International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August, 2019

Fault Identification by Extreme Pressure in Oil Pipelines using Artificial Neural Network

E.B.Priyanka, S.Thangavel, P.Parameswari, S.Ravi Sankar, N. Selva Kumar, R.Vignesh

Abstract: ***Conventional Artificial Neural Network approaches such as Feed-Forward Networks has been used in diverse applications but are not naturally predictive and also require supervised learning. Feed-forward Artificial Neural Network also trained by backpropagation poses the problem of varnishing gradient. Algorithm using Gaussian membership function with a context-decision gate for detection operations has been proposed as an alternative to the traditional Feed Forward Architecture. The AI monitoring System shows promising results in solving many recurrent problems, particularly those requiring long-term storage dependencies - the Vanishing Gradient problem (VGP) and has the ability to use contextual information when mapping between input and output sequences. The Oil monitoring system employs dynamic data flow modeling to simulate the behavior of probably militant behaviors. The contextual information (context data) includes such context as Pressure from the lab scale experimental setup of oil pipeline system. In this approach, not only the networks are trained to adapt to the given training data, the training data (the expected outputs of fault indices) is also updated to adapt to the neural network. During the training procedure, both the neural networks and training data are updated interactively. Dynamic simulations were performed using a real-time data obtained from the Radial Bias Kernel Network. The data is tested using AI system in MATLAB-SIMULINK environment to verify the performance of the proposed system. The results were promising indicating the real state of fault identification in oil pipeline system caused by extreme pressure during transportation.***

Keywords: Oil Pipelines, Pressure, Neural network.

I. INTRODUCTION

The natural resources that have been the key of life and source of economy for most countries in the world are oil, gas and water. These are transported to their destinations through the pipeline extension which spreads among all over the world. These pipelines serve as the backbone between the producers and the consumers.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
E.B.Priyanka*, Research Scholar, Department of Mechatronics Engineering, Kongu Engineering College, Perundurai, India.
S.Thangavel, Faculty, Department of Mechatronics Engineering, Kongu Engineering College, Perundurai, India.
P.Parameswari, Faculty, Department of Computer Appplication, Kumaraguru College of Technology, Coimbatore, India.
S.Ravi Sankar, UG Scholar, Department of Mechatronics Engineering, Kongu Engineering College, Perundurai, India.
N.Selava Kumar, UG Scholar, Department of Mechatronics Engineering, Kongu Engineering College, Perundurai, India.
R.Vignesh, UG Scholar, Department of Mechatronics Engineering, Kongu Engineering College, Perundurai, India.

Maintaining pipelines is very important to uphold economic growth, stability and also safety. Delays in detecting leakage on pipeline may lead to more serious effects such as fire and fatal accident. To protect and monitor pipelines there are number of technologies. Most of the technologies are designed particularly for sensing and pointing out the pipeline leakages without history of database. Pipeline monitoring systems have been using wireless devices for the communication and information transfer is very crucial because of its precious resources as well as for safety precautions. Meanwhile during oil transportations, pressure prevailing more important than other contributing parameters like flow rate, viscosity, compressor condition, pour and flash point etc. Hence neural network serves better platform for predicting and identifying faults like leakage or crack of extreme pressure values by its efficient training data set and its enhanced robust structure. The proposed approach adopts a novel approach for training neural network in ANN [3]. Different from General ANN based methods [1], the proposed approach updates/modifies the training data simultaneously as the neural network evolves. Given limited monitored data, the proposed approach is able to identify the abnormal value of pressure accurately, and allow the pressure sags at unmonitored buses being estimated [5].

II. ARTIFICIAL NEURAL NETWORK

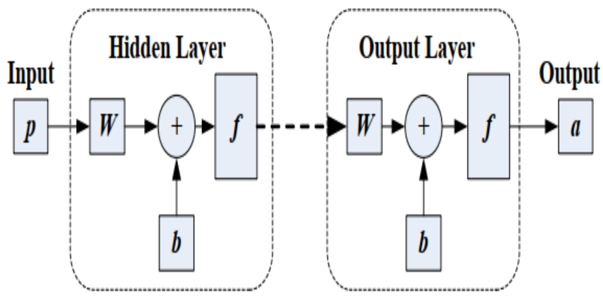

Figure 1. Structure of ANN

The structure of an ANN is illustrated in Figure. 1, in which hidden layers and an output layer are presented. For each layer there are an input vector, a weight matrix ***W***, a bias vector ***b***, a sum operator, a transfer function (TF) *f* and an output vector [4].

