

An Intelligent System for Mineral Prospecting Using Supervised and Unsupervised Learning Approach

Professor Anigbogu, Sylvanus Okwudili and Olanloye, Dauda Odunayo.

Abstract— Intelligent mineral prospecting with hyperspectral data involves reduction of high dimension spectral data to low dimension data that can be used for classification. In this work we used a metaheuristic method and fuzzy c-means clustering to obtain low dimension cluster center coordinates that are then used for classification using either unsupervised or supervised learning. The Unsupervised learning approach involves Kohonen self-organizing maps (SOM) which is used to obtain classes of mineral present in the data without recognizing the particular mineral. The inclusion of known labeled sample data involves the supervised learning approach, which include Adaptive Neurofuzzy Inference System (ANFIS) that learn to recognize particular minerals. R Programming language is used for implementation of the system.

Index Terms— Characterisation map, fuzzy c-mean clustering, ANFIS, Kohonen SOM

I. INTRODUCTION

Artificial intelligence(AI) may be defined as the branch of computer science that is concerned with automation of intelligent behavior [18].

It is one of the major areas of computer science that can be used to solve different types of real life problems. To be able to understand the concept of AI, one should understand the concept of intelligence. The word intelligence is not well understood or well defined. Although most of us can say confidently that we understand the meaning of the word intelligence but it is doubtful that anyone could come close to defining intelligence in a way that will be specific enough to help in evaluation of a supposedly intelligent computer program, while still capturing the validity of the human mind. Hence the problem of defining AI is that of defining intelligence.

Thus, the problem of defining artificial intelligence became one of defining intelligence itself. What is intelligence? Intelligence is a capacity of a system to think or reason in other to achieve a goal or sustain desired behavior under conditions of uncertainty without external aid or assistance.

Intelligent systems have to cope with sources of uncertainty like occurrence of unexpected event such as unpredictable changes in the world in which the system operate and incomplete, inconsistent and unreliable information available to the system for the purpose of deciding what to do next.

In fact, intelligent system has the ability to reason, learn and adapt to a particular environment. An intelligent system is a

system that is able to react promptly to a changing situation without input from a human operator.

Nikola [12], noted that an intelligent system exhibits the following behavior

- (i) They should from time to time accommodate new problem solving rules.
- (ii) They should be able to analyze themselves in term of behavior error and success.
- (iii) Once they are to interact, they should learn and improve through interaction with the environment.
- (iv) They should learn quickly from large amount of data.
- (v) They should have many base exemplar storage and retrieval capability.
- (vi) They should have parameter to present.

Agris [1] also summarized basic features of intelligent system as follows:

- (i) They have the ability to generate a new knowledge from already existing one.
- (ii) They have ability to learn.
- (iii) They have ability to sense environment.
- (iv) They should have ability to act.

A lot of research works have been carried out in the area of intelligent systems. For instance, Paul [13] developed an intelligent system to determine the best soil for different type of crops using artificial neural network. Again, Mesia [11] developed an intelligent system that can predict weather condition in South Korea using Neuro-fuzzy method.

Unlike the previous research work, the present research work was aimed at developing an intelligent system that can train hyperspectral remote sensing data set using Supervised and Unsupervised learning approach. In fact, the system has the capability to:

- (i) Classify the minerals into various classes
- (ii) Identify the type of Minerals in the particular class
- (iii) Identify the noble mineral
- (iv) Estimate the quantity or volume of minerals present in a particular area so as to advice whether it is sufficient enough for exploration or not

Mineral prospecting and exploration is a multidisciplinary task requiring a simultaneous consideration of numerous disparate geophysical, geological and geochemical dataset [8]. The size and complexity of regional exploration data available to geologist are increasing rapidly from variety of source such as remote sensing, airborne geophysics, large commercially available geological and geochemical data, [2]. This demands more effective integration and analysis of regional and various geospatial data with different format and attribute so as to be able to generate sufficient information for locating and exploring minerals. Unfortunately, various method used by different researchers are not sophisticated enough to handle this important task.

Manuscript received January 22, 2015.

Professor Anigbogu, Sylvanus Okwudili, Department of Computer Science, Nnamdi Azikiwe University, Awka, Nigeria.+2348066184124.

Olanloye, Dauda Odunayo., Department of Computer Science, Emmanuel Alayande College of Education, Oyo, Nigeria. +2348038196016

To solve this problem, an intelligent system that makes use of both supervised and unsupervised learning approach is developed.

II. ILITERATURE REVIEW

Saxena, et al [15] developed an intelligent system for emerging power system to solve the problem of power design, planning and distribution using computational intelligence technique.

This system solves the problem of using conventional method which created a lot of difficulties of derivative existence and provide optimal solutions.

Paula [14] developed a system for polyhedral approach to solve transportation problem. Jeroine & Bryan [7] used Neural Network and differential evolution for the prediction of stock returns.

Trina [17] also developed a system to forecast market volatility using Kalman filter method using

Claire [3] developed an intelligent system called swarm intelligent optimization of time capacitated Arc Routine problem using neural network.

