Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.8, No.5, 2017

Adaptive Cooperative Learning Methodology for Oil Spillage Pattern Clustering and Prediction

Udoinyang G. Inyang[1] Oluwole C. Akinyokun[2] Moses E. Ekpenyong[2]
1.Department of Computer Science, Faculty of Science, University of Uyo, Uyo, Nigeria
2.Department of Computer Science, Federal University of Technology, Akure, Nigeria

Abstract
The serious environmental, economic and social consequences of oil spillages could devastate any nation of the world. Notable aftermath of this effect include loss of (or serious threat to) lives, huge financial losses, and colossal damage to the ecosystem. Hence, understanding the pattern and making precise predictions in real time is required (as opposed to existing rough and discrete prediction) to give decision makers a more realistic picture of environment. This paper seeks to address this problem by exploiting oil spillage features with sets of collected data of oil spillage scenarios. The proposed system integrates three state-of-the-art tools: self organizing maps, (SOM), ensembles of deep neural network (k-DNN) and adaptive neuro-fuzzy inference system (ANFIS). It begins with unsupervised learning using SOM, where four natural clusters were discovered and used in making the data suitable for classification and prediction (supervised learning) by ensembles of k-DNN and ANFIS. Results obtained showed the significant classification and prediction improvements, which is largely attributed to the hybrid learning approach, ensemble learning and cognitive reasoning capabilities. However, optimization of k-DNN structure and weights would be needed for speed enhancement. The system would provide a means of understanding the nature, type and severity of oil spillages thereby facilitating a rapid response to impending oils spillages.
Keywords: SOM, ANFIS, Fuzzy Logic, Neural Network, Oil Spillage, Ensemble Learning

1. Introduction

Oil pollution is one of the main problems affecting coastal and marine spaces of most underdeveloped economies of the world, especially those involved in oil exploitation and explorations. It affects greater proportion of coastal (river systems, lakes) and terrestrial habitats daily, particularly in areas where substantial oil drilling, refining, transportation and marketing activities are carried out (Chang et al., 2014). In addition, storage and consumption of crude oil or its refined products are sources of oil related pollutants (Iwejingi, 2013). Moreso, spills resulting from oil production enhancements (such as drilling mud and fluids etc.) also occur during oil drilling process. Oil pipelines are media for transporting and marketing (distribution and delivery) of significant proportion of crude oil and other oil products within a country (Wilson, 2001). However, since defects and damages to oil pipelines may exist undetected over a long period; the repair of such faults takes a longer period while oil spill persists with attendant significant impact. Oil spillages and hydrocarbon emissions may also arise from accidents, involving tankers. Other causes of oil spillages include sabotage, oil terrorism, vandalism and theft through oil siphoning, availability and application of ineffective technologies and procedures for the prevention and mitigation of oil spillages, and the use of obsolete infrastructures incapable of coping with current challenges, equipment failure and human error (Egwu , 2012; Anderson and LaBelle, 2000; Adelana, *et al.,* 2011). In most developing economies, the absence of contingency plans for real-time response, poor detection, and mitigation guidelines increase the severity of oil spillages. Oil spillage has significant short-term and long-term impacts on the environment and ecology and to the socio-economic activities of the affected area. In other words, it causes great damage to the soil, plants and animals as well as serious human hazards, hostilities between host communities and oil industries, and disruption of economic, social activities, security and well-being of people in the affected areas (Egwu, 2012). Despite these severe impacts on ecosystems, the dependency on oil and gas, oil exploitation, exploration, transportation and marketing has significantly increased. Moreso, petroleum products (crude oil and its refined products) have sustained their role as major and dependable sources of fuel and income in many economies of the world. This has increased the frequency and magnitude of oil pollution and the adverse effect on lives and properties, thereby requiring effective management of the attendant risks.

Management of oil spillages are very demanding because of its complexity, dynamism, magnitude of its occurrences, and varying characteristics of contaminated areas, source and type of spilled oil. Oil pollutants' properties, hydrological conditions, nature of spillage, and diverse physical, biological and chemical processes are characteristics of the location of spill and the affected areas. Thus, suitable approaches for cleaning affected areas vary significantly and often require expertise on anticipation, preparedness, mitigation technologies, and environmental conditions (Geng, et al., 2001). In Santos et al., (2013), Akinyokun and Inyang, (2013) different approaches have been used to visualize, classify, or respond to emergencies resulting from oil spillages and leakages. These systems lack systematic ways of monitoring the time-variant pipeline activities, high probability

of false detection alarm and adopt inefficient approaches and tools for localization of abnormal activities due to non-inclusion of intelligent tools for explicit timing of operations, pattern recognition and uncertainty handling. A requirement for managing emergencies, arising from natural or human induced hazard, is an in-depth body of knowledge and information — understanding the nature, sources, impact and response strategies for preventing of their occurrences and mitigation impacts, as well as result-oriented responders in technical, coordination and cooperation aspects (Hernandez & Serrano, 2001, Akinyokun and Inyang 2013). Kao *et al.,* (2008) affirms that a vital task in a knowledge-driven society is the transformation of data held in data repositories into information and knowledge. This requirement can be achieved through knowledge discovery (KD) techniques — the process of identifying and extracting vital relationships and patterns within datasets, and summarizing the data and knowledge for enhanced comprehensibility, and a more efficient utilization (Mosley, 2005). KD, a multidisciplinary approach, unifies diverse approaches (statistics, mathematics, optimization and machine learning) for the discovery and translation of the desired knowledge into more comprehensible forms (patterns, rules, procedures and models), useful for problem solving and decision-making. KD has data mining (DM) as the most important step. DM tools include artificial neural networks (ANN), fuzzy logic (FL), support vector machines, ants colony algorithm (ACO), genetic algorithms, particle swarm and so on. ANNs are widely applicable in applications that require classification, prediction or forecasting (Gurney and Gurney, 1997). A type of ANN is self-organizing map (SOM) — a single layered ANN that implements unsupervised learning to produce a two-dimensional, discrete representation of the input space. SOMs are different from other ANNs since they adopt neighborhood function to preserve the properties of the input data-space topology. They are commonly used for visualizing, clustering of multidimensional data and dimension reduction and widely applied in areas like medicine, financial, ecological, military, engineering, security, forecasting and so on (Stefanovi and Kurasova, 2011). The combination of multiple intelligent tools has proven a promising approach to enhancing consistency, accuracy and generalization capabilities of classifiers.

Ensemble approach is one of the most efficient scientific methodologies for obtaining accurate, robust, better generalization and satisfactory results (Zainal *et al.,* 2009). Ensemble systems integrate two or more intelligent tools, and take advantage of the strengths of each technique while the weaknesses of each technique are compensated for or in most cases resolved. Several works have successfully proved the effectiveness of hybridization of ANN and FL in oil related applications. The unification of ANN and FL produce Adaptive Neuro Fuzzy Inference System (ANFIS) and applied in the prediction of oil spillages (Akinyokun and Inyang 2013, Inyang and Akinyokun, 2014). Several membership functions (MF) were created within an ANFIS in the classification and prediction of oil prices (Lima *et al.,* 2002), ensemble of ANFIS tested in four partitions of training and testing datasets in the accurate prediction of oil prices (Gabralla, et al., 2014). Ensemble learning has been applied to many areas and applications such as DM and medical applications and several others (Gabralla, *et al.,* 2014). This paper, proposes an ensemble of SOM, multiple DNNs and ANFIS in the prediction and classification of oil spillage patterns. The rest of the paper is organized as follows; Overview of components in the proposed ensembles is described in Section 2. In Section 3, system design and methodological workflow are presented. Section 4 describes the series of experiments and the analysis of results. Finally, conclusions are drawn in Section 5.

