Self-Organizing Map-Based Feature Visualization and Selection for Defect Depth Estimation in Oil and Gas Pipelines

Abduljalil Mohamed, Mohamed Salah Hamdi
Information Systems Department
Ahmed Bin Mohammed Military College
Doha, Qatar
{ajmaoham, mshamdi}@abmmc.edu.qa

Sofiène Tahar
Electrical and Computer Engineering Department line
Concordia University
Montreal, Quebec, Canada
tahar@ece.concordia.ca

***Abstract*— Magnetic Flux Leakage (MFL) sensors are commonly utilized to detect defects in oil and gas pipelines and determine their depths and sizes. As a preprocessing step, MFL data are often reduced into a representative feature set that is capable of accurately estimating pipeline defect depths. However, this estimation capability may vary depending on the features used, which necessitates the need for selecting the most relevant ones. In this paper, self-organizing maps (SOMs) are used as feature visualization tool for the purpose of selecting the most appropriate features. First, a self-organizing map (SOM), i.e., a two-dimensional discretized representation of the input space of the training samples for the features, is produced. The SOM weights for each individual input feature (weight plane) are displayed then visually analyzed. Irrelevant and redundant features can be efficiently spotted and removed. The remaining "good" features (i.e., selected features) are then used as an input to a feedforward neural network for defect depth estimation. Experimental work has shown the effectiveness of the proposed approach. For instance, within ±5% error-tolerance range, the obtained estimation accuracy, using the SOM-based feature selection, is 93.1%, compared to 74% when all input features are used (i.e., no feature selection is performed); and within ±10% error-tolerance range, the obtained estimation accuracy, using the SOM-based feature selection, is 97.5%, compared to 86% when all the input features are used (i.e., no feature selection is performed).**

Keywords-data visualization; feature selection; self-organizing map; neural networks; machine learning

I. INTRODUCTION

Metallic pipelines, used for oil and gas transmission, are venerable to different metal-loss defects. As reported in [1], corrosion of underground and underwater pipelines is responsible for an interesting part of pipeline defects. Metal-loss defects are characterized by a considerable reduction in the thickness of a certain area on the pipeline surface. Under sufficient pressure, a thin area of a defected pipeline surface can cause a product leakage, which may result in incalculable costs in terms of damage to the environment and loss to human life. It has become apparent that reliable methods for detecting and localizing such metal-loss defects are greatly needed. Most of the existing methods employ magnetic flux leakage signals and ultrasonic waves to detect the presence defects such as corrosion, cracks, dents, etc. One interesting property of MFL signals is that they have a distinct signature around the center of a metal-loss defect. It has been noticed that the amplitude of MFL signals gets higher as the measuring sensors approach the defect center. Using this useful property of MFL signals, we can determine the defect's depth and length. Numerous machine learning techniques for the purpose of detecting and localizing pipeline defects are reported in the literature, in particular artificial neural networks. In [2], they are extensively trained to classify three different types of intentionally-induced pipeline defects. Experimental results show that artificial neural networks were capable of identifying normal signals from anomaly signals with classification accuracy at 94.2%. For simulated defects, they were able to classify them with accuracy at 92.5%. A fuzzy artificial neural network is proposed in [3] to assess the corrosion-vulnerability of aging pipelines. Several corrosion-related parameters are first sampled from MFL signals and then used to train the neural network to estimate the pipeline failure probability. A hybrid approach where image analysis and a neuro-fuzzy inference system are combined to classify pipe cracks is introduced in [4]. Variations in crack-related feature values are captured using fuzzy membership functions. The learning of membership function parameters is achieved through the use of a backpropagation network. In [5], an Immune Radial Basis Function Neural Network (IRBFNN)-based technique for corrosion-spot localization is presented. Discriminant corrosion properties are obtained from MFL signals and used as a feature input to train the neural network. The IRBFNN recognizes the correct location of the corrosion spots with good accuracy.