Masao [10] in his work developed “Computing in Spatial Language Understanding Guided by Cognitively Inspired Knowledge Representation” which is a human friendly intelligent system.

Simonelli, [16] also developed an intelligent system titled “Black Scholes Fuzzy Numbers as indexes of performance”. The system was tested with data of Italian Stock Exchange to forecast the nature of the future market. Obviously, with respect to the probabilistic tree, the intelligent system is more simple and immediate to have a forecast on the financial market.

Daoxing et al [4] in their work, reviewed Gait Optimization Based on Evolution Computation using computational methods and presented other method of developing an intelligent system called Gait optimization based on evolution computation.

Xiuming Yu et al [19] used the concept of pattern recognition to generate algorithm for Discovery Access Pattern recognition from web log data. This

III. METHODOLOGY

A. Structure of Adaptive Neuro-Fuzzy Inference System

Adaptive Neuro-fuzzy Inference System (ANFIS) is a fuzzy inference system implemented in the framework of an adaptive neural network. By using a hybrid learning procedure, ANFIS can construct an input-output mapping based on both human-knowledge as fuzzy if-then rules and approximate membership functions from the stipulated input-output data pairs for neural network training. This procedure of developing a Fuzzy Inference System using the framework of adaptive neural networks is called an adaptive Neuro fuzzy inference system (ANFIS). There are two methods that ANFIS learning employs for updating membership function parameters. These include;

- (i) Back propagation for all parameters (a steepest descent method), and
- (ii) A hybrid method consisting of back propagation for the parameters associated with the input membership and least squares estimation for the parameters associated with the output membership functions.

As a result, the training error decreases, at least locally, throughout the learning process. Therefore, the more the initial membership functions resemble the optimal ones, the easier it will be for the model parameter training to converge. Human expertise about the target system to be modeled may aid in setting up these initial membership function parameters in the FIS structure [5][6]. The general ANFIS architecture is shown in Fig 1 below

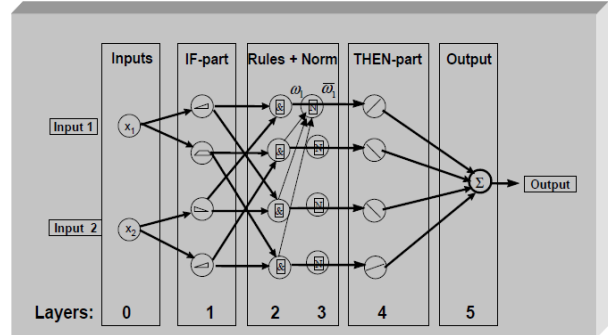


Fig 1: The general ANFIS architecture

Five network layers are used by ANFIS to perform the following fuzzy inference steps as depicted in fig 1. These are:

- (i) Input fuzzification
- (ii) Fuzzy set database construction
- (iii) Fuzzy rule base construction
- (iv) Decision making
- (v) Output defuzzification.

For instance assume that the FIS has two inputs \$x_1\$ and \$x_2\$ and one output \$y\$. For the first order Sugeno fuzzy model, a typical rule set with two fuzzy if-then rules can be expressed as:

$$\text{Rule 1: IF } (x_1 \text{ is } A_1) \text{ AND } (x_2 \text{ is } B_1) \text{ THEN } f_1 = p_1x_1 + q_1x_2 + r_1 \dots 1$$

$$\text{Rule 2: IF } (x_1 \text{ is } A_2) \text{ AND } (x_2 \text{ is } B_2) \text{ THEN } f_2 = p_2x_1 + q_2x_2 + r_2 \dots 2$$

Where \$A_1, A_2\$ and \$B_1, B_2\$ are the membership functions for the input \$x_1\$ and \$x_2\$, respectively, \$p_1, q_1, r_1\$ and \$p_2, q_2, r_2\$ are the parameters of the output function. The functioning of ANFIS is illustrated as follows:

Layer 1: Calculate Membership Value for Premise Parameter

Every node in this layer produces membership grades of an input parameter. The node output

$$o_{1,i} = \mu_{A_i}(x_1) \text{ for } i = 1,2, \text{ or } \dots 3$$

$$o_{1,i} = \mu_{B_{i-2}}(x_2) \text{ for } i = 3,4 \dots 4$$

Where \$x_1\$ (or \$x_2\$) is the input to the node \$i\$; \$A_i\$ (or \$B_{i-2}\$) is a linguistic fuzzy set associated with this node. \$O_{1,i}\$ is the membership functions (MFs) grade of a fuzzy set and it specifies the degree to which the given input \$x_1\$ (or \$x_2\$) satisfies the quantifier. MFs can be any functions that are Gaussian, generalized bell shaped, triangular and trapezoidal shaped functions. A generalized bell shaped function can be selected within this MFs and it is described as:

$$\mu_{A_i}(X_1) = \frac{1}{1 + \left| \frac{X_1 - c_i}{a_i} \right|^{2b_i}} \dots 5$$

Where \$a, b, c_i\$ are the parameter set which changes the shapes of the membership function degree with maximum value equal to 1 and minimum value equal to 0.