2. Overview of Proposed Ensemble Components

Ensemble methods implement multiple learning algorithms that construct a set of classifiers for the purpose of classifying new data points. Ensembles have been widely used and successfully applied in different fields, including decision-making, classification, and medical diagnosis and so on, owing to its versatile characteristics (Lei and Wan, 2012). Ensembles of deep neural network (DNN) involve a pool of individual DNNs designed to cooperate and complement aspects of the solution. Such ensembles integrate knowledge base embedded in NNs, and have frequently produced more accurate and robust models. The goal of ensemble learning methods is to construct a collection or hybrid of classifiers that are diverse and yet accurate (Lei and Wan, 2012). The components of the ensembles described in this paper are SOM, DNN and ANFIS .

2.1 SOM Topology and Clustering Procedure

SOM is a powerful and efficient unsupervised visualization tool with the capability of simultaneously performing clustering and projection operations without altering the vital data structures found in the dataset (Yao et al., 2014; Kohonen, 2013). The possession of rich visualization capabilities makes SOM efficient for configuration, exploration and clustering of highly dimensional dataset. SOMs are widely used for clustering, exploratory analysis and visualization tasks, data sampling and classification, feature extraction and data interpolation tasks (Lobo, 2009). A SOM network is a single-layered ordered network, consisting of series linearly connected neurons in a two-dimensional rectangular or hexagonal grid topology. Each connection to a neuron has assigned weights from the codebook vector, wij (i = 1,2…p; j = 1,2,…,q) where p and q represents the number of rows and columns respectively, and must be the same as the input dimension (Stefanovi˘c and

Kurasova, 2011). The unsupervised learning process of SOM which depends on vector adaptation, is performed in the following steps (Bação, et al. 2005, Amiryousefi, 2011; Stefanovic and Kurasova, 2011, Gorman, et al., 2015)

- randomly initialize each weight (w_{ij}) of the neurons in the output layer;
- for each training iteration (t), randomly or sequentially select from the training dataset an input vector, $X_p = \{X_1, X_2, ..., X_m\}$ (where *m* is the number of data samples). This input vectors form the nodes on the SOM network.
- for each node in the SOM, calculate the similarity between the each input vector and all neurons using the Euclidean distance metric ($d_{ik} = \|X_p - w_{ij}\|$).
- find the neuron with the highest similarity or minimum Euclidean distance or best match —the winning neuron (h_{winner})
- adapt each neuron using the learning rule: $w_{ij}(t+1) = w_{ij}(t) + \alpha(t)(h_{winner}(t))\|X_p - w_{ij}(t)\|$, where, α is the learning rate;
- repeat the above process, and fine tune the training parameters (learning rate and neighbourhood radius) until either a given number of iterations is reached, no neuron movement is detected, or any other specified convergence condition is met.
- evaluate the network quality using performance measures (quantization and topographic errors).

The quantization error in equation (1) measures degree of similarity of the trained SOM neurons pattern to the inputs patterns — it accumulates euclidean distances from all inputs patterns to their closest neuron; while the topographic error given in equation (2) measures the degree to which the input dataset structure is kept by the trained network (Stefanovic and Kurasova, 2011).

$$Error_Q = \frac{1}{m}\sum_{p=1}^{m}\left\|X_p - h_{(winner)}(p)\right\| \quad \ldots. \quad (1)$$

$$Error_T = \frac{1}{m}\sum_{p=1}^{m} u(X_p) \quad \ldots\ldots (2)$$

SOM and k-means minimizes the distances between input samples and cluster centres and also converge when the neighbourhood radius of the former is equal to zero(0). However, SOM is more robust than k-means in the following ways; it is less prone to local optima and explores the data attributes search space more than k-means algorithm (Stefanovic, and Kurasova, 2011, Lobo, 2009). SOM algorithm produces results with higher accuracy than other clustering algorithms (Abbas, 2008). Furthermore, SOM performs additional tasks while the clustering operation is in progress (Bação, *et al.,* 2005; Stefanovic, and Kurasova, 2011). It divides the data-points into clusters and provides a visual of each uniformly organized cluster as well as attributes dimension reduction. The resulting clusters are structurally arranged based on cluster similarity (Lobo, 2009). These considerations made SOM the best candidate for partitioning and visualization of oil spillage datasets.

2.2 Deep Neural Networks and Ensemble Learning

Deep learning (DL) is a recently and widely reported subfield of machine learning (ML) that concentrates on training hierarchical models of data — representations of functions with many levels of composition. Successful DL approaches involve multi-layered structures of NN models, such as DNN, deep belief networks (DBN), convolutional neural networks (CNN) and stacked auto-encoders (Vincent *et al*., 2010). The classification depends on the dataset input characteristics (dimension, data type, etc), type of layers and learning methods with either perceptron, restricted Boltzmann machines or auto-encoders as building blocks (Min *et al.,* 2016). DNN is a conventional feed-forward, multi-layer perceptron (MLP) network with more than two hidden layers between its inputs space and output space layers (Li, *et al*., 2014, Deng and Yu, 2013). Since the depth of human brain inspires it, each hidden layer has learning features at hierarchically different levels of abstraction using either supervised or unsupervised learning approaches (Deng and Yu, 2013). DNNs are efficient in modeling highly varying mapping functions for building representations high dimensioned and correlated features, which enhances generalization in the tuning of weights (Ekpenyong, *et al.,* 2015, Larochelle *et al.,* 2009). It is proven efficient in the classification and recognition tasks associated with complex and large datasets, such as text corpora, images, sound, speech, and video.

Figure 1 is a generic structure of DNN and consists of sequentially ordered layers —input layer, multiple hidden layers, and output layer. Typically, DNN, comprises inputs ($\lambda_1, \lambda_2, ..., \lambda_n$) and outputs θ_k (*where* $k \geq 1$), the input vector at layer ℓ is v^1, while the weight matrix and bias vector as w^ℓ and δ^ℓ respectively. The hidden layers employ non-linear functions such as sigmoid, hyperbolic tangent, or rectified linear (Nair and

Hinton, 2010). Therefore, for a DNN having H hidden layers, the *h*-th hidden layer is $v^{l+1} = \delta(z(v^{\ell}), 0 \le \ell \le H$ where $z(v^{\ell}) = w^{l} v^{l} + \delta^{l}$ and $1/(1+e^{\lambda})$ is the sigmoid function. Equations (3) and (4) give representation of the nodes in the hidden layer (Huang *et al*, 2015);

$$\lambda_i = \begin{cases} w^1 \theta^t + \delta^1 \; ; & i = 1 \\ w^i y^i + \delta^i \; ; & i > 1 \end{cases} \tag{3}$$

$$y^h = \begin{cases} sig \bmod(\lambda_i); & h < H \\ soft \max(\lambda_i); & h = H \end{cases} \tag{4}$$

where, θ^t is the input frame at time t, H is the number of the hidden layers in the DNN, and both sigmoid and softmax functions are element-wise operations. The input vector λ_i is the pre-nonlinearity activations, and y^i and y^H are the output neurons vectors of *h-th* hidden layer and output layer, respectively. In each hidden layer, h_j, a mapping of its total input space from the preceding layer, x_j, to the succeeding layer, y_j, is performed. The output layer is a softmax layer, with values serially derived along other layers as the entire input values are fed into the network (Huang *et al.,* 2015). This is achieved by multiplying each hidden layer weight vector by their corresponding input vector (including the nodes' output values in the preceding layers) for each unit in the current layer to produce the weighted sum.

Larochelle *et al.,* (2009) gives three salient principles for training DNNs as follows: i.) Pre-training of layer nodes with unlabeled data for information preservation their respective layers configuration ii.) Activation of greedy algorithm for layer-by-layer training and modeling especially when unlabeled data are fed into the network (Hinton *et al.,* 2006) iii.) supervised fine-tuning of the whole network adopting appropriate deterministically computable training criterion of interest. The value of the results from these approaches largely depends on the nature and characteristics of the dataset. Optimization of the training parameters and assessment of the network quality are obtained by computing the deviation of the target from the actual outputs using any of these back-propagation derivatives (mean square error (MSE), root mean squared error (RMSE) mean absolute deviation (MAD), normalized root mean squared error (NRME), cross-entropy error function and correlations cost functions). DNNs have recorded inspiring performances in machine learning tasks such as image classification or speech recognition, largely because of their hierarchical layers of abstraction. However, the sequential, multi-layer and non-linear nature of their structure makes the degree of comprehensibility of the classification, recognition or prediction decisions, (especially when DNN is acting on new instances) process very insignificant (Samek et al., 2015).