As stated earlier, amplitudes of MFL signals get higher as magnetic sensors pass right above the defect center. However, the defect-MFL shape relationship that can describe this MFL signal behavior is not known. As a result, an analytical model cannot be derived and used for estimating defect depths. This contributes handsomely to the suitability of using neural networks in these situations. The abovementioned relationship can be learned given large enough data. In this paper, we investigate the applicability of artificial neural networks in estimating defect depth based on MFL signals. Descriptive defect-related features are obtained from the raw MFL data, by using statistical feature extraction methods. Moreover, MFL signals are

1550-6037/15 $31.00 © 2015 IEEE
DOI 10.1109/iV.2015.50

approximated by using polynomial series. The coefficients of the polynomial series are then extracted and added to the feature input pattern to train the neural networks.

II. PIPELINE INSPECTION TECHNIQUES

A. Magnetic Flux Leakage-Based Inspection Technique

To monitor conditions of a pipeline, a special device equipped with magnetic sensors is dispatched into the targeted pipeline, and moves along pipeline axis. The attached sensors are arranged around the device to allow them to measure any *magnetic flux leakage signals* (MFL) on the circumference of the pipeline. This process is known as MFL scanning. The theoretical foundation of the MFL scanning is explained in detail in [6], and it is briefly introduced as follows: a magnetic field is first initiated on the surface of a pipeline, using two magnets of opposite polarity. The magnetic field basically consists of lines of magnetic force (called magnetic flux) on the pipeline surface that flow from the south pole to the north pole. When the pipeline wall has a crack or a thinning (due to corrosion, for example), then at the edges of the crack two new poles appear. However, due to air resistance, the flow of the magnetic lines deflects outwardly and creates what is known as Magnetic Flux Leakage (MFL). Any magnetic flux leakage detected by the sensors indicates the occurrence of a defect. The valuable information provided by MFL signals are exploited to verify the existence of defects and determine their locations and depths.

B. Defect Localization Using Wavelet-based Techniques

The mathematical foundation of Wavelet tools can be found with great details in [13, 14, 15]. A plethora of successful applications are reported in the literature in different domains such as high-efficiency data compression [16], data analysis and classification [17], signal de-nosing [7, 8, 9]. Hybrid approaches also utilize Wavelet techniques in detecting the presence of metal-loss defects [10]. Since MFL signals take a certain shape at the location of a metal-loss defect, Wavelet techniques can also be used to locate metal-loss defects and determine their length. This same shape occurs in a dilated form depending on the length of the defect. Let $B(x)$ denote the MFL signal measured from a pipeline. Figure 1 shows a sample MFL scan containing three defects of cuboidal shape. Each of these defects has a different length along the x-axis of the pipeline. All three components of the MFL signal (B_x , B_y , and B_z) are represented. As can be seen in Figure 1, each component of the MFL signal consists of a sum of curves, each of which is translated and dilated version of a reference pattern. If we choose the reference pattern as a mother wavelet, $\psi(x)$ and derive an orthonormal wavelet basis $\langle \psi_{j,k}(x) \rangle$ from it, then the MFL scan $B(x)$ can be expressed in the basis as:

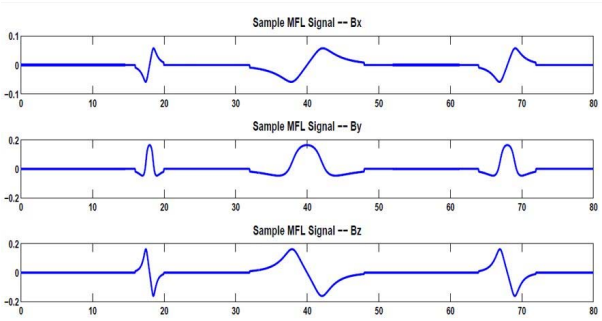

Figure 1. MFL signal signature of three metal-loss defects of cuboidal shape

$$B(x) = \sum_{j,k} c_{j,k} \psi_{j,k}(x),$$

The non-zero coefficients $c_{j,k}$ in the above representation indicate that the signal contains a copy of that particular instance of $\psi_{j,k}(x)$. To detect a metal-loss defect along the pipeline, and estimate their length, the wavelet transform of the MFL scan $B(x)$ is first computed with respect to the basis $\langle \psi_{j,k}(x) \rangle$. Next, the set of non-zero coefficients, which indicates the locations and dilation factors of the reference pattern, is determined. This in turn yields the location and width of all metal-loss defect along the pipeline

$$B(x) = \sum_{j,k} c_{j,k} \psi_{j,k}(x),$$

III. FEATURE VISUALIZATION AND SELECTION FOR DEFECT DEPTH ESTIMATION USING SOMS

The general structure of the proposed defect depth estimation approach is shown in Figure 2. It has been shown in the literature that irrelevant and redundant features can degrade the performance of most learning algorithms. One of the practical means of determining defect depth-relevant features is by utilizing data visualization. However, since the obtained input features form usually a high dimensional input vector, because their number is high, they are projected onto a lower-dimensional space, where it is easier to interpret them and gain useful information about their correlation. We use the self-organizing map (SOM) approach to project the multi-dimensional input vector space (consisting of the obtained features) into a two-dimensional map.