Layer 2: Firing Strength of Rule

Every node in this layer, labeled Π , whose output is the product of all incoming signals:

$$o_{2,i} = w_i = \mu_{A_i}(x_1) \mu_{B_i}(x_2) \text{ for } i = 1,2 \quad \dots 6$$

Layer 3: Normalize Firing Strength

The i^{th} node of this layer, labeled N , calculates the normalized firing strength as,

$$o_{2,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2} \quad i = 1,2 \quad \dots 7$$

Layer 4: Consequent Parameters

Every node i in this layer is an adaptive node with a node function,

$$o_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x_1 + q_i x_2 + r_i) \quad \dots 8$$

Where i is the normalized weighting factor of the i^{th} rule, f_i is the output of the i^{th} rule and p_i, q_i, r_i is consequent parameter set.

Layer 5: Overall Output

The single node in this layer is a fixed node labeled Σ , which computes the overall output as the summation of all incoming signals:

$$\text{Overall output} = o_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad \dots 9$$

ANFIS requires a training data set of desired input/output pair $(x_1, x_2, \dots, x_m, y)$ depicting the target system to be modeled. ANFIS adaptively maps the inputs (x_1, x_2, \dots, x_m) to the outputs (y) through MFs, the rule base and the related parameters emulating the given training data set. It starts with initial MFs, in terms of type and number, and the rule base that can be designed intuitively.

ANFIS applies a hybrid learning method for updating the FIS parameters. It utilizes the gradient descent approach to fine-tune the premise parameters that define MFs. It applies the least-squares method to identify the consequent parameters that define the coefficients of each output equation in the Sugeno-type fuzzy rule base. The training process continues till the desired number of training steps (epochs) or the desired root mean square error (RMSE) between the desired and the generated output is achieved. In addition to the training data, the validation data are also optionally used for checking the generalization capability of ANFIS.

IV. SYSTEM DESIGN

A. Features of Kohonen's Self-Organizing Map (KSOM)

In 1981, Teuvo Kohonen from Finland demonstrated how systems could be built to organize input data without being supervised or taught in any way. He called this process a **self-organizing feature map** and showed how it could be performed by a neural network. Similarities among patterns are mapped into closeness relationships on the competitive layer grid.

Generally, Kohonen's later publications in 1995 and 2001 [9] are regarded as the major references on SOM. Kohonen's description is "it is a tool of visualization and analysis of high-dimensional data". Additionally, it is useful for clustering, classification and data mining in different areas. He described as an unsupervised learning method, the key feature of which is that "there are no explicit target outputs or environmental evaluations associated with each input". During the training process, there is no evaluation of correctness of output or 'supervision'.

First, it is different from other neural networks, and it only has two layers which are input layer and output layer (or called competition layer) respectively. Every input in input space connects to all the output neurons in the map. The output arrangements are mostly of two dimensions. The Fig 2 shows below conventional 1D and 2D arrangements as described by Kohonen [9]

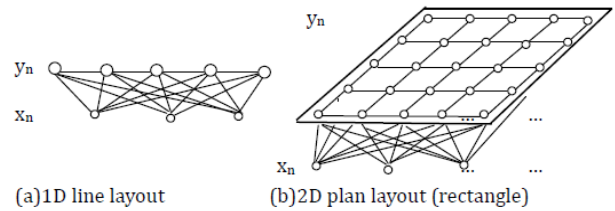


Fig 2: Conventional 1D and 2D arrangement

In Fig 2, x_n represent the input neurons in input space, y_n represents the outputs in the output space. Fig 2a shows a one dimensional arrangement in form of a line layout. Fig 2b shows a two-dimensional arrangement in form of rectangular layout. The Fig 2 shows that compared to general NN, SOM has no hidden neurons and the discrete layout of the inputs map to output space in a regular arrangement. Besides the rectangular layout, 2D SOM also has the form of hexagonal arrangement.

Next, the main process of Self-Organizing Maps (SOM) is introduced generally. According to Kohonen, the process is made up three main phases which are **competition, cooperation and adaptation**.

Competition: The output of the neuron in self-organizing map neural network computes the distance (Euclidean distance) between the weight vector and input vector. Then, the competition among the neurons is based on the outputs that they produce, where $i(x)$ indicate the optimal matching input vector x , the formula can be represented:

$$i(x) = \arg \min_j \|x - w_j\|, \quad j = 1,2, \dots, l \quad \dots 10$$

In formula 10 above, x is the input vector, w_j is the j^{th} neuron's weight vector. It uses "Nearest neighbor search", which is interpreted as proximity search, similarity search or closest point search, that can be used in finding closest points in metric spaces. The neuron j which satisfies the above condition is called the "winning neuron".

Cooperation: the winning neuron is located at the center of the neighborhood of topologically cooperating neurons. The winning neuron tends to activate a set of neurons at lateral distances computed by a special function.