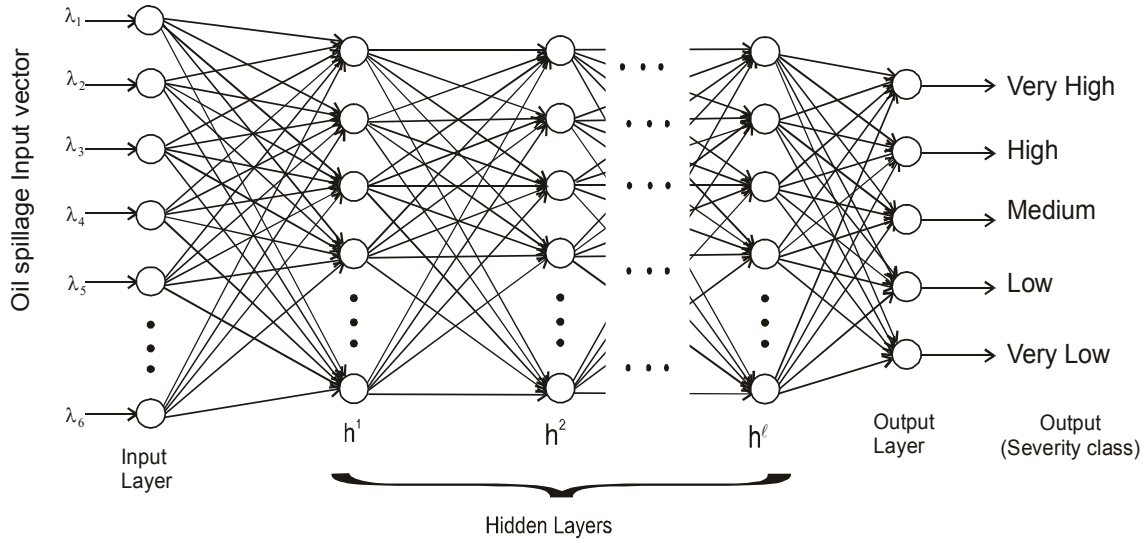

Figure 1: A generic DNN topology for oil spillage classification

2.3 Neuro-fuzzy Topology

Neuro-fuzzy framework is a composition of NN and FL models for synergic and complemetary adaptation and qualitative cognitive reasoning (Silarbi *et al*., 2014). This fusion results in an intelligent hybrid framework

known as Adaptive Neuro Fuzzy Inference System (ANFIS), having NN components (for recognition, adaptive learning and self-organization) and fuzzy inference system (for decision making in uncertain and imprecise environments). The fuzzy inference system (FIS) consists of fuzzification, rule inference and fire-strength computation, and output defuzzification as stages. The main strength of this hybrid sturcture is the encoding of FIS parameters as weights in the NN which can be optimized by using appropriate learning methods. ANFIS development involves four distinct phases; initialize, train, test and deploy. ANFIS deployment spans four main steps, firstly, model extraction from input vector space determination of the degree of satisfaction for each premise parameter by extracting rules defining the input-output trends that describes the dataset. Conjunction operation on the premise and fuzzy inferred parameters and fine tuning of the earlier extracted fuzzy system rules to produce the final ANFIS system — defuzzification (Buragohain, 2008).

The ANFIS model adopted for this work, implements a five-layered structure of a first order Sugeno model that is designed to accept oil spillage input features and then sequentially performs mappings to the respective MFs. The process begins with the mapping of input MFs to their respective rules and then rules mapping to output MFs. Thereafter, output MFs are subsequently mapped to the outputs values (Vaidhehi, 2014). The first layer are adaptive nodes, consisting of oil spillage features {type, cause and location} and executes the fuzzification operation — generates a degree of belonginess for each oil spillage feature. In this paper, the Guassian MF given in equation (5) (Yadav, 2014) is adopted in the prediction of Oil spillage severity level.

$$\mu_{A_i}(x) = \exp\left[-\frac{1}{2}\frac{(x-\beta_i)^2}{\sigma_i}\right] \quad \cdots \quad (5)$$

Where, β_i and σ_i are the two premise parameters governing Gaussian MFs and represents, the centre and width of the *ith* linguistic label A_i, respectively, *x* is the input to node *i (i=1,2,..., n).* The second layer, has fixed nodes for the computation of the firing strength of rules (using fuzzy OR or AND operator) by fuzzifying the condition variables of each rule. The ratio of the *ith* rule's firing strength to the sum of all rule's firing strengths, is performed by nodes in the third layer. Another layer of adaptive nodes, where the product of rules' normalized firing strength and the first order polynomial of Sugeno consequent parameters (r_0, r_1, r_2, r_3) are calculated is the fourth layer. Iterations involving forward and backward passes are responsible for the identification, generation and tuning parameters of the FIS. In the forward pass, nodes premise parameters are not affected (Zhang, et al., 2014) while layer-wise nodes' output computation up to the fourth layer is performed leaving the rules' conclusion parameters updated by least-squares estimation (LSE). Thereafter, the quality of the model is assessed using the RMSE given in equation (6) (Yadav *et al.*, 2014). The error signals moves backwards and back-propagation gradient descent models tune antecedent parameters without updating or altering the conclusion parameters during the backward pass phase. The single node (summation neuron) in the fifth layer computes the final output of the ANFIS by accumulating arriving signals of the fourth layer and executing defuzzification sub-task (the process of converting fuzzy results of each rule into crisp value) using the updated FIS parameters. The output function, O_i^l, for each layer is given in equations (7) – (11) (Inyang and Akinyokun, 2014, Odabas, *et al.,* 2014, Vaidhehi, 2014, Yadvav et al., 2014).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_{desired} - X_{Anfis})} \quad (6)$$

where $X_{desired}$ is the actual oil spillage severity level, X_{Anfis} is the observed output from ANFIS

$$O_i^1 = \mu_{A_i}(x) = \exp\left[-\frac{1}{2}\frac{(x_i-\beta_i)^2}{\sigma_i}\right] \qquad i=1,2,3 \quad (7)$$

$$O_i^2 = \alpha_i = \mu_{A_n}(T_i)\mu_{T_n} * (C_i)\mu_{C_n} * (L_i), \mu_{L_n} \quad (8)$$

$$O_i^3 = \overline{\alpha_i} = \frac{\alpha_i}{\sum_i \alpha_i} \quad (9)$$

$$O_i^4 = \alpha_i f_i = \alpha_i \times (r_0^i + r_1^i T_i + r_2^i C_i + r_3^i L_i) \quad (10)$$

$$O_i^5 = \sum_i \overline{\alpha_i} f_i = \overline{\alpha_1} f_1 + \overline{\alpha_2} f_2 + \overline{\alpha_3} f_3 \quad (11)$$

$$= \frac{\sum_i \alpha_i f_i}{\sum_i \alpha_i}$$

3. System Design and Methodological Workflow

The methodological workflow in Figure (2), has dataset collection, pre-processing and oil spillage attribute selection as activities in its initial phase. The next stage had series of experiments aimed at identifying the appropriate number of natural divisions in the dataset using k-means algorithm, and silhouette criterion-based cluster validity analysis. The other steps spans in a three-staged learning process —SOM for unsupervised learning, DNN ensembles, and ANFIS for supervised learning. The optimal number of clusters (k) guided SOM network in the partitioning of the dataset into k distinct clusters and the assignment of class to each record of the dataset. Each cluster consists of data points with similar statistical distributions for the adaptation to the non-stationery attributes of oil spillages. In the next phase, k-DNN ensembles were created with each DNN assigned to a cluster dataset. Rules and weights extracted from each DNN and their respective output vectors were data utilized by the ANFIS model for tuning of ANFIS parameters when the training, validation, and classification of the severity levels of oil spillage activities are performed.