From the obtained SOM, the weights corresponding to each individual input feature are displayed as a weight plane. The weight planes corresponding to the different features are then analyzed visually in order to detect correlations between them and select the appropriate feature set that will be used as input to a neural network based approach for defect depth estimation.

A. Self-Organizing Map Neural Network

In 1990, T. Kohonen introduced the self-organizing map (SOM) as an unsupervised neural network learning algorithm [18]. In the literature, SOMs have been used in different application domains [19, 20, 21], and more important, they are also widely used in visualization as a dimension (feature) reduction tool [22, 23, 24]. A two-dimensional SOM basically consists of a two-dimensional lattice of neurons. The aim of the iterative training strategy of the SOM is to classify the input data elements into several categories such that similar data elements fall in the same category. Moreover, network weights for each input feature component can be visualized and examined. This useful property of the SOM makes it an appealing data visualization technique that will be effectively exploited in this work to identify the redundant features. Each neuron, n_i in the network is represented by an n-dimensional weight vector:

$$w_i = [w_{i1}, w_{i2}, ..., w_{in}],$$

The weight vectors are randomly initialized. Each neuron in the map is connected to its immediate neighbors. Given a randomly selected sample vector X from the input data, all the weight vectors of the SOM are calculated using some distance measure such as the Euclidian distance. The wining neuron, called the best matching unit (BMU), is the neuron whose weight vector is closest to X. The weight vectors of the wining neuron, along with its topological neighbors, are updated so that they are moved closer to the input vector in the input space. The new weight vectors are updated as follows:

$$w_i(t+1) = w_i(t) + \alpha(t).h(n_{BMU}, n_i, t)(X - w_i(t))$$

the neighborhood function, $h(n_{BMU}, n_i, t)$, is monotonically decreasing with respect to the distance between the wining neuron and the neuron n_i in the network grid and the time t. The purpose of the learning rate parameter, $0 \le \alpha(t) \le 1$, is to control the adjustments of the weight vectors according to the iterative step t. Initially, it is set to a large value, and then it decreases over time.

B. Feature Extraction

In order to efficiently train neural networks, MFL signals are first transformed into a smaller subset of representative features. Moreover, extracting signal-descriptive features may actually enhance the network performance in terms of estimation accuracy of defect depths. Thus, five statistical features including maximum magnitude, peak-to-peak distance, integral of the normalized signal, mean average, and standard deviation are obtained. To expand the feature input, we also approximate MFL signals by using polynomial series of the form $a_n X^n + ... + a_1 X + a_0$, and consider their coefficients, $a_n + ... + a_0$ as features. The best approximations for the MFL components are found to be degree three for the B_x component, and degree six for both B_y and B_z components. We will refer to the thirty-three obtained features by F1, F2, …, and F33.

C. SOM-based Feature Selection

To improve the defect depth estimation, the most relevant features are selected, and redundant features should be eliminated. As shown in Figure 2, feature selection is realized by the SOM approach. The features mentioned in the previous section will be used to train the self-organizing network. Four features, however, yield very small values and are deemed inappropriate by the Matlab neural network tool. Thus, these features (F3, F6, F8, and F13) are removed from the feature input pattern. The network structure is shown in Figure 3. The SOM network consists of a competitive layer which can classify the 29-input data vectors into up to 100 classes (the number of the neurons in the output layer arranged in a 2D topology: 10x10). The network is trained with 1357 data samples using the SOM batch algorithm. The network hits are shown in Figure 4, in which the number displayed in each neuron indicates how many of the data samples are associated with that neuron. For example, fifty-five data samples are associated with the neuron located at row 8 and column 6. The SOM is visualized by displaying weigh planes in Figure 5. The figure shows a weigh plane for each of the 29 features; where darker colors represent larger weights. Upon closer inspection, it is noticeable that several features, namely, F4, F11, F12, F13, F14, F15, F23, F26, and F27, exhibit similar connection patterns. These nine features are considered highly correlated and thus eight of them should be removed from the feature set. The remaining twenty one features constitute the new input feature pattern and are used to train the feedforward Neural Network (NN) for defect depth estimation.