The distance function must satisfy two requirements: 1) it is symmetric; 2) it decreases monotonically, as the distance increases. A distance function $h(n,i)$ which satisfies the above requirements is Gaussian:

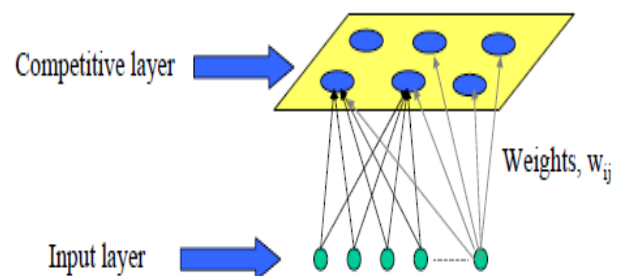


Fig 3: Segment of unsupervised learning system (KSOM)

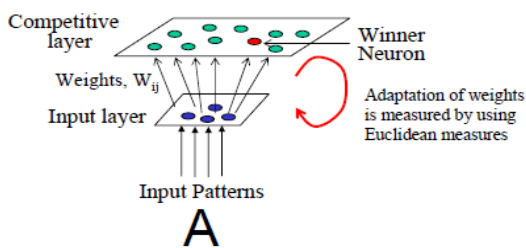


Fig 4: Adaptation of weight in KSOM

Adaption: it is in this phase that the synaptic weights adaptively change. Since these neural networks are self-adaptive, it requires neuron j 's synaptic weight w_j to be updated toward the input vector x . All neurons in the neighborhood of the winner are updated as well in order to make sure that adjacent neurons have similar weight vectors. The following formula state the weights of each neurons in the neighborhood of the winner which are updated:

$$h(j, i) = \exp(-dj, i/2\sigma^2) \quad \dots 11$$

In formula 11 above, $h(j, i)$ is the topological area centered around the winning neuron i . The dj, i is the lateral distance between winning neuron i and cooperating neuron j , and σ is the radius influence.

$$h(i, j) = \exp(w_j + \eta h(j, i) * (x - w_j)) \quad \dots 12$$

In formula 12, η is a learning rate, i is the index of winning neuron, w_j is the weight of the neuron j . The $h(j, i)$ function has been shown in equation 12.

These three phases are repeated during the training, until the changes become less than a predefined threshold.

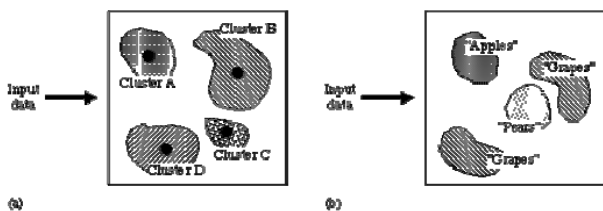


Fig 5(a,b): Cluster processing using KSOM

5(a, b) By observing the input patterns, KSOM reorganizes them by clustering similar patterns into groups

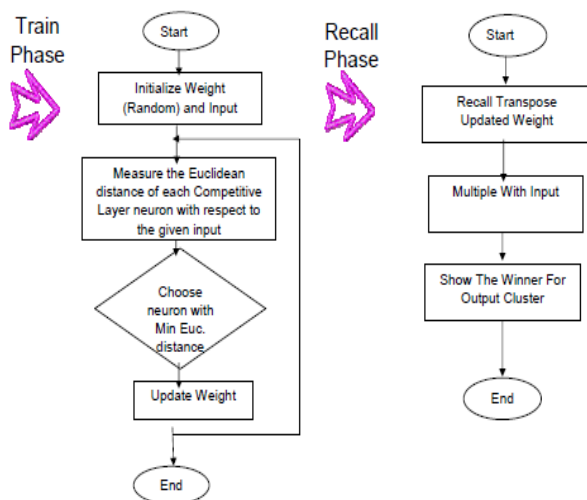


Fig 6: Simple KSOM Flowchart

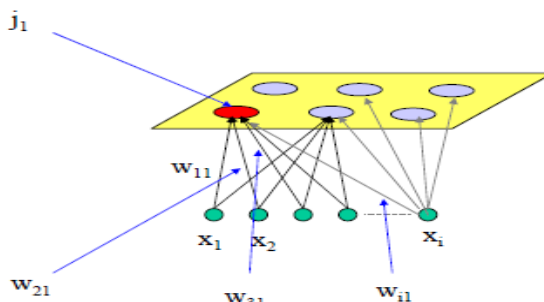


Fig 7: Diagrammatic illustration of the KSOM Algorithm

A. Basic Principles of the KSOM

The KSOM neural network is basically a single-layer feed forward network.

Step 1: Set up input neuron matrix, In_X and In_Y .

Thus, total number of input neurons,

$$I = In_X * In_Y.$$

We used $i = 0$ to $I-1$ for numbering the neurons in this layer. Eg., X_j is the label for the input neurons i.e. X_0 to X_{I-1}

Step 2: Set up competitive layer matrix, Out_X and Out_Y .