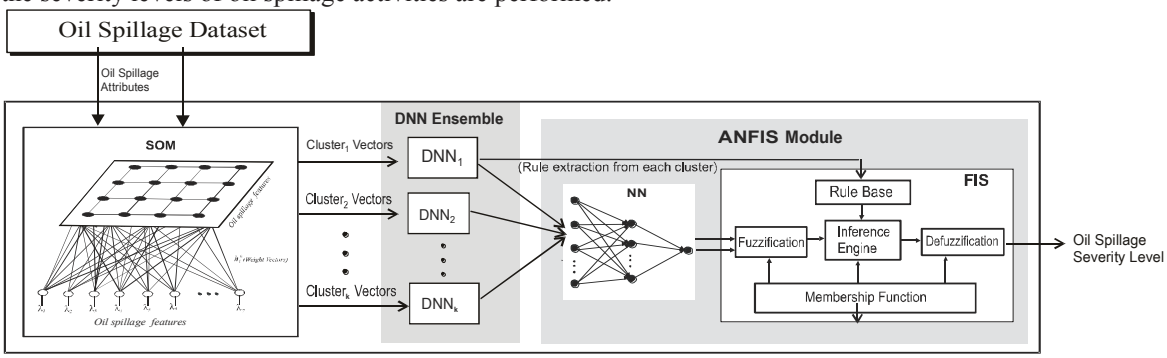

Figure 2: schematic workflow of the adaptive cooperative system

3.1 Oil spillage Dataset pre-processing and Description

The dataset for this research consists of 1008 samples of oil spillage data collected by National Oil Spill Detection and Response Agency (NOSDRA) from the Southeast costal region of Nigeria. Attributes of the dataset employed as indicators are day, year, time, location, type and cause. Attribute selection and pre-processing relied on methodology and sensitivity analysis results provided in Inyang and Akinyokun, (2014) and Akinyokun and Inyang (2013) with *type of spill Oil, cause* and *location of spill* as significant attributes to the magnitude of spillage. The inputs are categorical, therefore are represented with codes in the range [1, 6], for Cause and Type while Location has the range [1,2]. The representation of each attribute is described as follows; attribute Cause — operational/maintenance error(1), sabotage (2), equipment failure (3), corrosion (4) , yet -to – be determined (5) and miscellaneous sources (6). Attribute Type of spilled — refined product (1), crude oil (2), chemical (3), AGO(4) condensate(5) and Others (6). Location of spillage — onshore (1) and offshore (2). The target variable is a function of fuzzy sets (equation 12) with elements mapped to their respective degree of membership of the target attribute [Yadav *et al.,* 2014].

$$A = \{\mu_A(x) : x \varepsilon X\}$$

$$A = \left\{ \frac{\mu_A(x)}{x} : x \varepsilon X \right\} \tag{12}$$

Where $A = \{\mu_A(x) : x \varepsilon X\}$, is a MF of the fuzzy set A and $\mu_A(x)$ is the degree of membership of x in X in the fuzzy set A. Furthermore, mapping of $\mu_A(x)$ to the every MF of the fuzzy linguistic values of *"low", "medium", "high"* or *"very high"* is carried out, and also transformed into binary encoding for efficient processing by the DNN. However, the unsupervised learning assigns to each record, a severity level which serves as the target variable. The binary encoded target variable has four elements, comprising three zeros (0) and a one (1) in the location of the associated severity class.

3.2 Distinct clusters discovery and validity analysis experiment

Clustering deals with the analysis and organization of datasets into separate classes based on similarities exhibited by the data samples in the dataset. Clustering is necessary for effective classification and visualization analysis, especially in datasets where there are no natural boundaries or divisions. This therefore requires an approach of finding a specific number of clusters that inherent in the data or that best satisfies the natural partitions in the dataset without any a priori group knowledge. Rendón, *et al.,* (2011) reports that cluster validation (finding the partitioning that best fits the underlying data) is the panacea for inefficient clustering results and requires the application of appropriate validity indexes. However, the suitability of an index is

dependent on the charateristics of the data repository; environmental factors and the type input and output values (Deborah, *et al*., 2010). External, internal and relative criteria are three relevant measures clusters quality evaluation. Internal and external criteria are computationally costly and are driven by statistical tests techniques. External validity indexes requires user specified parameters and knowledge, that are not extracted from the dataset, in contract to internal criteria, that utilizes the quantities, features and relationships inherent in the dataset for the assessment of cluster partitions (Liu, *et al.,* 2010, Deborah, *et al.,* 2010, Rendón, *et al.,* 2011). The approach adopted with relative criteria is to compare different clustering schema according to a defined criterion, by executing a number of algorithms several times with specific set of parameters and data points from where the best performing algorithm is chosen. Internal validity indices rely on cohesion (proximity of data points within a given cluster) and separation (a measure of degree of distinctiveness or distance between any two or more clusters) measures (Rendón *et al,* 2011). Pham *et al.,* (2005) and Rendón, *et al.*, (2011) describe various cluster evaluation techniques with emphasis on the silhouette criterion. The silhouette criterion relies on the pairwise difference between within-cluster data-points distances in evaluating clustering performance (Liu, *et al*., 2010). The silhouette value (SV) for a point specifies the degree of similarity between that data-point and other data-points in its cluster. The optimal cluster number is determined by maximizing the derived SV metric value. The SV for the *ith* point, S_i, is stated in equation (13) (Rendón, et al., 2011)

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{13}$$

where a_i is the average distance separating the *ith* data-point and other data-points in the same cluster as i, and b_i is the minimum average distance separating the *ith* data-point and data-points in the other clusters. A high SV implies tightly clustered elements within its own cluster, and very loosely clustered elements to members of other clusters. A high SV indicates that the solution is appropriate whereas a low or negative SV, suggests an inappropriate clustering solution consisting either too many or too few clusters.

In this paper k-means is the partitioning algorithm while silhouette criterion is the validation index for optimal number of cluster (k) determination in the oil spillage dataset. The general procedure has the following stages; first is the initialization of k-means parameters, followed by iterative partitioning of the dataset into distinct number of clusters with varying distance measures (cityblock, SqEuclidean, correlation) and number of clusters. The Mean silhouette value (MSV) — the average SV of all data-points in a cluster, is computed for each corresponding partition obtained in the second step. In the final stage, the best partition and the optimal cluster number according to the silhouette criterion is chosen. The experiment was conducted with a maximum of 4 clusters and the results presented in Table 1 and Figure 3.

Table 1: performance of number of clusters on distance measures

Number of Cluster	Cityblock	SqEuclidean	Correlation	Total
2	0.4682	0.6422	0.7231	1.8335
3	0.347	0.7841	0.6313	1.7624
4	0.5677	0.2279	0.7769	1.5725
5	0.5209	0.6235	0.7321	1.8765
Average	0.47595	0.569425	0.71585	

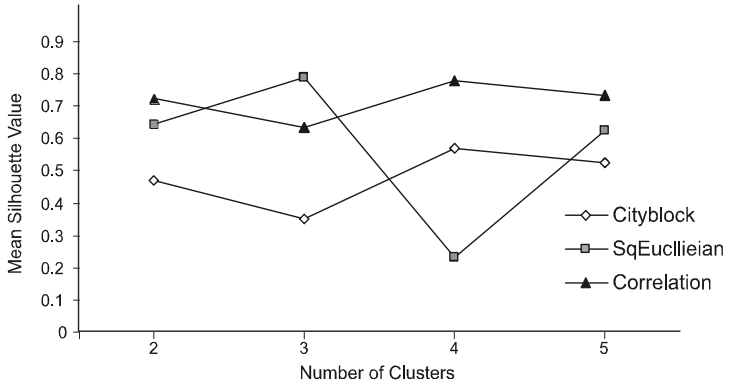

Figure 3: Graph of distance metric performance across number of cluster

As shown in Figure 3, the correlation metric – the sample correlation between points treated as sequences of values, yielded the best SVs across clusters with a MSV of 0.71585, followed by Squared Euclidean distance (SqEuclidean) with MSV of 0.5694 while *cityblock* had the least value of 0.47595.