The weighting matrix weighs the input elements, the bias vector biases the weighed inputs via the sum operator, the sum operator gathers the weighed inputs and the biases to produce an intermediate variable for the TF, and the TF produces the final output of the layer. The output of the hidden layer is the input of the output layer. The weight matrix and biases are determined through training process which adapts the network to match the inner-pattern of the given input data [2].

A. Experimental approach of neural network to identify fault by extreme pressure

The procedure of training the network is illustrated in Figure 2. Firstly, a series of potential faults are identified and assigned with fault indices. Assuming there are *NF* faults in the test network, the faults can be named from 1 to *NF*. In step 1 in Figure 2, each potential fault is simulated to generate its corresponding pressure residuals at monitored buses. The training data consists of the monitored pressure residuals as the inputs *x* and the corresponding fault indices as the expected outputs [15, 16].

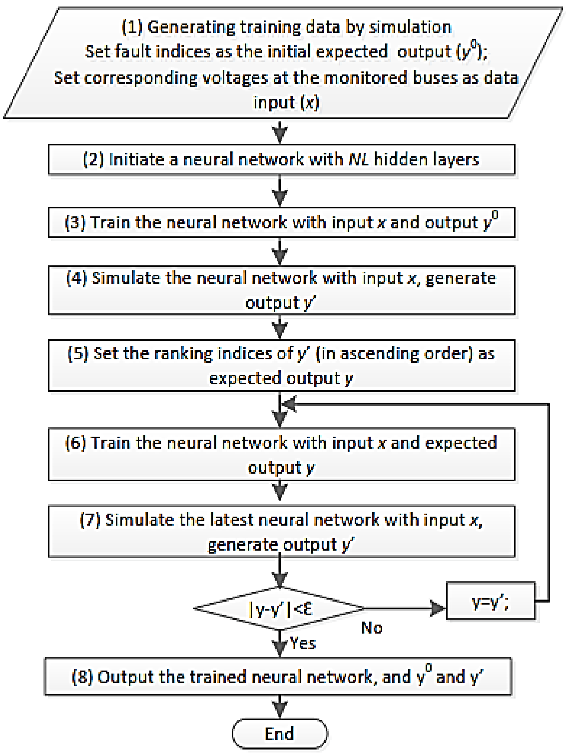

Figure 2.Procedure For Obtaining Neural Network And Fault Indices Of Pressure.

With the initialized neural network in step 2, the training data, including the input and expected outputs, are applied to train the neural network and generate an updated neural network. At this point, the updated network cannot provide satisfactory mapping between the monitored pressure residuals and original fault indices. In step 4, the trained neural network is simulated with the same inputs *x*, and generate outputs denoted as *y*'. In step 5, the fault indices are re-numbered based on the ranking of *y*' in ascending order. In this way, the fault indices are re-numbered by taking into account the magnitude of the simulated output of the trained neural network. The evolution of the fault indices are illustrated in Table 1. In generation 2 in Table 1, the fault indices *y* are updated based on the ranking of the simulated output *y*'. It can be seen that in generation 2 the updated fault indices *y* are still integer numbers. With the same inputs *x* and the updated expected fault indices *y*, the neural network is re-trained in step 6. With the same inputs *x* the latest updated neural network is simulated, and the output is denoted as *y*' in step 7. If *y*' converges to the expected output y, the iteration will terminate. Otherwise, the fault indices will be updated by setting *y*=*y*', which is again employed as the expected fault indices and used to re-train the neural network [16]. It can be seen from Table 1 that from generation 3, the fault indices are adjusted to decimal numbers rather than integer numbers. In steps 6-7 in Figure 2, the fault indices and the neural networks are updated simultaneously until the simulated outputs of the latest updated neural network are very close to the modified (expected) fault indices. Once the simulation converges, the procedure outputs the latest neural network and the modified fault indices generated in the last generation as illustrated in Table 1. The whole procedure here is to obtain the neural network and the logical numeric indices which are associated with the fault locations [6-8].