For simplicity $Out_X = Out_Y$ (such that we have a square map). Therefore, total number of competitive neurons, $J = Out_X * Out_Y$

We used $j = 0$ to $J-1$ for numbering the neurons in this layer

Step 3: Initialize connection weights (randomize)

between input layer neurons and competitive layer neurons, W_{ij} .

Set initial topological neighborhood parameters, d_0 .

Usually,

$$d_0 \leq \frac{Out_X}{2}$$

Set initial learning rate parameter, α_0

(usually between 0.2 to 0.5)

Set total number of iterations, T (usually 10,000).

Start with iteration $t = 0$.

Apply the first pattern to the input of the KSOM.

Step 4: Compute the winning neuron (j_c) in the competitive layer which is the minimum Euclidean distance from input layer to competitive layer such that

First, for each j (from $j = 0$ to $J-1$), compute the Euclidean distance as follows:

$$\|E(j)\| = \sqrt{\sum_{i=0}^{i=I-1} (w_{ij} - x_i)^2} \quad \dots 13$$

Then compare all these distances i.e. from

$$\|E(0)\| \text{ to } \|E(j)\|$$

and find the minimum distance $\|E(j_c)\|$ which is the winner neuron, j_c .

$$\|E(j_c)\| = \min \|E(j)\| \quad \dots 14$$

Calculate Euclidean distance for each Competitive layer neuron

- When an input pattern is presented, each unit in the 1st layer takes on the value of the corresponding entry in the input pattern.
- The 2nd layer units then sum their inputs and compete to find a single winning unit.

$$\|E(j1)\| = \sqrt{(w_{11} - x_1)^2 + (w_{21} - x_2)^2 + \dots + (w_{ij} - x_i)^2} \dots (15)$$

Step 5: Update weight for each connections i.e. For all neurons j within a specified neighbourhood of J, and for all i :

$$W_{ij}(new) = w_{ij}(old) + \Delta w_{ij}(new) \dots (2)$$

Where

$$\Delta W_{ij} = \begin{cases} \alpha_t(x_i - w_{ij}(old)) & \text{if unit is in neighbourhood } d_t \\ 0 & \text{otherwise} \end{cases}$$

Step 6: Update learning rate α_t such that:

$$\alpha_t = \alpha_0 \left(1 - \frac{t}{T}\right) \dots (16)$$

Step 7: Reduce radius of topological neighborhood at specified times:

$$d_t = \text{int} \left[d_0 \left(1 - \frac{t}{T}\right) \right] \dots (17)$$

Step 8: Increase iteration t: t=t+1

Repeat Steps 5 to 8 until t=T

Step 9: Repeat with next pattern chosen randomly

(Do Steps 4-9)

V. SYSTEM IMPLEMENTATION

A. Computer simulations of self-organization in the Kohonen SOM.

1. Initialize weights to 0.5 + 10 % randomized value.
2. 2 input vectors, X1 and X2 with several scores of entries between the range of 0 and 1.
3. The Fig 7 given, shows a plot of initial weights, w_{ij} .
4. Each unit in the competitive layer shows a point on this graph
5. The coordinate values of this point are the values of the incoming weights for the unit, thus w_{i1} , w_{i2} are plotted for each competitive unit j.
6. All pairs of units in the competitive layer that are adjacent are connected.
7. This will allow us to see how the pattern of weights changes as the network organization evolves during training.

B. Data

The test data consists of specter data for Cuprite, Nevada. The data consists of 600 by 320 pixels with 357 band spectrum ranging from 0.4 μm to 2.5 μm . Fig. 8 shows the data cube visualization while Fig. 9 shows a colormap slice (a band) of the given data. Fig 10 shows the main interface window of the system and Fig 11 is the 3D plot of some bands of input data.

