validated on the above-mentioned layers of the hidden neurons. We need to find MAE "Mean Square Error" to evaluate RMSE, and for all developed models the correlation coefficient (R) values can be determined.

$$MAE = \frac{\sum_{i=1}^N |y_i - \hat{y}_i|}{N} \quad (22)$$

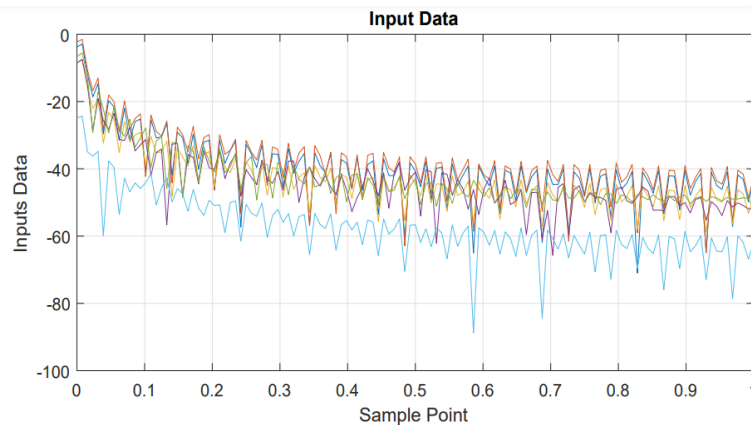
Here  $y_i$  and  $\hat{y}_i$  are actual and model predicted values for the  $i$ th observation and N is the number of observations.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y - \hat{y})^2}{N}} \quad (23)$$

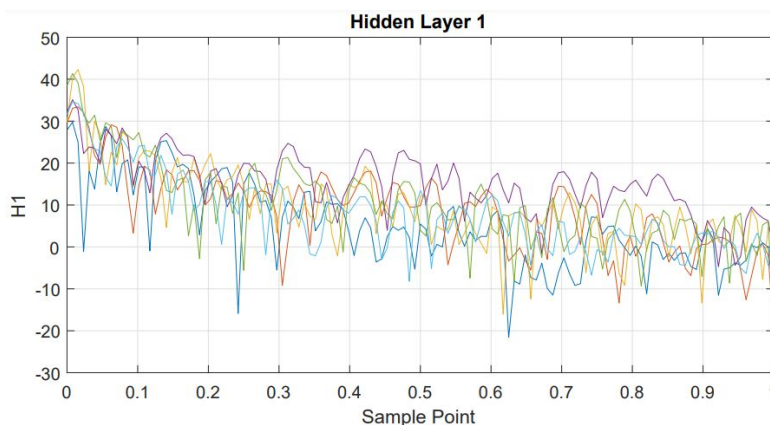
$$R = \left( \frac{\sum (y - \hat{y})(\hat{y} - \hat{y}')}{\sum (y - \hat{y}')^2} \sum (\hat{y} - \hat{y}')^2 \right) \quad (24)$$

To enhance the execution of the deep network, both the hidden layers numbers and data sizes has to be increased simultaneously. It should be noted and observed that in this case, all the results are mean values of 20 times the training and predictions. So, the number of hidden layers will be set to 3 in the following tests.

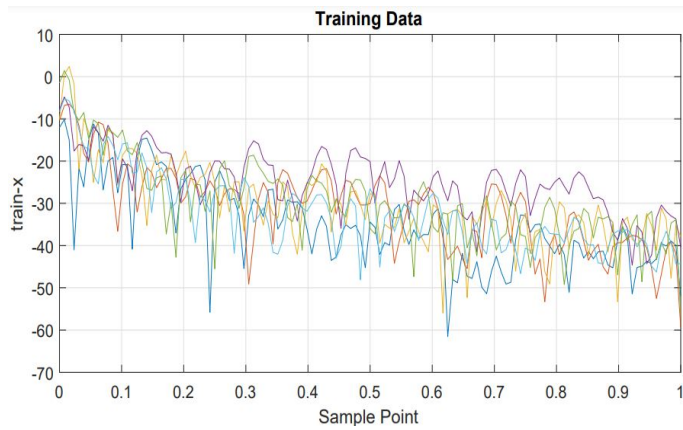
Second, MLLM 's research work is carried out on various scales of unlabeled samples to show that the extra unlabeled samples will have a significant change in the output of the prediction. The unlabeled samples increased from 0 to 700. (FIGURE 8: FIGURE 18). The following simulations are now showing the final details of the;