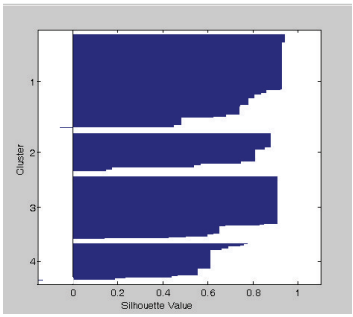

Figure 4: Silhouette plot of filled clusters

Thus, the optimal of the number of natural clusters is based on the correlation distance metric with 4 natural clusters had the highest MSV of 0.7769. This reveals the existence of four (4) appropriate and compact clusters. The distribution of the data-points in the four clusters are as follows: cluster 1(408), cluster 2 (167), cluster 3 (273) and cluster 4 (160). The silhouette plot in Figure 4, reveals that majority of data points clustered around SV greater than 0.7769 – an indication of a very good and dense cluster within the oil spillage dataset.

3.3 SOM Clustering of Oil Spillage Dataset

SOM was applied to identify and classify oil spillage instances into the segments with similar spillage features. The design was implemented with Matlab R2015a as front-end tool, while Microsoft Excel was the database management tool. Oil spillage instances in the dataset were randomly partitioned into training (70%), testing (15%) and validation (15%) datasets. The input layer of SOM has three nodes while the number of neurons in the output layer was set to k (k=4, as determined from the k-means experiment). The connection weights vector ($h_1^1, h_2^1, ..., h_n^j$), link each neuron and has same dimension as the input vector. Training of weights and bias learning rules with batch updates was driven by trainbu algorithm in two basic steps —rough and fine training phases). At the rough training phase, which had 1000 epochs, an initial and final neighborhood radius was 5 and 2 respectively with decreasing iterating learning rates ranging from 0.5 and 0.1. During updating phase, the learning rate converged at 0.2 with a maximum of 500 iterations. The Euclidean distance guided the selection of the best centroids of the oil spillage feature in each cluster. The map topology and the cluster centres are as shown in Tables 2 and 3 respectively. The map quality (Total Sum of Squares) of 71% as an indication an appropriate representation of the inputs patterns by the trained SOM neurons. As shown in Table 2, cluster 1,1 (cluster 1) has 406 data points while cluster 1,2 (cluster 2) and cluster 2,1 (cluster 3) consists of 168 and 336 members respectively, cluster 2,2 (cluster 4) has 98 data-points. The target class of each turple (assigned by SOM) together with inputs space constitutes the dataset for subsequent stages of this work.

Table 2: Cluster map topology

	1	2
1	406	168
2	336	98

Table 3: Attribute based cluster centers

Attribute	Cluster			
	1	2	3	4
Location	31.87	31.44	32.00	31.04
Cause	41.17	41.45	45.36	44.60
Type	51.21	54.71	51.51	53.27

4. DNN Ensemble Development

Every DNN in the ensemble, is a feed-forward network structure, consisting of an input layer, multiple (more than two) hidden layers and an output layer. The input layer has three neurons (location, cause and type) that are fully connected to the first hidden layer neurons. It sends and receives signals to and from external environment and presents the oil spillage trends to the DNN. As the input vector gets into the DNN through the input layer, nodes of the DNN compute their values based on the activities of the other nodes linked to them in the succeeding layers (Panchal1, et al., 2011). Oil Spillage inputs are associated with four target classes assigned by SOM — (very high, high, medium, and low) of oil spillage magnitude. Thus, the output layer of each DNN has four neurons, each assigned to a target class. The workflow for each DNN design and implementation proceeded in the following phases; data collection and preparation, DNN creation, configuration, initialization training,

validation and deployment.

4.1 Data preparation and DNN ensemble creation

The SOM clustering results—partitioning of oil spillage dataset into four distinct clusters and assignment of every data-point to a cluster, each is the dataset for the respective DNN models. Data points belonging to each cluster are organized in two matrices, the cluster input space matrix (c_Input_Data) and the cluster target space matrix (c_Target_Data). Each *ith* column of the c_Inputdata matrix has three elements representing an oil spillage pattern whose severity class is explicitly stated in the *c_Target*data. The *c_Target*data matrix has four columns (elements), comprising three zeros (0) and a 1 in the position of the associated severity level (class). Specifically, DNN_1 dataset is $c\_Inputdata_1$ consisting of 376 columns of oil spillage instances) and three rows (for three oil spillage attributes) while $c\_Targetdata_1$ has four rows (representing severity class of the spillage). The summary of the sizes of the input and target dataset for DNNs is given in Table 4. Each DNN dataset (clustered dataset) was randomly divided such that, 70% of the samples were for training by computing the gradient and updating the network weights and biases, 15% for validation and 15% for testing of the DNN models. Each DNN was created with *tansig* and *purelin* as transfer functions for hidden and the output layer respectively. Decision on the number of hidden layers and hidden layer neurons in DNN is an important design task of DNN (Panchal1 et al., 2011, Karsoliya, 2012).

Table 4: Description of DNN datasets

DNN Ensembles	Size	
	c_Inputdata	c_Targetdata
DNN_1	3×406	4×406
DNN_2	3×336	4×336
DNN_3	3×168	4×168
DNN_4	3×98	4×98

The rules for choosing the optimal size of hidden layer described in (Karsoliya, 2012; Boger and Guterman,1997; Berry and Linoff, 1997), informed fixing the maximum size of the hidden layers to 6, the backward pass approach (Panchal and Panchal, 2014) was then applied to determine the best performing number of hidden layer based on percentage accurate classifications. The number of neurons in each hidden layer was the same as the number of input neurons (Karsoliya, 2012).

4.2 DNN Training, performance analysis and utilization

The process of training, validation, testing and utilization each DNN in the ensemble, begins with initialization of the weights and biases. Scaled conjugate gradient (SCG) back propagation, which is independent of user-defined parameters, proves to be the fastest and reliable training algorithm (Moller, 1993). In this work, the supervised learning therefore utilized SCG algorithm and gets the entire input samples of each clustered training dataset before updating weights at a learning rate of 0.02. Thus, this methodology iteratively computes gradients, updates weights, and biases either in the forward or backward direction by a decreasing performance function. The stopping criterions for training were 0.00001 and 6 for gradient magnitude and validation checks count respectively. Furthermore, the performances of the DNNs were also assessed using RMSE —the average squared error between the DNN results and the target outputs and confusion matrix (Koco and Capponi, 2013). Figures 5-8, are the objective functions indicating the performances of the respective DNNs in fitting each cluster's dataset. In all cases, the training, validation and testing errors reduced as the epoch increased without any overfitting. Specifically, in cluster 1 DNN, the validation error (0.01136) reached a minimum at 52nd epoch while the best performance of DNN 2 was noticed at epoch 21. In the training of cluster 3 and cluster 4 datasets, validation checks on the DNNs were best at epoch 23 (validation error of 0.02772) and 20 (validation error of 0.057762) respectively. Table 5 and Figure 9 show the performances of DNN models in the classification of oil spillage cluster datasets.