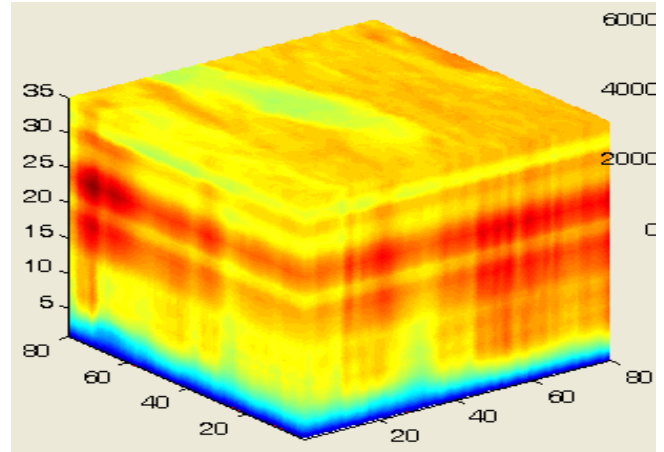


Fig 8: Data cube image of hyperspectral data

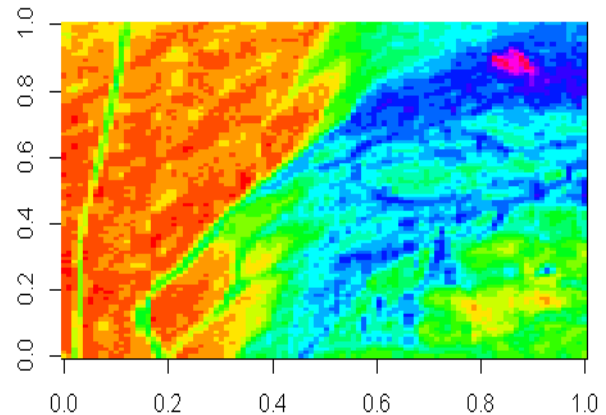


Fig9: A band (slice) of hyperspectral data

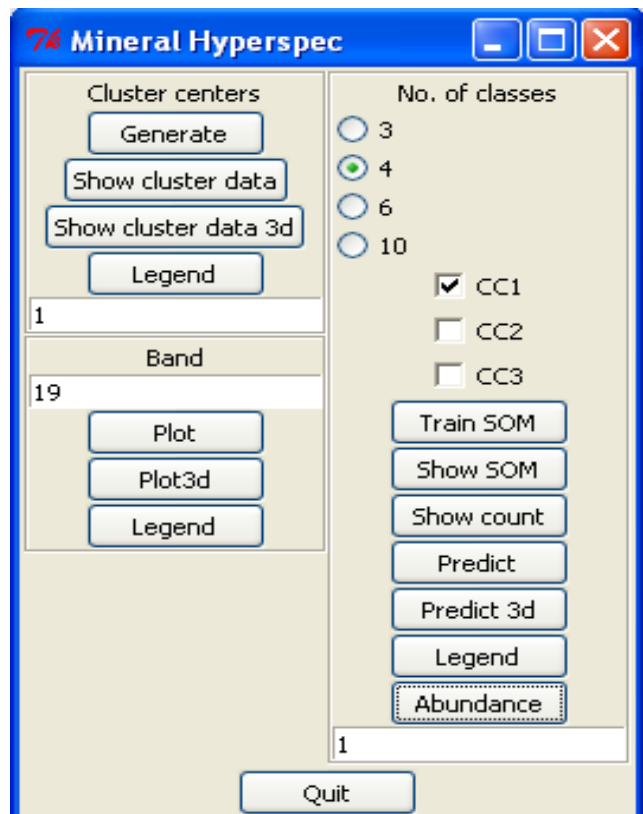


Fig 10: The main interface window of the system

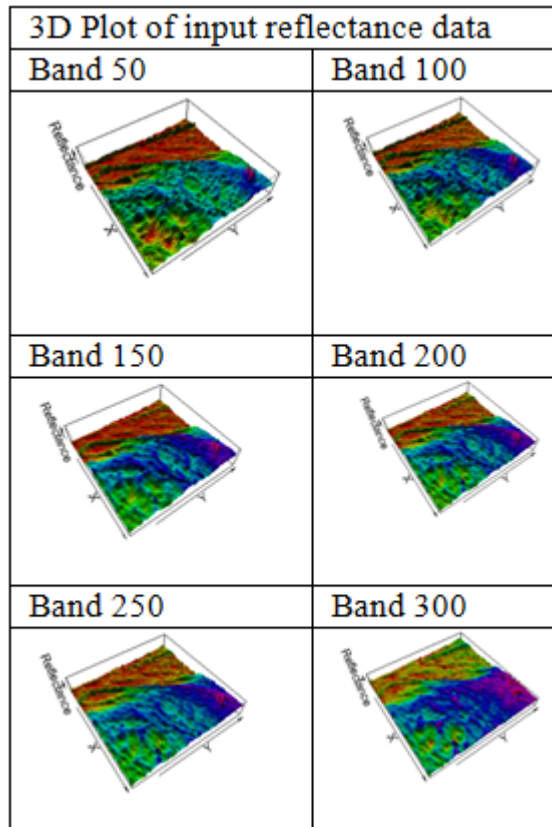


Fig 11: 3D Plot of some bands of the input data

C. Processing of spectrum for each pixel.

The spectrum for each pixel is selected in turn, displayed in Fig 12 and turned into characterization map as explained in the design and implementation. The characterization map for a particular pixel is shown in Fig 13. The characterization map is then clustered to obtain 3 cluster centers marked in Fig 13. The cluster center data for each pixel is thus calculated in turn and stored in a cluster center data structure for file storage or further processing. The cluster center data (Fig 14) distill the essential features for classification and recognition of mineral classes in the given data.