IV. PERFORMANCE EVALUATION

In this section, the performance of the proposed approach is evaluated for three different feedforward network (FFNN) structures, namely static, cascaded, and dynamic. The main performance measure is the estimation accuracy of the failure depth within a certain level of error-tolerance. The error-tolerance levels used in this study are ±1%, ±5%, ±10%, ±15%, ±20%, ±25%, ±30%, ±35%, and ±40%. For each network structure, the FFNN is experimented with different numbers of hidden layers, each varies in size from 10 neurons up to 100 neurons. The mean squared error (MSE) performance function:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{Y}_i - Y_i\right|^2,$$

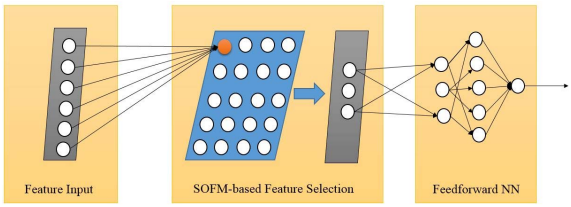

Figure 2. Feedforward NN-based defect depth estimation

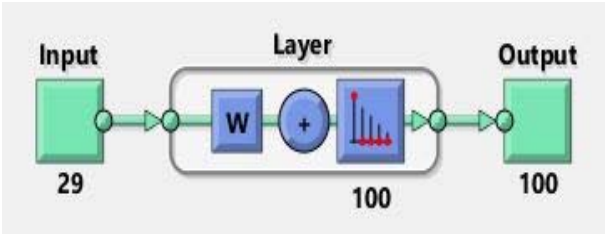

Figure 3. SOM structure

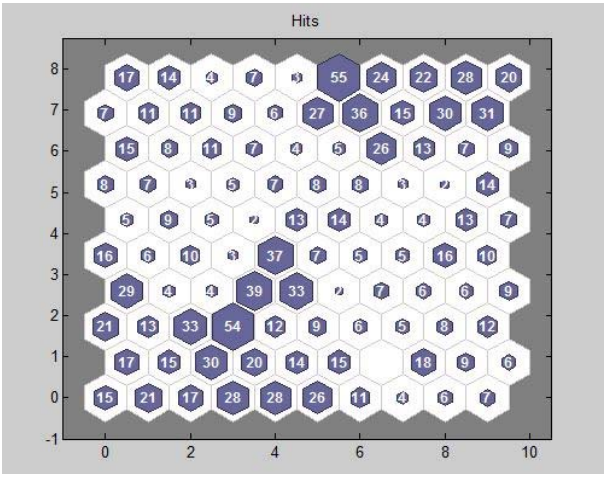

Figure 4. SOM sample hits. The number shown in each neuron indicates the number of samples associated with that neuron.

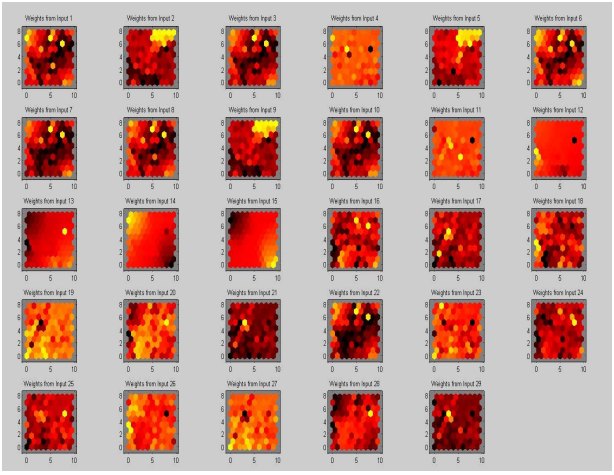

Figure 5. SOM weights for each input feature

and the log-sigmoid transfer function:

$$\log sig = \frac{1}{(1-e^{-x})},$$

are used in the experimental work. The results of the experimental work are reported in the following subsections.