Table 1. Illustration of the evolution of fault indices during training procedure.

Gen.	Procedure	Evolution of fault indices		
		Fault 1	Fault 2	Fault 3
1	Initial Fault indices	1	2	3
2	Set fault indices as the ranking of *y*' in step 5	3	1	2
3	Set fault indices as y'	3.28	1.97	2.87
…	Set fault indices as y'	…	…	…
NG	Set fault indices as y'	2.93	1.37	1.98

III. RADIAL BIAS FUNCTION (RBF) NETWORK

RBF networks have been proven to be universal approximates and have the properties of fast convergence, easy solution to regularization, and robustness to outliers. The construction of Radial Basis Function network (RBFN) based on three layer architecture namely, an input layer is made up of source nodes that connected to the network. The second layer hidden layer with nonlinear Radial Basis activation function, such as Gaussian function that receives nonlinear transformation from the input layer to the hidden layer.

Retrieval Number E7491068519/2019©BEIESP
DOI: 10.35940/ijeat.E7491.088619
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication

The output layer implements a linear combination of Radial Basis Function that constitutes hidden layer function in a network to the activation pattern applied on it [10-13].

Step 1:- Set the weight to small random values.

Step 2:-Perform Step 2-8 when the stopping condition is false.

Step 3:- Perform Step 4-8 for each input.

Step 4:- Each input unit receives input signals and transmits to the next hidden layer unit.

Step 5:- Calculate the radial basis function.

Step 6:- Select the centers for the radial basis function. The centers are selected from the set of input vectors. It should be noted that a sufficient number of centers have to be selected to ensure adequate sampling of the input vector space.

Step 7:- Calculate the output from the hidden layer unit:

$$V_i(x_i) = \frac{\exp[-\sum(x_{ij} - \widetilde{x_{ij}})]^2}{\sigma_i^2}$$

where

$\widetilde{x_{ij}}$ = Center of the RBF unit for input variables

σ i = width of ith RBF unit

xji = jth variable of input pattern

Step 8 :- Calculate the output of the neural network :

$$Y_{net} = \sum_{i=1}^{k} W_{im} \; V_i(x_i) + W_o$$

Where

K= number of hidden layer nodes (RBF function). Ƴnet = output value of mth node in output layer for the nth incoming pattern. wim = weight between ith RBF unit and mth output node. wo = biasing term at nth output node.

Step 9: Calculate the error and test for the stopping condition. The stopping condition may be number of epochs or to a certain extent weight change.

The procedure to design the SORBF neural network for the application problems is shown below: 1. Initialize the number of RBF nodes N; 2. Initialize the position v and width d of the RBF nodes; 3. Initialize the radius r of the receptive field for each RBF node; 4. Calculate the optimum number of the RBF nodes; 5. Apply local search algorithm to adjust the position v, the width d and radius r; 6. Calculate the optimum value for the weights w. This SORBF neural network can add or reduce the RBF nodes. It is a modification of the neural network structure design method with the aim of catching the suitable RBF neural network architecture.

IV. RESULT AND DISCUSSION

Figure 3. Lab Scale Experimental Set Up Of Oil Pipeline Transport System

The original fault indices are set arbitrarily from 1 to 3 as given in the *x*-axis in Figure 4. The 3 faults like leakage, bursts and cracks in oil pipelines are simulated separately and the pressure residuals at pressure sensors 1, 2 and 3 are recorded from the experimental setup shown in Figure 3. The initial neural network is assigned with five hidden layers. The data used to train the neural network consists of the pressure residuals at monitored sensors (as inputs) and the corresponding fault indices (as the expected outputs). The procedure converges after three generations, and the adjusted fault indices are presented in Figure.4