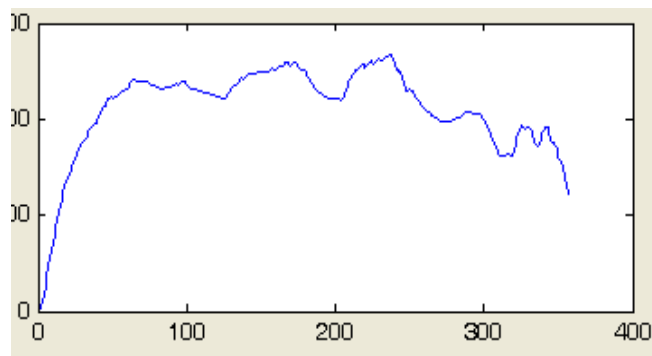


Figure 12: The spectrum of a pixel

The x and y coordinate data for each of the 3 cluster center for each the pixel is shown below with their aggregate (Fig 15). The aggregate maps show clearly the correctness of the method in generating mineral predictor maps. The obtained predictor maps now form the basis for mineral classification with intelligent system. In absence of a priori knowledge about the kinds of minerals available in the given

area, We used self-organizing kohonen maps to obtain classes of minerals in the given data. We select some samples from the cluster center data and used it to train Kohonen network which predicts the classes of minerals. The program is run for different numbers of output neurons and epochs and the results are shown variously in Figs 14-17.

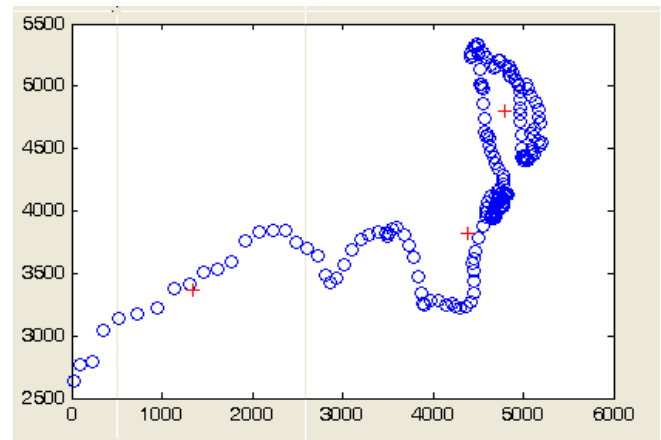


Figure 13: Clustered image of a pixel with 3 cluster centers shown in red

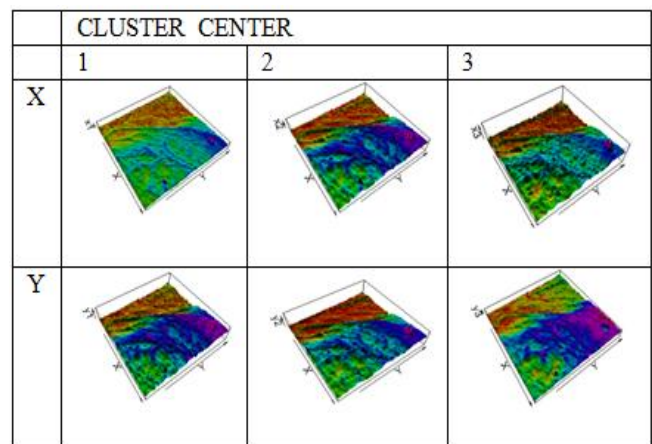


Fig 14: 3D Plots of the cluster center data.