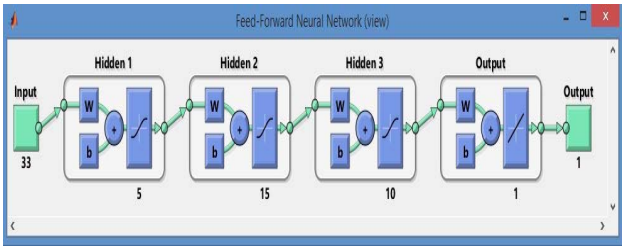

Figure 6. Architecture of the static FFNN

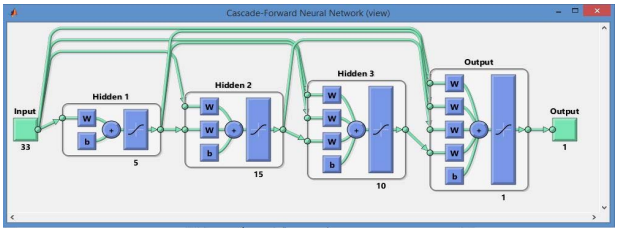

Figure 7. Architecture of the cascaded FFNN

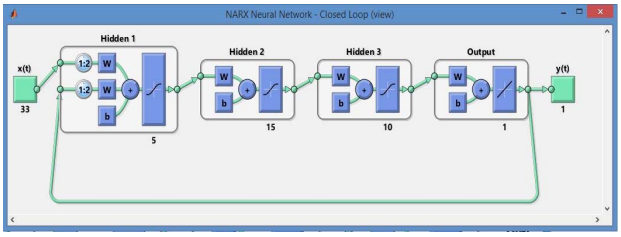

Figure 8. Architecture of the dynamic FFNN

A. *Complete Feature Set*

The complete feature set (33-input features) are first used to train the static (Figure 6), cascaded (Figure 7), and dynamic (Figure 8) FFNN. Table I shows the estimation accuracy for each network structure with different numbers of hidden layers.

B. *Selected Feature Set*

The 21 features selected by the SOM approach are used to evaluate the performance of the three FFNN structures. The results for the static, cascaded, and dynamic networks are shown in Table II, Table III, and Table IV, respectively.

C. *Discussion*

For the complete feature set, it should be noted from table I that dynamic networks with a single hidden layer yield the best performance results for error-tolerance levels of ±1%, ±5%, ±10%, ±15%, and ±20% at 23%, 74%, 86%,

TABLE I.
BEST ESTIMATION ACCURACY OF STATIC, CASCADED, DYNAMIC FFNN USING 33 FEATURES

Error-Tolerance	Feed Forward Network Architecture											
	Static				Cascaded				Dynamic			
	Hidden Layers				Hidden Layers				Hidden Layers			
	1	2	3	4	1	2	3	4	1	2	3	4
±1%	0.11	0.11	0.09	0.08	0.07	0.06	0.06	0.06	0.23	0.22	0.23	0.40
±5%	0.47	0.41	0.38	0.40	0.31	0.30	0.25	0.25	0.70	0.68	0.63	0.74
±10%	0.67	0.64	0.66	0.70	0.49	0.60	0.43	0.45	0.86	0.84	0.79	0.83
±15%	0.81	0.77	0.75	0.81	0.65	0.72	0.63	0.62	0.89	0.89	0.85	0.86
±20%	0.87	0.84	0.82	0.88	0.75	0.78	0.75	0.74	0.90	0.89	0.89	0.91
±25%	0.92	0.87	0.88	0.90	0.81	0.83	0.82	0.80	0.91	0.89	0.91	0.93
±30%	0.94	0.90	0.92	0.93	0.85	0.85	0.86	0.86	0.92	0.90	0.92	0.95
±35%	0.95	0.92	0.93	0.94	0.88	0.91	0.89	0.89	0.92	0.90	0.92	0.96
±40%	0.95	0.94	0.93	0.95	0.90	0.93	0.90	0.91	0.92	0.90	0.92	0.96