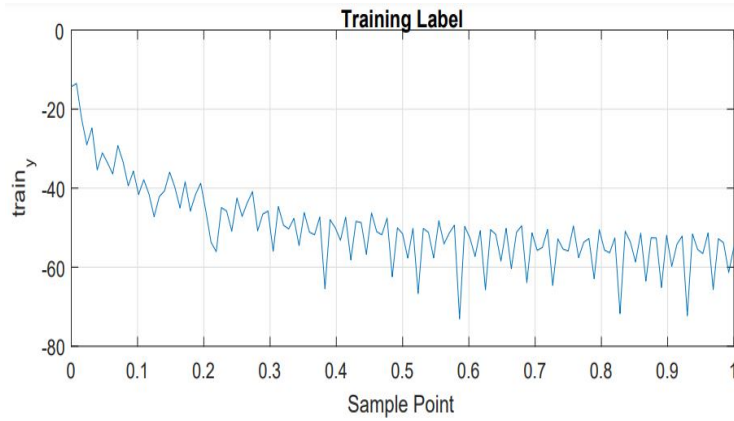
**Figure 8: Input Data.**



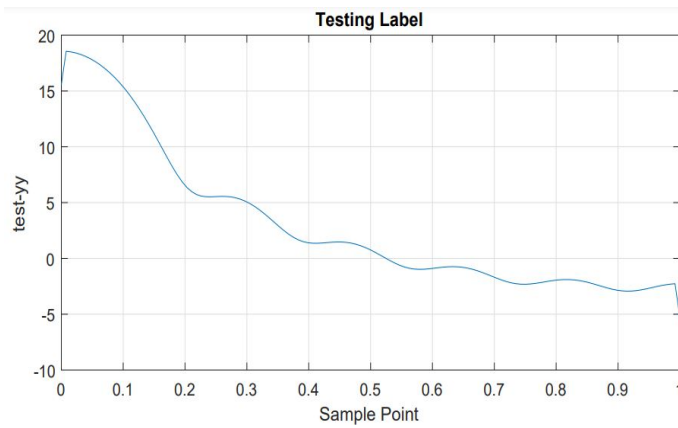
**Figure 9: Output Data.**



**Figure 10: Training Data.**

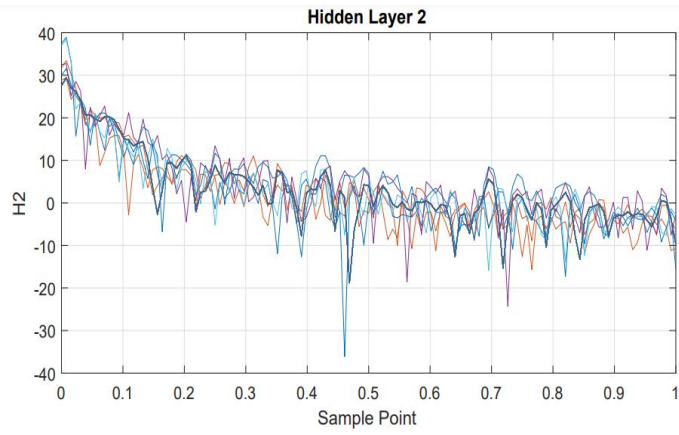


**Figure 11: Training Label.**

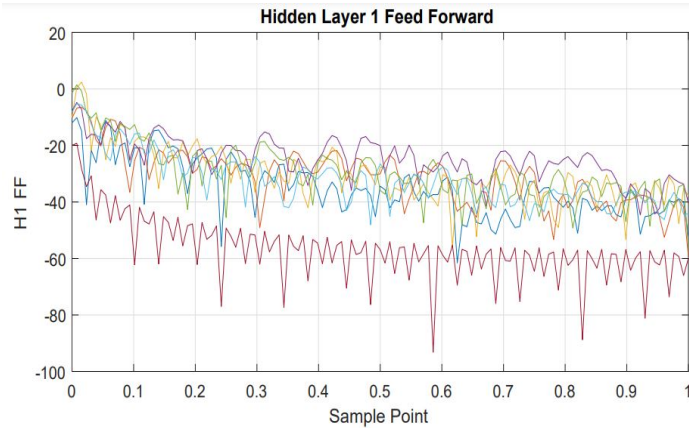


**Figure 12: Testing Data.**

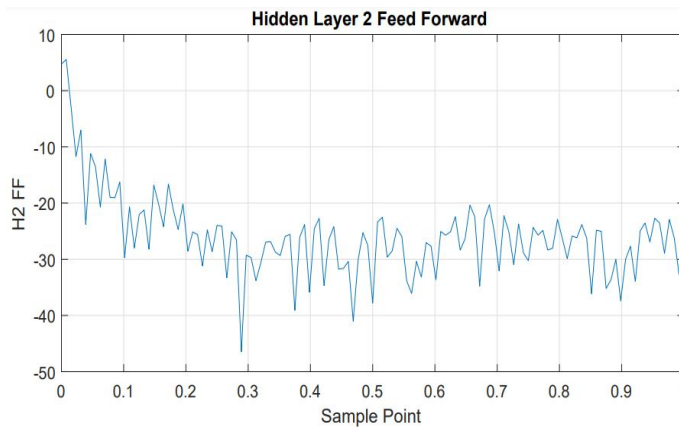




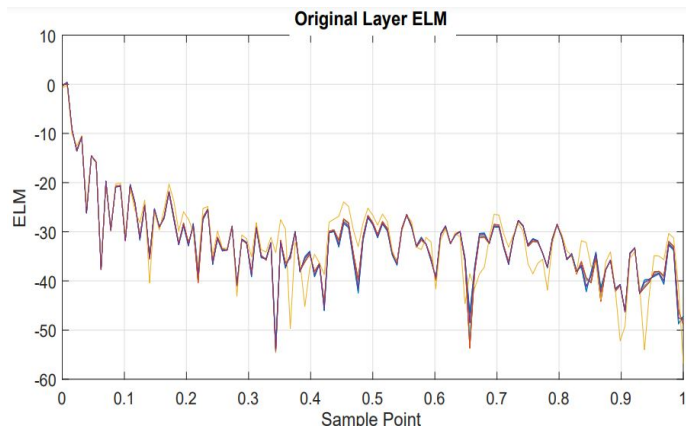
**Figure 13: Testing Label.**



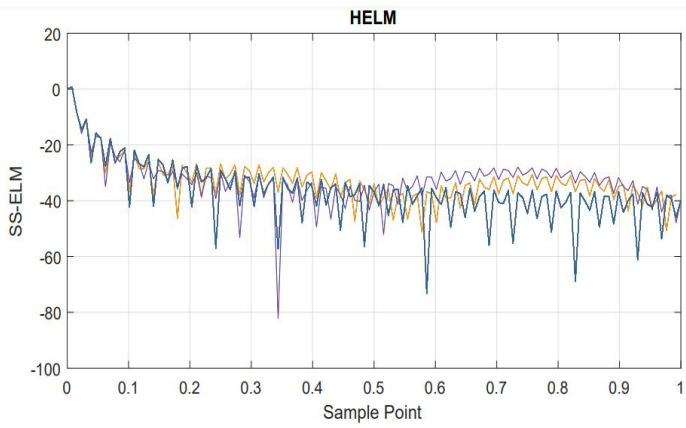
**Figure 14: Hidden Layer 1 Feed Forward.**



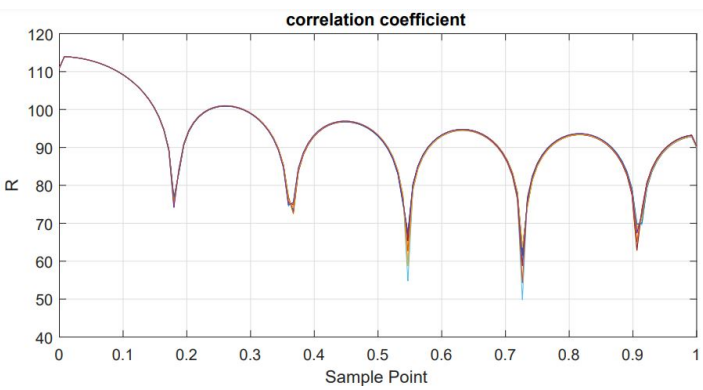
**Figure 15: Hidden Layer 2 Feed Forward.**



**Figure 16: Original Layer ELM.**



**Figure 17: MLLM Response.**



**Figure 18: Correlation Coefficient.**

## 6. RESULTS

**Table 3: Results.**

Layer 1: Max Val of Output 26213.761587
Min Val -26120.951476
Layer 2: Max Val of Output 505.539933 Min Val 11.250511
Layer 3: Max Val of Output 0.116162 Min Val 0.100000
Layer 3: Max Val of Output 0.097980 Min Val 0.109091
Layer 3: Max Val of Output 0.102020 Min Val 0.101010
Layer 3: Max Val of Output 0.097980 Min Val 0.117172
Layer 3: Max Val of Output 0.097980 Min Val 0.046465
Layer 3: Max Val of Output 0.110101 Min Val 0.000000
Training has been finished!
The Total Training Time is : 0.01506 seconds
Training Accuracy is : 44.4444 %
Testing has been finished!
The Total Testing Time is : 0.011848 seconds
Testing Accuracy is : 95 %
RMSE = $1.0e + 05 \times$
3.5349 3.5349 3.5349 3.5349 3.5349 3.5349
3.5349 3.5349 3.5349 3.5349 3.5349

From the results and simulations, we clearly see that how MLLM is having fast response to train and test a huge number of data in a very short time but with higher accuracy utilizing the hidden layers and treating each layer separately. Also, we can see clearly the difference between ELM and MLLM where more data has been used for training and testing data.