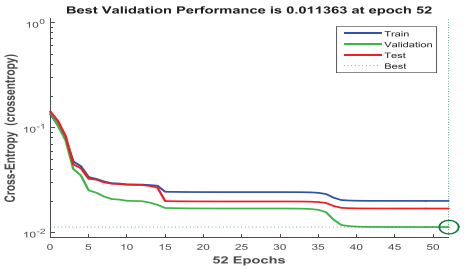

Figure 5 : Performance of DNN1 during training, validation and testing of oil spillage features of cluster 1

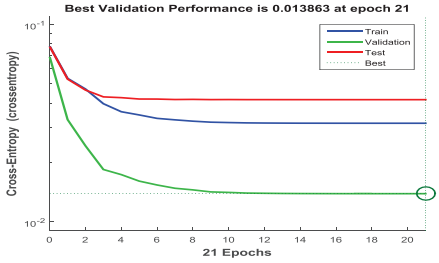

Figure 6 : Performance of DNN2 during training, validation and testing of oil spillage features of cluster 2

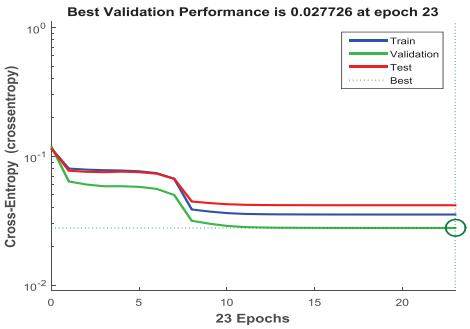

Figure 7 : Performance of DNN3 during training, validation and testing of Oil spillage features of cluster 3

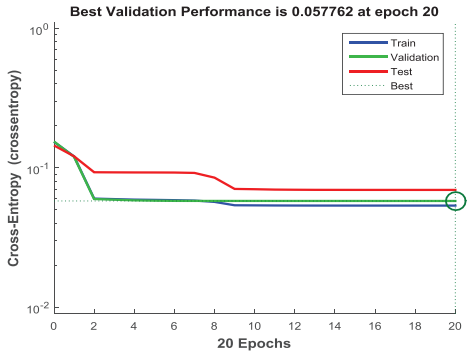

Figure 8: Performance of DNN4 during training, validation and testing of Oil Spillage features of cluster 4

Table 5: Performances of DNNs on each clustered dataset

	Train	Test	Overall performance
DNN1	0.4434	0.1682	0.0162
DNN2	2.8082	7.9681	0.0335
DNN3	0.1795	0.4487	0.0351
DNN4	0.2033	0.4544	0.0637

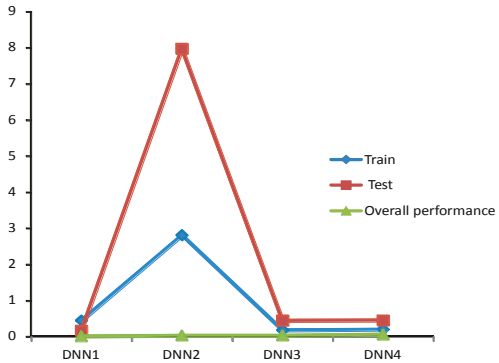

Figure 9: Performance of DNN models on each clustered dataset

As shown, in the confusion plots, given in Figures (10) – (13), the diagonals depict accurate classifications, bottom-right cells give the overall accuracy while other cells represent incorrect classifications. In Figure 10, out of the 406 samples classified as cluster 1 by SOM, 239 examples (58.9%) and 124 samples (30.5%) were correctly classified as cluster 1(very high) and cluster 2 (high) members respectively by DNN1 resulting in an overall accuracy of 89.4%. In DNN2, 71.4% (240 samples) was correctly classified into *high* while 3.9% (13 samples) were misclassified as cluster 3 (medium) elements. The overall accuracy of DNN3 and DNN4 are 86.9% and 67.3% respectively. In general, it quite obvious, that the level of confusion has decreased and only limited to two classes in each DNN. This affirms the suitability of the two staged clustering and further confirms the existence uncertainty in the oil spillage dataset. Thus, a methodology for handling these uncentainty is necessary.

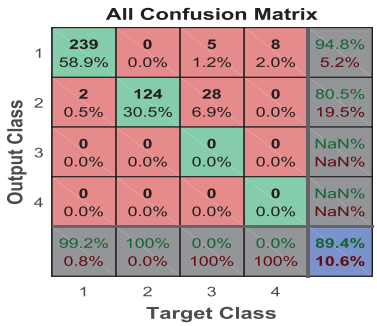

Figure 10 : Confusion Matrix DNN1 classification

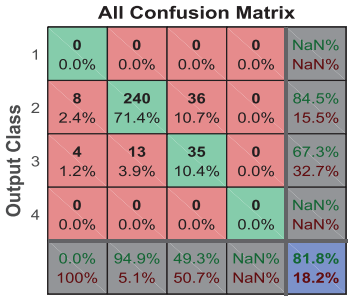

Figure 11 : Confusion Matrix DNN_2 Classification

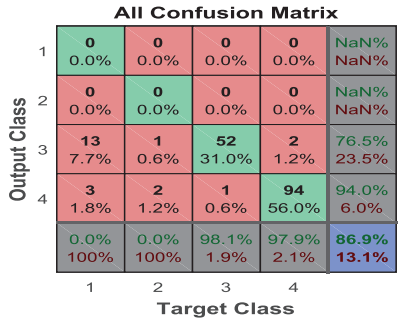

Figure 12 : Confusion Matrix DNN_3 Classification

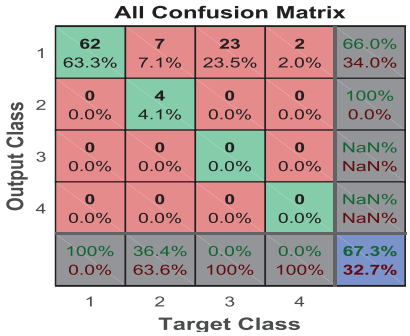

Figure 13 : Confusion Matrix DNN_4 Classification

The rule extraction procedure in Lu, *et al.,* (1996) was adopted to extract seventy two (72) rules from the DNN models and for each set of rule, associated weights, w_{ij} _ where *i* is the rule number and *j* is the ensemble DNN component, the rule with the maximum weight is selected and processed by the ANFIS model.

4.3 ANFIS Model and Results

The ANFIS model is a 5-layered structure consisting three nodes in the input layer, fuzzification layer (12 nodes), rule layer (72 nodes), normalization layer (72 nodes) and 1 node each for defuzzification and output layer. It uses crisp values of neuron weights and output from trained DNN ensembles as input data spaces and through fuzzification, inference and defuzzification, builds fuzzy rules using sugeno-inference mechanism. The Fuzzy c-means (FCM) algorithm generated the Fuzzy Inference System (FIS) with Gaussian membership function *(gaussmf)* using *genfis3* for 4 clusters. Out of the 200 epochs, it converges at 32nd iteration with an objective function value of 48.801 (Figure 14). The parameters (centre and width) of the *gaussmf* for each input indicator in the 4 clusters are as presented in Table 6.

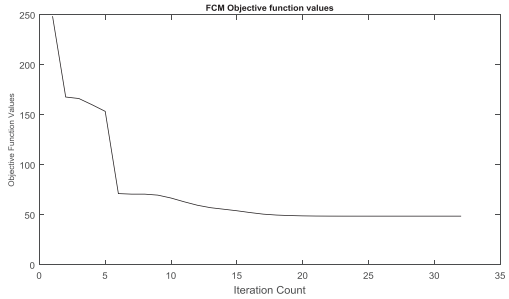

Figure 14: Objective function values

Table 6. Cluster Centres Parameters

	Cluster Index			
Oil Spillage indicators	1	2	3	4
Location	[1.7794 , 2.3837]	[-2.8867 , -2.6166]	[1.7295 , -2.3545]	[-2.2477 , 1.0219]
Type	[2.8570, 2.6681]	[0.16783, 5.07452]	[-24.1981 , 7.8102]	[3.650, 4.3957]
Cause	[0.68021, 4.7491]	[3.22106 , 7.024]	[9.7496 , -20.9543]	[8.5270, 2.999]

At the training phase, a NN based algorithms — using a combination of LSE and gradient descent approaches, identified and updated the FIS consequent parameters (p), $\{r_0, r_1, r_2, r_3\}$ and the results in 4x 4 dimension are given in equation (14)

$$p = \begin{bmatrix} 0.9626 & 1.5226 & 0.9930 & 2.03089 \\ 2.6987 & 0.0286, & 0.06805 & -7.1462 \\ -5.9289 & 3.2108 & -4.3565 & 42.6670 \\ 0.24118 & -0.0300 & -0.10251 & 1.70668 \end{bmatrix} \quad (14)$$

The row represents the clusters; column 1 is the constant value vector while parameters of each cluster of oil spillage feature are represented in the succeeding columns. A row has variables of the first order polynomial for deriving the output values in each cluster as given in equation (10) above. The training (Figure 15) and testing (Figure 16) results show significant positive drift from the results of the DNN in each cluster. The *mse* value of the training is 0.045 while testing error is 0.049.