TABLE II.
BEST ESTIMATION ACCURACY OF STATIC FFNN USING 21 FEATURES

	Number of Neurons									
	10	20	30	40	50	60	70	80	90	100
±1%	0.0931	0.0784	0.1078	0.1029	0.1029	0.1520	0.1029	0.1078	0.1078	0.0735
±5%	0.4265	0.4363	0.4216	0.4265	0.3676	0.4363	0.4461	0.4314	0.3971	0.4314
±10%	0.6863	0.7059	0.7059	0.6618	0.6618	0.7353	0.6667	0.6765	0.6422	0.6520
±15%	0.8578	0.7892	0.8382	0.8039	0.7745	0.8235	0.8431	0.8725	0.8039	0.7696
±20%	0.9167	0.8627	0.9020	0.8676	0.8529	0.9118	0.9167	0.9265	0.8971	0.8186
±25%	0.9510	0.9216	0.9265	0.9118	0.9020	0.9412	0.9461	0.9510	0.9314	0.8873
±30%	0.9657	0.9510	0.9412	0.9363	0.9412	0.9510	0.9706	0.9755	0.9657	0.9167
±35%	0.9706	0.9804	0.9412	0.9608	0.9608	0.9559	0.9853	0.9902	0.9706	0.9314
±40%	0.9853	0.9804	0.9608	0.9706	0.9853	0.9804	0.9853	0.9902	0.9853	0.9510

TABLE III.
BEST ESTIMATION ACCURACY OF CASCADED FFNN USING 21 FEATURES

	Number of Neurons									
	10	20	30	40	50	60	70	80	90	100
±1%	0.0588	0.0882	0.0784	0.0784	0.0588	0.0637	0.1078	0.0735	0.0784	0.0686
±5%	0.3333	0.2843	0.3137	0.3088	0.2647	0.3480	0.3725	0.3431	0.2598	0.3088
±10%	0.5833	0.5294	0.5343	0.5343	0.5441	0.6029	0.5833	0.6176	0.5392	0.5784
±15%	0.7402	0.7402	0.7206	0.7010	0.7059	0.7500	0.7402	0.7402	0.6716	0.7255
±20%	0.7941	0.8235	0.8186	0.7843	0.7794	0.8186	0.8186	0.7941	0.7353	0.8137
±25%	0.8676	0.8676	0.9118	0.8971	0.8873	0.8824	0.8873	0.8824	0.8284	0.8725
±30%	0.8922	0.8922	0.9216	0.9069	0.8971	0.9167	0.9118	0.9167	0.8480	0.9069
±35%	0.9412	0.9216	0.9363	0.9363	0.9559	0.9657	0.9412	0.9657	0.9069	0.9510
±40%	0.9510	0.9412	0.9461	0.9461	0.9559	0.9657	0.9510	0.9804	0.9216	0.9559

TABLE IV.
BEST ESTIMATION ACCURACY OF DYNAMIC FFNN USING 21 FEATURES

	Number of Neurons									
	10	20	30	40	50	60	70	80	90	100
±1%	0.4187	0.2857	0.3695	0.3202	0.1379	0.2069	0.2365	0.1429	0.1823	0.1527
±5%	0.8916	0.8670	0.9310	0.8867	0.5123	0.7143	0.6700	0.5222	0.5961	0.5665
±10%	0.9458	0.9754	0.9704	0.9704	0.7635	0.9212	0.8621	0.7340	0.7783	0.7488
±15%	0.9754	0.9803	0.9852	0.9803	0.8768	0.9754	0.9360	0.8473	0.8670	0.8473
±20%	0.9803	0.9852	0.9852	0.9852	0.9310	0.9852	0.9507	0.9015	0.9113	0.8916
±25%	0.9803	0.9852	0.9852	0.9901	0.9754	0.9901	0.9606	0.9409	0.9409	0.9163
±30%	0.9803	0.9852	0.9852	0.9901	0.9754	0.9951	0.9852	0.9458	0.9557	0.9409
±35%	0.9852	0.9852	0.9901	0.9901	0.9803	0.9951	0.9951	0.9606	0.9655	0.9507
±40%	0.9852	0.9852	0.9951	0.9901	0.9852	0.9951	1.0000	0.9704	0.9704	0.9704