As a result, the usage of semi-supervised techniques to increase the training data size, will take place whenever it is needed to produce an accurate model without the availability of enough labeled data and the ability or resources to retrieve additional data. Then the dataset will be labeled with fake data samples which we know, while the remaining data will stay unlabeled. Then, the retrained model will be applied to new data, with better accuracy identifying fake samples using supervised machine learning techniques.

## 7. CONCLUSION AND OUTLOOK

The exact model of ELM that has been developed is shown in Figure. in this work. 6., It takes several procedure variables as outputs and inputs the projected value of C4 level in the bottom de-butanizer's product column. The current optimized MLLM model architecture, Weight is valuing of outer and hidden layers, and the input/output data utilised for model building and assesment are provided as supplementary materials to support the reported findings and encourage future study.

The butane content of the bottom product in the de-butanizer column should be limited to a minimum in order to achieve best process performance. However, to achieve this level of product standard control, online tracking of the product composition is needed. For example, a gas chromatograph would be used as a hardware sensor in such an application to monitor gas composition, which is positioned a little far from the column and shows significant changes in the time delay of the measured value.

Thus, installing a smart sensor at the outlet line of the product, Boost the efficiency of product quality monitoring and control with continuous monitoring and control A simulation analysis reveals that actual data collected by the smart sensor, It outperforms the ELM-based multiple linear regression and principle component regression models in this study.

A semi-supervised MLLM algorithm can be retrieved or derived from the simple MLLM by using the manifold regularization approach where the same has been used for smart sensor modeling. For unsupervised extraction of both unlabeled and labeled samples, the deep network structure of the AEs was first applied, In the final hidden layer, a semi-supervised ELM based on manifold regularisation was utilised for regression.

The combination of an unsupervised extraction feature and a semi-supervised learning machine enhances the basic ELM and MLLM 's predictive performance, which implies that the proposed approach obtains comprehensive and wide information from the process data. The real high-low unit industrial process has shown how effective and superior the proposed MLLM algorithm in smart sensor modeling is, which can be further developed and used in the estimation process for the C4 product. Sensors and instruments are used so often in industrial processes, and research on the application of smart sensors will therefore become a major subject [25,29], which can further help to obtain the health of distributed industrial systems used for monitoring. ELM-based algorithms using fast learning speed can also be used in the intensive data area for distributed modeling, which is often aimed at effective [30] computing. In addition , it is possible to use several process data types, such as videos, voices and graphs, then learn additional detailed information via the proposed DL network and control. This approach proposed here will, therefore, be extended further.

## 8. ACKNOWLEDGMENTS

This work is supported by the Deanship of Scientific Research (DSR) at KFUPM through distinguished professorship award project no. **DUP19106**.

## 9. REFERENCES

1. Kadlec P, Gabrys B, S. Strandt. Data-Driven Soft Sensors In The Process Industry. Computers & chemical engineering. 2009;33:795-814.
2. Kadlec P, R. Grbic, and B. Gabrys. Review of Adaptation Mechanisms For Data-Driven Soft Sensors. Computers & chemical engineering. 2011;35: 1-24,.
3. Gonzaga J, Meleiro LAC, Kiang C, Filho RM. Annbased Soft-Sensor For Real-Time Process Monitoring And Control Of An Industrial Polymerization Process. Computers & chemical engineering. 2009; 33:43-49.
4. Rani A, Singh V, Gupta J. Development Of Soft Sensor For Neural Network Based Control Of Distillation Column. ISA transactions. 2013;52:438-449.