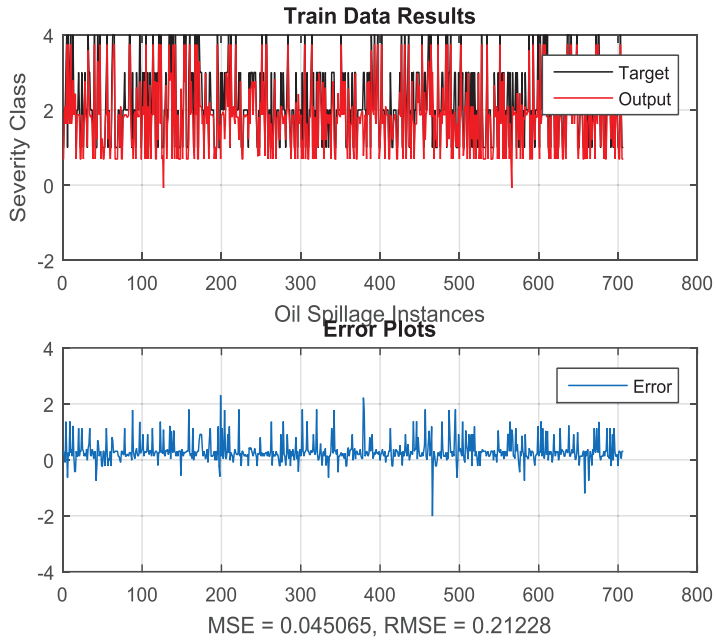

Figure 15: Target and output relationship and error plots during training of oil spillage dataset.

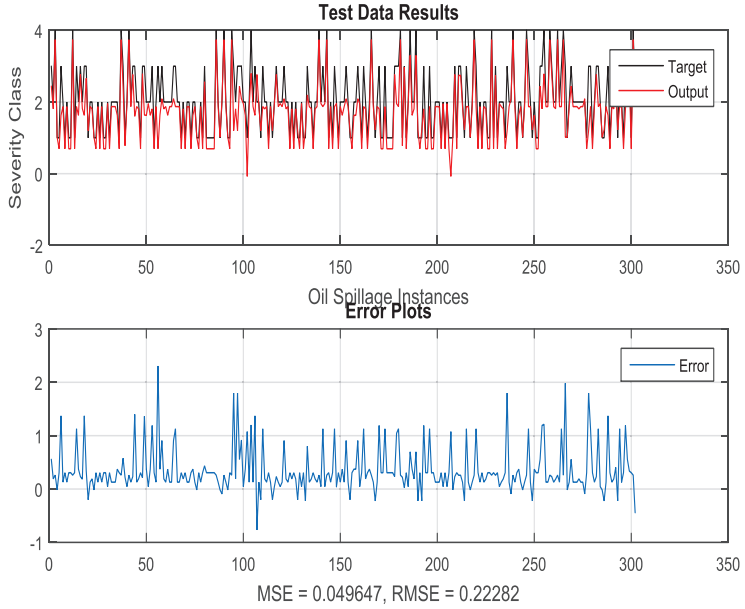

Figure 16: Relationship between target and output, and error plots during Testing of oil spillage dataset.

This improvement is because of the ANFIS's capability in handling uncertainty and vagueness characterizing oil spillage severity analysis. The relationship between the target vector and the ANFIS result also improved significantly with R=0.8314 (Figure 17).

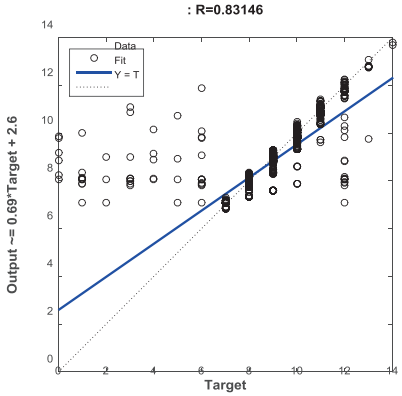

Figure 17: Relationship of Actual targets relative to Predicted Oil Spillage Severity level

5. Conclusion

The understanding oil spillages patterns is a roadmap for precise and real time predictions of severity (as opposed to existing rough and discrete prediction) to give decision makers a more realistic picture of the state of the environment. This paper seeks to address this problem by exploiting oil spillage features with sets of collected data of oil spillage scenarios. The proposed system integrates three state-of-the-art tools: self-organizing maps, (SOM), ensembles of deep neural network (k-DNN) and adaptive neuro-fuzzy inference system (ANFIS). It begins with unsupervised learning using SOM, (for clustering and visualizing the patterns and assignment of classes as targets), thereby making the data suitable for classification and prediction (supervised learning) by ensembles of k-DNN and ANFIS. Four compact and appropriate clusters were discovered with high silhouette values. DNN and ANFIS performed classification and predictions of severity classes. Results obtained yeilde R=.83 which is significant classification and prediction improvements, which is largely attributed to the hybrid learning and ensemble learning approaches, and fuzzy cognitive reasoning. However, optimization of k-DNN structure and weights would be needed for speed enhancement in the system while fuzzy type II would eliminate the uncertainty lingering in the dataset.

References

Abbas, O. A, (2008). Comparisons between data clustering Algorithms. The international Arab Journal of Information Technology 5(3), 320-32.

Adelana, S.O , Adeosun.T. A, Adesina , A.O and Ojuroye, M.O (2011). Environmental pollution and remediation: challenges and Management of oil Spillage in the Nigerian coastal areas American Journal Of Scientific And Industrial Research. doi:10.5251/ajsir.2011.2.6.834.845

Akinyokun, O. C. and Inyang, U.G. (2013). Experimental Study of Neuro-Fuzzy-Genetic Framework For Oil Spillage Risk Management. Artificial Intelligence Research. 2(4) 13-35

Amiryousefi, M. R., Mohebbi, M., Khodaiyanb, F.,and Asadi, S. (2011). An Empowered Adaptive Neuro-Fuzzy Inference System Using Self-Organizing Map Clustering to Predict Mass Transfer Kinetics in Deep-Fat Frying of Ostrich Meat Plates. Computers and Electronics in Agriculture 76 (2011), 89–95

Bação, F., Lobo, V. and Painho, M. (2005). The Self-organizing Map, The Geo-SOM, And Relevant Variants For Geosciences. Computers & Geosciences, 31(2), 155-163. Doi:10.1016/j.cageo.2004.06.013

Berry, M. J. and linoff, G. (1997). Data Mining Techniques: For Marketing, Sales, and Customer Support. John wiley & Sons New York, NY, USA.