89%, and 90% estimation accuracies, respectively. Moreover, dynamic networks with 4 hidden layers yield the best performance for error-tolerance levels of ±25%, ±30%, ±35%, and ±40%, at 91%, 93%, 95%, 96%, and 96% estimation accuracies, respectively. Cascaded networks, however, have performed the worst for error-tolerance levels of ±1%, ±5%, ±10%, ±15%, and ±20%, at 7%, 4%, 60%, 72%, and 78% estimation accuracies, respectively. At other error-tolerance levels, they yield comparable results. Static networks performed better than cascaded networks but less than dynamic networks. We observed that increasing the number of hidden layers has not necessarily improved the performances of the networks. With the exception of dynamic networks (with 4 hidden layers, and for the error-tolerance levels ±1% and ±5%), it has actually reduced the overall performance of the feed-forward neural networks.

Based on the above discussion, for the selected feature set, we only focus on networks with a single hidden layer with different number of neurons ranging from 10 to 100. Based on the results reported in Tables II, III, and IV, it is obvious that the networks using the selected features yield a superior performance compared to those using the whole feature set. The best defect depth estimation accuracies for each network configuration are highlighted in red. It is also noted that the dynamic networks yield the best performance results for error-tolerance levels of ±1%, ±5%, ±10%, ±15%, and ±20% at 41%, 93%, 97%, 98%, and 98% estimation accuracies, respectively. Cascaded networks, however, have performed slightly better than their corresponding networks using the complete feature set, for error-tolerance levels of ±1%, ±5%, ±10%, ±15%, and ±20%, at 10%, 37%, 61%, 75%, and 78% estimation accuracies, respectively. As expected, static networks performed better than cascaded networks but less than dynamic networks. For the purpose of comparison, the weight-correlation-based feature selection method [25] is utilized to assign weights for each feature. Features with the highest weights are then selected to estimate the defect depths. The estimation accuracies for error-tolerance levels of ±1%, ±5%, ±10%, ±15%, and ±20%, obtained using this method, are 0.0637, 0.3529, 0.6225, 0.8039, and 0.8775, respectively. Clearly, the performance of the SOFM is superior to that of the weight-correlation, particularly for the dynamic networks.

Although the self-organizing sematic map was reported to be more robust and better suited than principal component analysis (PCA) [26, 27] for some applications [28, 29, 30], our aim in future work is to examine the applicability of PCA as a feature selection technique to identify and remove redundant features that are degrading the ANN performance in estimating defect depths.

V. CONCLUSIONS

The use of self-organizing maps for feature visualization for the purpose of feature selection for estimating pipeline defect depths is investigated. Extensive experimental work using different levels of error-tolerance for different network

structures has been conducted. It has been shown that dynamic networks, using the proposed feature selection scheme, yield superior performance compared to that of their corresponding networks that use the complete feature set at 93% and 97% defect depth estimation accuracy within ±5% and ±10% of error-tolerance, respectively. We intend in future work to obtain more sophisticated features and examine other visualization tools. Moreover, principal component analysis will be investigated as a feature selection tool.

ACKNOWLEDGMENT

This work was made possible by NPRP grant # [5-813-1-134] from Qatar Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the authors.

REFERENCES

[1] P. Nicholson, "Combined CIPS and DCVG survey for more accurate ECDA data," Journal of World Pipeline, vol. 7, pp. 1-7, 2007.

[2] A. Carvalho, et al, "MFL signals and artificial neural networks applied to detection and classification of pipe weld defects," Ndt & E International, vol. 39(8), pp. 661-667, 2006.

[3] S. Sinha and M. Pandey, "Probabilistic neural network for reliability assessment of oil and gas pipelines," Computer-Aided Civil and Infrastructure Engineering, vol. 17(5), pp. 320-329, 2002.

[4] S. K. Sinha and F. Karray, "Classification of underground pipe scanned images using feature extraction and neuro-fuzzy algorithm," IEEE Transactions on Neural Networks, vol. 13(2), pp. 393-401, 2002.

[5] M. Zhongli and H. Liu, "Pipeline defect detection and sizing based on MFL data using immune RBF neural networks," IEEE Congress on Evolutionary Computation, pp. 3399-3403, 2007.

[6] K. Mandal and D. L. Atherton, "A study of magnetic flux-leakage signals," Journal of Physics D: Applied Physics, vol. 31(22), pp. 3211, 1998.