5. Huang GB, Zhu QY, Siew CK. Extreme Learning Machine: Theory And Applications. *Neurocomputing*. 2006;70:489-501.
6. Huang GB, Zhou H, Ding X, Zhang R. Extreme Learning Machine For Regression And Multiclass Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. 2011; 42:513-529.
7. Fortuna L, Graziani S, Xibilia MG. Soft Sensors For Product Quality Monitoring In Debutanizer Distillation Columns. *Control Engineering Practice*. 2005;13:499-508.
8. Ge Z, Song Z. A Comparative Study Of Just-In-Time-Learning Based Methods For Online Soft Sensor Modeling. *Chemometrics and Intelligent Laboratory Systems*. 2010;104:306–317.
9. Ge Z, Huang B, Song Z. Nonlinear Semisupervised Principal Component Regression For Soft Sensor Modeling And Its Mixture Form. *Journal of Chemometrics*. 2014;28:793-804.
10. Pani AK, Mohanta HK. Online Monitoring And Control Of Particle Size In The Grinding Process Using Least Square Support Vector Regression And Resilient Back Propagation Neural Network. *ISA transactions*. 2015;56:206-221.
11. Huang GB, Chen L, Siew CK. Universal Approximation Using Incremental Constructive Feedforward Networks With Random Hidden Nodes. *IEEE Trans. Neural Networks*. 2006;17:879-892.
12. He YL, Geng ZQ, Zhu QX. Data Driven Soft Sensor Development For Complex Chemical Processes Using Extreme Learning Machine. *Chemical Engineering Research and Design*. 2015;102:1-11.
13. Shang C, Yang F, Huang D, Lyu W. Data-Driven Soft Sensor Development Based On Deep Learning Technique. *Journal of Process Control*. 2014;24:223-233.
14. Yan W, Tang D, Lin Y. A Data-Driven Soft Sensor Modelling Method Based On Deep Learning And Its Application. *IEEE Transactions on Industrial Electronics*. 2016;64:4237-4245.
15. Pedrycz W, Chen SM. *Development and Analysis of Deep Learning Architectures*. Springer. 2020.
16. Sarker IH. *Machine learning: Algorithms, Real-World Applications And Research Directions*. *SN Computer Science*. 2021;2:1-21.
17. Kotsiopoulos T, Sarigiannidis P, Ioannidis D, Tzovaras D. Machine Learning And Deep Learning In Smart Manufacturing: The Smart Gridparadigm. *Computer Science Review*. 2021;40:100341.
18. Uzair M, Shafait F, Ghanem B, Mian A. Representation Learning With Deep Extreme Learning Machines For Efficient Image Set Classification. *Neural Computing and Applications*. 2018;30; 4:1211-1223.
19. Tang J, Deng C, Huang GB. Extreme Learning Machine For Multilayer Perceptron. *IEEE Transactions On Neural Networks And Learning Systems*. 2015; 27:809–821.
20. Fujiwara K, Kano M, Hasebe S, Takinami A. Soft-Sensor Development Using Correlation-Based Just-In-Time Modelling. *AIChE Journal*. 2009;55:1754-1765.
21. Belkin M, Niyogi P, Sindhvani V. Manifold Regularization: A Geometric Framework For Learning From Labeled And Unlabeled Examples. *Journal Of Machine Learning Research*. 2006;7:2399-2434.
22. Huang G, Song S, Gupta JN, Wu C. Semi-Supervised And Unsupervised Extreme Learning Machines. *IEEE Transactions On Cybernetics*. 2014;44:2405-2417.
23. Shi J, Liu XG. Product Quality Prediction By A Neural Soft sensor Based On msa And pca. *International Journal of Automation and Computing*. 2006;3:17-22.
24. Ranzato M, Huang FJ, Boureau YL, LeCun Y. Unsupervised Learning Of Invariant Feature Hierarchies With Applications To Object Recognition,” in 2007 IEEE Conference On Computer Vision And Pattern Recognition. *IEEE*. 2007:1-8.

25. Rossi PS, Ciuonzo D, Ekman T. Hmm-Based Decision Fusion In Wireless Sensor Networks With Noncoherent Multiple Access. *IEEE Communications Letters*. 2015;19:871-874.
26. Hinton GE, Salakhutdinov RR. Reducing The Dimensionality Of Data With Neural Networks. *Science*, 2006;313: 504-507.
27. Guo T, Zhang L, Tan X. “Extreme Latent Representation Learning For Visual Classification” in *International Conference on Extreme Learning Machine*. Springer. 2018:65–75.
28. Kaneko H, Funatsu K. Nonlinear Regression Method With Variable Region Selection And Application To Soft Sensors. *Chemometrics and Intelligent Laboratory Systems*. 2013;121:26-32.
29. Ciuonzo D, Buonanno A, D’Urso M, Palmieri FA. “Distributed Classification Of Multiple Moving Targets With Binary Wireless Sensor Networks,” in *14th International Conference on Information Fusion*. IEEE. 2011;1–8.
30. Zhu J, Ge Z, Song Z. Distributed Parallel Pca For Modeling And Monitoring Of Large-Scale Plant-Wide Processes With Big Data. *IEEE Transactions on Industrial Informatics*. 2017;13:1877–1885.