Boger, Z., and Guterman, H., (1997). Knowledge extraction from artificial neural network models. IEEE Systems, Man, and Cybernetics Conference, Orlando, FL, USA

Buragohain, M (2008). A Novel Approach for ANFIS Modelling Based on Full Factorial Design. Applied soft computing. Journal of applied soft computing. 8(1) pp 609-625

Deborah, L. J. , Baskaran, R., and Kannan, A. (2010). A Survey on Internal Validity Measure for Cluster Validation. International Journal of Computer Science & Engineering Survey (IJCSES) 1(2),

Deng, Li and Yu, D. (2014). Deep Learning Methods and Applications. Foundations and Trends in Signal Processing (7) 3–4), 197–387. DOI:10.1561/2000000039

Egwu , S. A. (2012): Oil Spill Control and Management Petroleum Technology Development Journal (ISSN 1595-9104)

Ekpenyong, M. E., Inyang, U. G. and Ekong, V. E. (2015). A DNN Framework for Robust Speech Synthesis Systems Evaluation. In Proceedings of 7th Language and Technology Conference (LTC), Poznan, Poland, Zygmunt V. and Mariani, H. (eds.), Fundacja Uniwersytetu im. A. Mickiewicza: 256-261. http://ltc.amu.edu.pl/book/papers/LRT-4.pdf

Gabralla1, L. A, Wahby1, T., Ojha, V. K. and Abraham, Ajith (2014). Ensemble of Adaptive Neuro-Fuzzy Inference System Using Particle Swarm Optimization for Prediction of Crude Oil Prices. International Conference on Hybrid Intelligent Systems (HIS) …

Geng, L.; Z. Chen; C.W. Chan; G.H. Huang (2001). An intelligent Decision Support System for Management of Petroleum Contaminated Sites. Expert System with Applications. Vol. 20, pp. 251-260

Gorman, C., Rogers, C. and Valova, I. (2015): No Neuron Left Behind: A Genetic Approach to Higher Precision Topological Mapping of Self-Organizing Maps. Open Computer Science, De Gruyter 5:1–12

Gurney, K. and Gurney, K.H. (1997) An Introduction to Neural Networks, CRC Press.

Hinton, G. E., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18, 1527–1554.

Huang, Z, Li, J., Siniscalchi, S. M., Chen, I. F J, Wu, and C-H Lee (2015). Rapid Adaptation for Deep Neural Networks through Multi-task Learning. in Proceedings of Interspeech,.

Hernandez, J.Z.; J.M. Serrano. (2001). Knowledge-based models for emergency management systems. Expert Systems with Applications, 20(2), 173-186. Elseviet Science Ltd

Inyang and O.C. Akinyokun (2014). A Hybrid Knowledge Discovery System for Oil Spillage Risks Pattern Classification. Artificial Intelligence Research, 3(4): 77-86.

Iwejingi, S. F (2013), Socio-Economic Problems of Oil Exploration and Exploitation in Nigeria's Niger Delta . Journal of Energy Technologies and Policy 3(1), 76-80

Kao, W,. Hung, J. C. and Hsu, V. (2008): Using Data Mining in MURA Graphic Problems. Journal of Software. 3(8), pp73-99.

Karsoliya, S. (2012) :Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture. International Journal of Engineering Trends and Technology- Volume3Issue6- 2012

Koco, S. and Capponi, C.(2013). On Multi-Class Classification Through the Minimization of The Confusion Matrix Normam Workshop and Conference Proceedings 29:277–292.

Kohonen, T. (2013). Essentials of Self-organizing Maps. Neural Networks. 37(2013): 52-65.

Larochelle, H., Bengio, Y., Louradour, J., and Lamblin, P. (2009): Exploring Strategies for Training Deep Neural Networks. Journal of Machine Learning Research 1 (2009) 1-40

Lei, K. S., and Wan, F. (2012). Applying ensemble-learning techniques to ANFIS for air pollution index prediction in Macau. Advances in Neural Networks, 7367, 509- 516. J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part I, LNCS 7367, pp. 509–516, 2012. © Springer-Verlag Berlin Heidelberg

Li, J, Zhao, R.,. Huang, J. and Gong, Y. (2014). Learning Small-size DNN with Output-Distribution-Based Criteria. In Proceedings of the Annual Conference of International Speech Communication

Association (

Liu, Yanchi, Li, Zhongmou, Xiong, Hui, Gao1, Xuedong, Wu, Junjie (2010). Understanding of Internal Clustering Validation Measures. In proceedings of IEEE International Conference on Data Mining pp 911-916

Lobo, V. S. (2009). Application of Self-Organizing Maps to the Maritime Environment. V.V. Popovich et al, (eds.), Information Fusion and Geographic Information Systems, Lecture Notes in Geoinformation and Cartography. DOI: 10.1007/978-3-642-00304-2_2, © Springer-Verlag Berlin Heidelberg 2009

Lu, H., Setiono, R. and Liu, H. (1996). Effective Data Mining Using Neural Networks. IEEE Transactions on Knowledge and Data Engineering, 8(6), 8(6), 957-961

Min, S. , Lee, B. and Yoon, S. (2016). Deep Learning in Bioinformatics. Briefings in Bioinformatics arXiv:1603.06430 [pdf

Moller, M., F. (1993).A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks. 6(4):525-533. doi:10.1016/S0893-6080(05)80056-5, (1993).

Mosley, R (2005): The Use of Predictive Modeling in the Insurance Industry", Pinnacle Actuarial Resources.

Odabas, M. S., Temizel, K., Caliskan, O., Senyer, N., Kayhan, G., and Ergun, E. (2014). Determination of Reflectance Values of Hypericum's Leaves Under Stress Conditions Using Adaptive Network Based Fuzzy Inference System. 79-87. Neural Network World. Volume: 24 DOI:10.14311/NNW.2014.24.004

Panchal, F. and Panchal, M (2014). Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network. International Journal of Computer Science and Mobile Computing. 3(11), 455 – 464

Panchal1, G., Ganatra2, Amit, Kosta, Y. and Panchal, Devyani (2011). Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers. International Journal of Computer Theory and Engineering, Vol. 3, No. 2, April 2011

Pham, D. T., Dimov, S. S. and Nguyen, C. (2005). Selection of K in K-means clustering. Proc. IMechE Vol. 219 Part C: J. Mechanical Engineering Science 103-109. DOI: 10.1243/095440605X8298

Rendón, E., Abundez, I., Arizmendi, A. and Quiroz, E. (2011). Internal versus External cluster validation indexes. International Journal Of Computers And Communications 1(5), 27-34

Samek, W., Binder, A., Montavon, G., Bach, S. and Muller, K.(2015). Evaluating The Visualization Of What A Deep Neural Network Has Learned. CoRR, vol. abs/1509.06321, 2015.

Santos R. B, de Sousa E. O, da Silva F. V, da Cruz S. L and Fileti A. M. (2013). Detection and on-line Prediction of Leak Magnitude in a Gas Pipeline using an Acoustic Method and Neural Network Data Processing. Brazilian Journal of Chemical Engineering, 13(1), 60-67.

Silarbi, S., Abderrahmane, B., Benyettou, A. (2014). Adaptive Network Based Fuzzy Inference System for Speech Recognition Through Subtractive Clustering. International Journal of Artificial Intelligence & Applications (IJAIA), 5(6). 43-52. DOI : 10.5121/ijaia.2014.5604

Stefanovic, P. and Kurasova, O. (2011). Visual Analysis of Self-Organizing Maps. Nonlinear Analysis: Modelling and Control, 16(4), 488–504

Vaidhehi, V. (2014). The role of Dataset in Training ANFIS System for Course Advisor. International Journal of Innovative Research in Advanced Engineering (IJIRAE) 1(6), 250-259

Vincent P, Larochelle H, Lajoie I, Bengio, Y. and Manzagol, Pierre-Antoine (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network With a Local Denoising Criterion. The Journal of Machine Learning Research, 11(2010): 3371- - 3408.

Wilson, R.A. (2001). Transportation in America, Eighteenth Edition. Washington, D.C.: Eno Transportation Foundation, Inc.

Yadav, Ramjeet Singh, Soni, A.K., and Pal, Saurabh(2014). Modeling Academic Performance Evaluation Using Hybrid Fuzzy Clustering Techniques 8(3), 98-111

Yao, Z., Sarlin, P., Eklund, T., and Back, B. (2014). Combining Visual Customer Segmentation and Response Modeling. Neural Computing and Applications, 25(1): 123-134.

Zainal, A., Maarof, M. and Shamsuddin, S. M (2009). Ensemble Classifiers for Network Intrusion Detection System. Journal of Information Assurance and Security 4 (2009), 217-225

Zhang, L. Liu, J. Lai, J. and Xiong, Z. (2014). Performance Analysis of Adaptive Neuro Fuzzy Inference System Control for MEMS Navigation System. Mathematical Problems in Engineering. Volume 2014 (2014) pages http://dx.doi.org/10.1155/2014/961067