[7] S. Shou-peng and Q. Pei-wen, "Wavelet based noise suppression technique and its application to ultrasonic flaw detection," Ultrasonic, vol. 44(2), pp. 188-193, 2006.

[8] A. Muhammad and Satish Udpa, "Advanced signal processing of magnetic flux leakage data obtained from seamless gas pipeline," Ndt & E International, vol. 35(7), pp. 449-457, 2002.

[9] S. Mukhopadhyay and G. P. Srivastava, "Characterization of metal loss defects from magnetic flux leakage signals with discrete wavelet transform."Ndt & E International, vol. 33(1), pp. 57-65, 2000.

[10] K. Hwang, et al, "Characterization of gas pipeline inspection signals using wavelet basis function neural networks," NDT & E International, vol. 33(8), pp. 531-545, 2000.

[11] K. Levenberg, "A method for solution of certain problems in least squares," Quart. Applied Math, vol. 2, pp. 164-168, 1944.

[12] D. W. Marquardt, "An algorithm for least squares estimation of nonlinear parameters," SIAM Journal on Applied Mathematics, vol. 11(2), pp. 431-441, 1963.

[13] I. Daubechies, Ten Lectures on Wavelets. SIAM: Society for Industrial and Applied Mathematics, 1992.

[14] S. Mallat, A Wavelet Tour of Signal Processing. Academic Press, 2008.

[15] M. Misiti, Y. Misiti, G. Oppenheim, and J. M. Poggi, Wavelets and their Applications. Wiley-ISTE, 2007.

[16] J. N. Bradley, C. M. Brislawn, and T. Hopper, "FBI wavelet/scalar quantization standard for gray-scale fingerprint image compression," In: SPIE Procs, Visual Information Processing II, vol. 1961, pp. 293–304, 1993.

[17] M. Unser and A. Aldroubi, "A review of wavelets in biomedical applications," Proceedings of the IEEE, vol. 84(4), pp. 626–638, 1996.

[18] T. Kohonen, "The self-organizing map," Proceedings of the IEEE, vol. 78(9), pp. 1464-1480, 1990.

[19] G. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," Computer vision, graphics, and image processing, vol. 37(1), pp. 54-115, 1987.

[20] S. Becker and G. Hinton, "Self-organizing neural network that discovers surfaces in random-dot stereograms," Nature, vol. 355(6356), pp. 161-163, 1992.

[21] A. Gabor, R. Leach, and F. Dowla, "Automated seizure detection using a self-organizing neural network," Electroencephalography and clinical Neurophysiology, vol. 99(3), pp. 257-266, 1996.

[22] M. Hamdi, "SOMSE: A semantic map based meta-search engine for the purpose of web information customization," Applied Soft Computing, vol. 11(1), pp. 1310-1321, 2011.

[23] H. Chen, C. Schuffels, and R. Orwig, "Internet categorization and search: a machine learning approach," Journal of Visual Communications and Image Representation, vol. 7 (1), pp. 88-102, 1996.

[24] R. Orwig, H. Chen, and J.F. Nunamaker, "A graphical, self-organizing approach to classifying electronic meeting output," Journal of the American Society for Information Science, vol. 48 (2), pp. 157–170, 1997.

[25] M. Hall, Correlation-based feature selection for machine learning. PhD diss. The University of Waikato, 1999.

[26] H. Hotelling, "Analysis of a complex of statistical variables into principal components", Journal of Educational Psychology, vol. 24, pp. 417-441, 1933.

[27] H. Hotelling, "Relations between two sets of variates," Biometrika, vol. 27, pp. 321-377, 1936.

[28] A. Suwardi, K. Takenori, and S. Koyama, "Principal component analysis and self-organizing map for visualizing and classifying fire risks in forest regions, agricultural information research," Atoms Environ, vol. 16(2), pp. 44-51, 2007.

[29] L. Peeters and A. Dassargues, "Comparison of Kohonen's self-organizing map algorithm and principal component analysis in the exploratory data analysis of a groundwater quality dataset," GeoENV, 2006.

[30] D. Reusch, R. Alley, and B. Hewitson, "Relative performance of self-organizing maps and principal component analysis in pattern extraction from synthetic climatological data," Polar Geography, vol. 29(3), pp. 188-212, 2005.