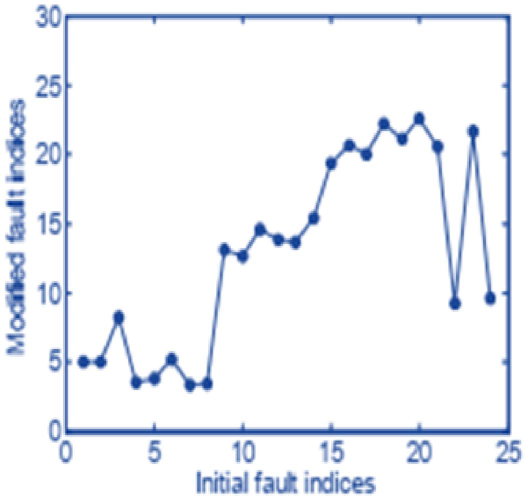

Figure 4. Initial And Adjusted Fault Indices Of Crack, Bursts And Leakage In Oil Pipelines.

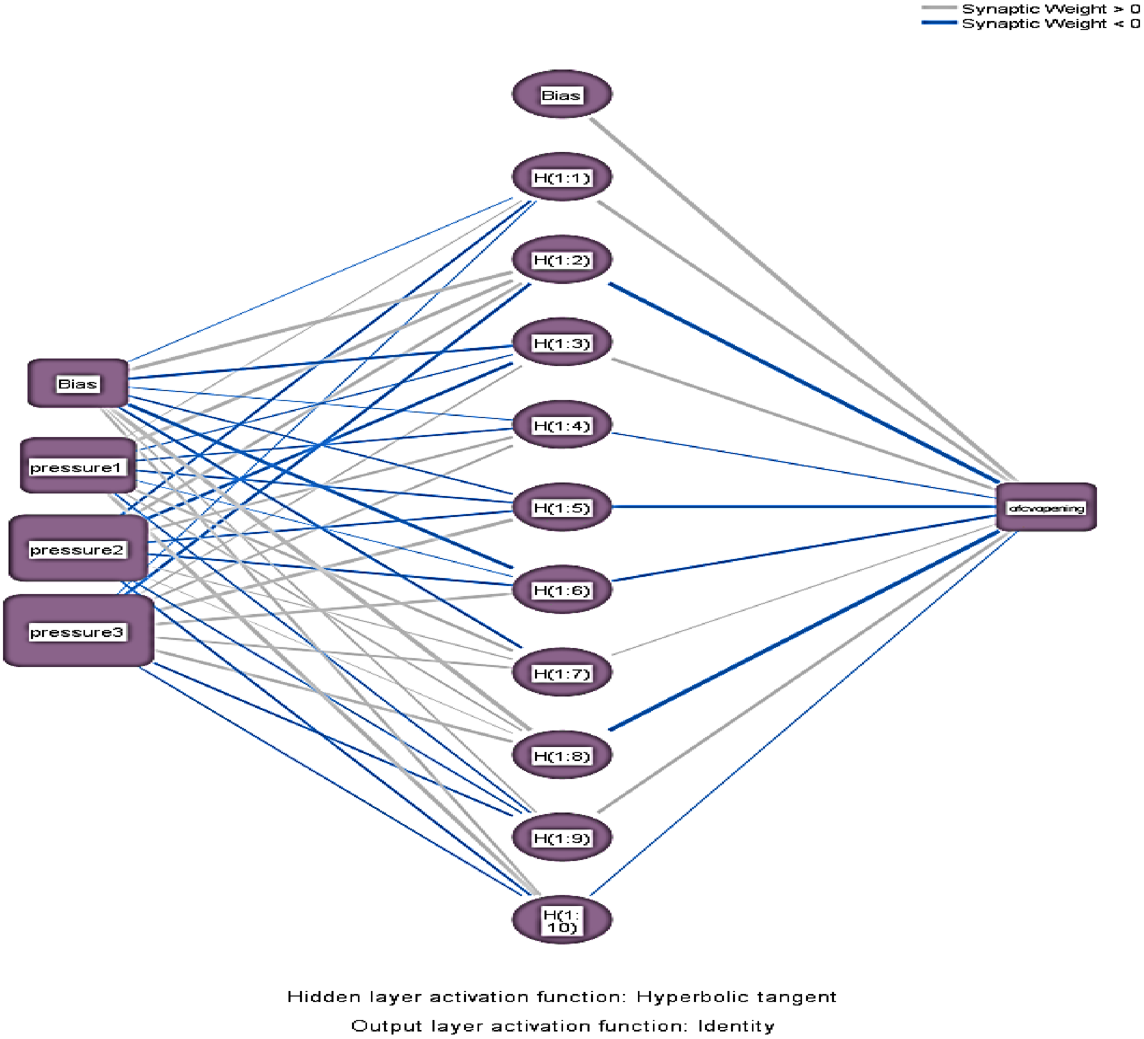

Figure 5. Neural Network Diagram Using Radial Basis Function On Pressure In Oil Pipelines.

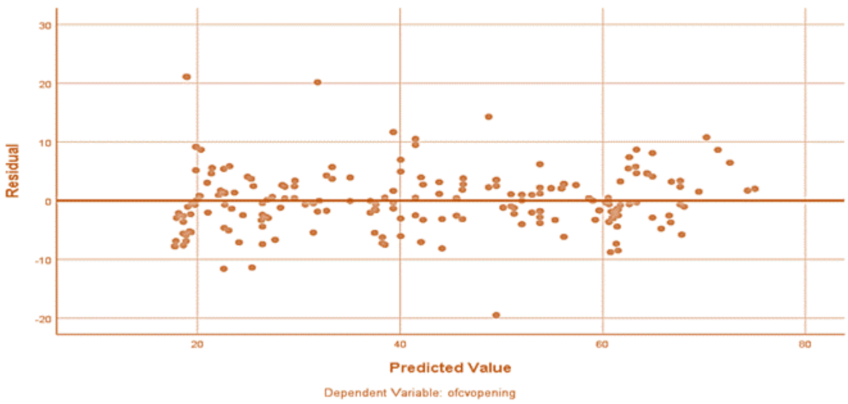

Figure 6. Consolidated Residuals Obtained By Radial Basis Kernel Function Of Pressure.

Table 2. Summary Result Regarding The Training Data Set Counts

Case Processing Summary			
		N	Percent
Sample	Training	143	71.5%
	Testing	57	28.5%
Valid		200	100.0%
Excluded		9	
Total		209	

Table 3. Network Information Of Result Of Real-Time Data Of Three Pressure Sensor Data.

Network Information			
Input Layer	Factors	1	Of cv opening
	Number of Units		59
Hidden Layer	Number of Units		10[a]
	Activation Function		Softmax
Output Layer	Dependent Variables	1	pressure1
		2	pressure2
		3	pressure3
	Number of Units		3
	Rescaling Method for Scale Dependents		Standardized
	Activation Function		Identity
	Error Function		Sum of Squares
a. Determined by the testing data criterion: The "best" number of hidden units is the one that yields the smallest error in the testing data.			

Table 4. Model Summary Based On Radial Basis Function.

Model Summary			
Training	Sum of Squares Error		60.705
	Average Overall Relative Error		.285
	Relative Error for Scale Dependents	pressure1	.285
		pressure2	.285
		pressure3	.285
	Training Time		0:00:00.69
Testing	Sum of Squares Error		40.598[a]
	Average Overall Relative Error		.451
		pressure	.451

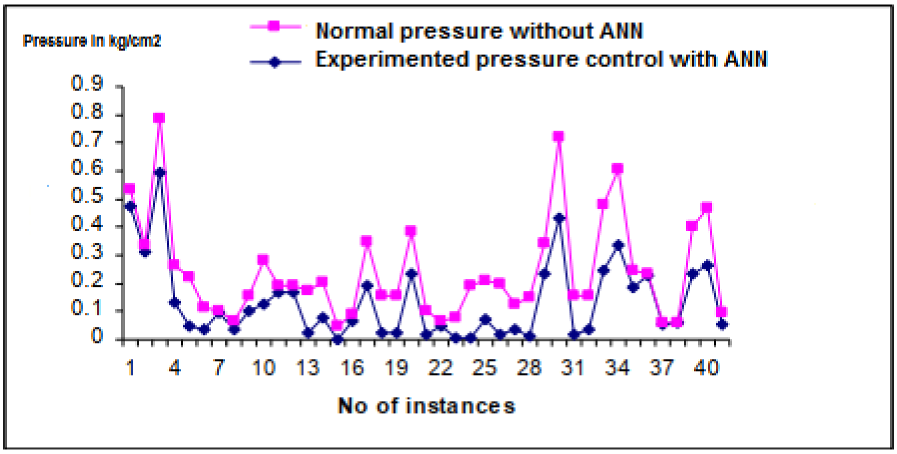

Figure 7. Pressure With ANN Results With Reduced Fault Indices.

The initializations of the number of the RBF nodes are very important, if an unsuitable initialization number of the RBF nodes is performed, the training time will be heavy, and the approximation error hard to be achieved.

The reason is that during the execution of a local search algorithm to make a fine tuning of the positions and widths of the RBF nodes, the search algorithm is hard to offset the flaw of the architecture. So, a self-organizing algorithm is necessary for the RBF structure design. This algorithm can be used to add new RBF nodes and reduce redundancy RBF nodes, which solves the problem of the architecture. Meanwhile, the position and width of the RBF nodes are very important, if the position and width are not appropriate for the RBF nodes, there is a possibility of falling into a bad local minimum. When the RBF neural network selects the correct number of the RBF nodes, the parameters of the weights, the position and width of the RBF nodes will be adjusted at the same time. The key problems of the RBF neural network are the structure design and the parameters-learning approaches. The structure design mainly relies on the RBF nodes in the hidden layer.The network structure of the three pressure sensor data are applied in radial bias function neural network structure to enumerate its training data and summary of the network information and to estimate the residuals of the consolidated three pressure sensor data obtained from the lab scale experimental setup and its statistics result generated by feeding real-time data of pressure are given in Table 2, 3 and 4. The neural network shows that ten hidden layers were incorporated and three pressure sensor data as pressure 1, 2 and 3 with bias as input and the residual with fault indices as output as shown in this figure 5 along with its synaptic weights. The results have minimized error as compared with actual value. Real measurements can be characterized by their own ranges of measurement errors which are primarily determined by the corresponding measurement devices. The uncertainty of the measurement should be considered when evaluating the performance of fault location. The deviated pressure residuals are served as inputs to the trained neural network for estimating fault indices. In this case, all 3 faults can be accurately identified with the deviated monitored pressure. Given the uncertainty of the monitored data, the simulated outputs of the neural network (i.e., the estimated fault indices) for different faults are given in Figure 7. It can be seen that even with reasonable measurement errors, the fault indices can be accurately estimated, and the fault location can be identified accurately.

V. CONCLUSION

This research work presents a fault identification and control based on extreme pressure estimation using artificial neural network. Different from general neural network training procedure in which the neural network is trained to adapt to the training data. Both the neural networks and training data are updated simultaneously during training procedure in the proposed approach of Radial basis function neural network. The fault indices numbered arbitrarily originally are adjusted to a series of numerical values which can reveal the difference of their impact on pressure residuals. Using this approach, the faults can be accurately identified using limited monitored data even when measurement uncertainty is taken into account. With this approach, the faults can be identified accurately using only one neural network, rather than employing a considerable number of matrices to register various types of faults.

CONFLICT OF INTEREST

All the contributors in this research work have no clashes of attention to announce and broadcasting this article.

ACKNOWLEDGMENT

This research work is carried out under the Senior Research fellowship received from CSIR (Council for Scientific and Industrial Research) with grant no.678/08(0001)2k18 EMR-I.

REFERENCES

1. Bradshaw, J.A., Carden, K.J., Riordan, D., 1991. Ecological ―Applications using a Novel Expert System Shell.
2. Rich , Knight and B Nair, "Artificial Intelligence", TMH Publication
3. Jacek M . Zurada, Introduction to Artificial Neural System", Jaico publishing house.
4. Ajith Abraham, "Artificial Neural Networks", Stillwater,OK, USA, 2005.
5. Carlos Gershenson, "Artificial Neural Networks for Beginners", United Kingdom.
6. G Beccari, S Caselli, F Zanichelli and et al. "Vision-based line tracking and navigation in structured environments," IEEE Int. Symposium on computational intelligence in robotics and automation, 1997, pp. 406-411.
7. Liu Z and Kleiner Y., "Computational intelligence for urban infrastructure condition assessment: Water transmission and distribution systems," IEEE Sensors Journal, vol. 14, no. 12, pp. 4122–4133, 2014.
8. Priyanka E.B, C.Maheswari, Parameter monitoring and control during petrol transportation using PLC based PID controller, Journal of Applied Research and Technology, 14 (5) (2016) 125-131.
9. Priyanka E.B, C. Maheswari, S. Thangavel, Remote monitoring and control of an oil pipeline transportation system using a Fuzzy-PID controller, Flow Measurement and Instrumentation, (2018) Article in press. https://doi.org/10.1016/j.flowmeasinst.2018.02.010.
10. P.Parameswari, Dr.M.Manikantan 2017,'Geo-Intelligence System: A Frame work for agricultural improvements', International Journal of Pure and Applied Mathemetics, vol 116, no 12, pp 117-125.
11. Parameswari, P, Abdul Samath, J & Saranya, S 2015, 'Efficient birch clustering algorithm for categorical and numerical data using modified co-occurrence method', International Journal of Applied Engineering Research, vol. 10, no. 11, pp. 27661-27673.
12. Parameswari, P, Abdul Samath, J & Saranya, S 2015, 'Scalable Clustering Using Rank Based Preprocessing Technique for Mixed Data Sets Using Enhanced Rock Algorithm', International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, no. 5, pp. 1327-1334.
13. Priyanka E.B, C. Maheswari, S. Thangavel "Remote monitoring and control of LQR-PI controller parameters for an oil pipeline transport system" Journal of Systems and Control Engineering, Article in Press, https://doi.org/10.1177/0959651818803183.
14. Jaswinder Kaur, Satwinder Singh, Dr. Karanjeet Singh Kahlon, Pourush Bassi, (2010) "Neural Network-A Novel Technique for Software Effort Estimation", International Journal of Computer Theory and Engineering, Vol. 2. No. 1, pp. 1793 – 8201.
15. Chintala Abhishek, Veginati Pavan Kumar, Harish Vitta, Praveen Ranjan Srivastava (2010) "Test Effort Estimation Using Neural Network", Journal of Software Engineering & Applications (JSEA), Vol. 3, pp. 331-340.
16. Yu-Shen Su, Chin-Yu Huang, (2007) "Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models", The Journal of Systems and Software, Vol 80, No. 4, pp. 606-615.

AUTHORS PROFILE

E.B.Priyanka received Bachelor Degree (B.E) in Electronics and Communication Engineering and completed Master Degree (M.E) in Mechatronics Engineering. Currently started Doctorate Degree (Ph.D) under Mechanical Engineering in Mechatronics Department Research Centre at Kongu Engineering College. Areas of specialization are IoT, Machine Learning, Instrumentation and control. The publication details includes 10 International conferences, 03 National Conference, 10 International Journals, all these works concerned with Industrial Automation and Process Control Stream.

S.Thangavel received Bachelor Degree (B.E) in Electrical and Electronics Engineering from Anna University, Chennai.
He completed his Master Degree (M.E) in Mechatronics Engineering with First class from Kongu Engineering College, Erode. He currently works as Assistant Professor in Mechatronics Department at Kongu Engineering College. His areas of specialization are Programmable Logic Controller, Instrumentation and control, Electrical Machines. His publication details includes 03 International conferences, 02 National Conference, 07 International Journals, all these works concerned with PLC, MEMS and Electrical Machines.

Dr.P.Parameswari is an Assistant Professor (Senior Grade) in the department of Computer Applications at Kumaraguru College of Technology, Coimbatore. She received B.Sc (Computer Science) and MCA degree from Bharathiar University and Ph.D degree in Computer Applications from Anna University, Chennai. Her research areas are Artificial Intelligence, machine learning and data science. . She has published around 20 research papers in national and international journals and conferences respectively. She has permanent membership in ISTE.