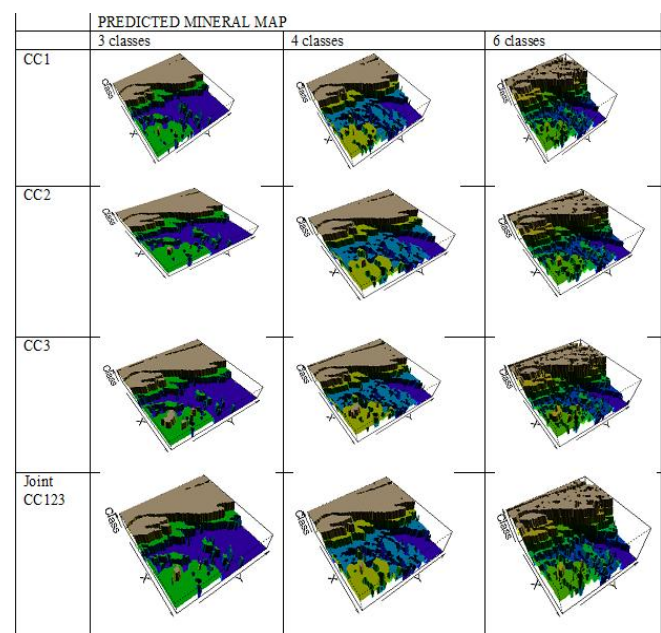


Fig 15: 3D Plots of the predicted mineral maps

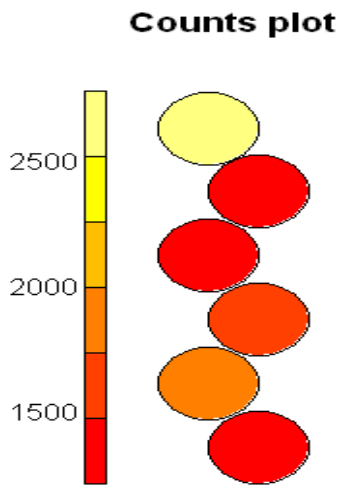


Fig 16: Plot of learning SOM

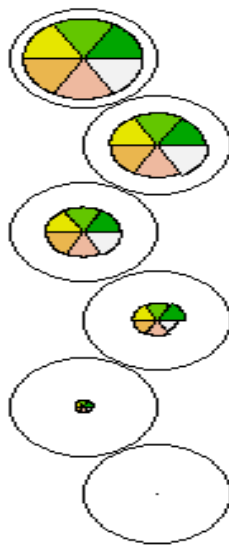


Fig 17: Class count plot of trained SOM

VI. RESULTS AND DISCUSSIONS

A. Supervised learning with ANFIS

It is furthermore noted that to create a system that learns to recognize classes of mineral with time as discussed under the design and implementation section, a test of this methodology is carried out using an sugeno-type ANFIS network to learn to classify minerals overtime. The ANFIS network consists of six input node and one output node. The cluster center coordinate data are used as input with the aggregate value from the kohonen network classification used as output. The scheme of the network is shown in Fig 18. The initial training data set is shown in Fig 19. The question arises as to optimum choice of the number of membership functions required. We wish to minimize the number of membership functions so as to reduce the size of our network. But initial test runs show that two membership functions do not suffice as the network did not converge and mixes most of the targets during test as shown in the comparison Figs 20. Therefore we have to use three membership functions and therefore shall discuss the results for three membership functions, fig 20.

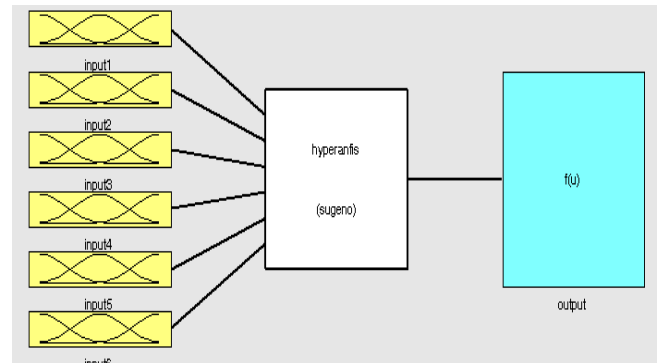


Fig 18: Scheme of the ANFIS network

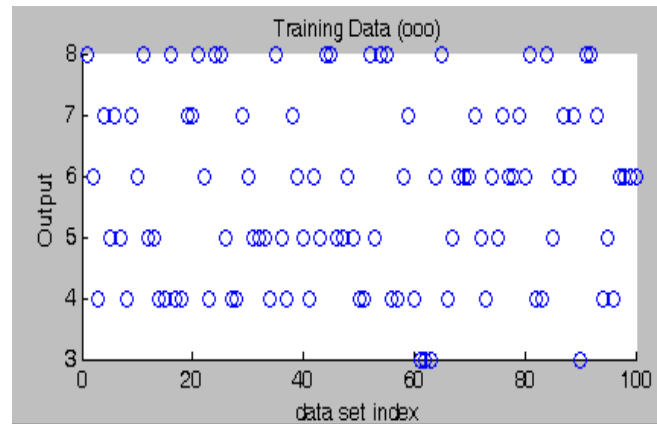


Fig 19: Training data set for ANFIS network

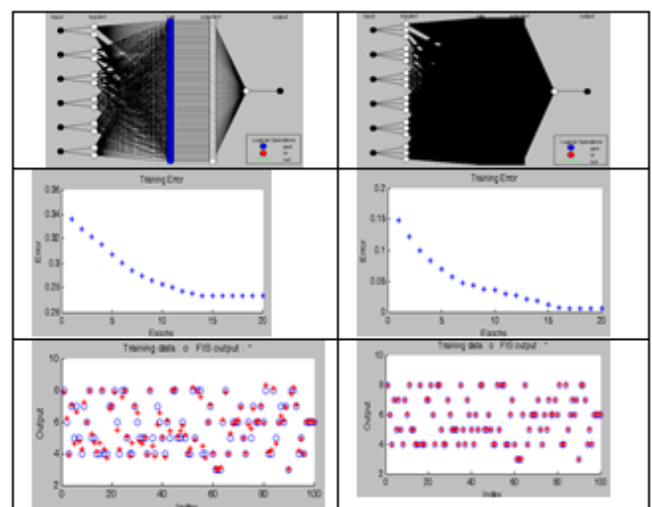


Fig 20: Comparison of performance of ANFIS network with 2 or 3 membership functions

With three membership function, the network learnt to classify the samples with approximately zero error level within 20 epochs as shown in Fig 20. Test with original training sample shows a 100% score as shown in Fig 20.

VII. NOVEL MINERAL

In Fig 20, it can be noticed that minerals in a particular class are aligned on a straight line. If a sample for a novel mineral (not yet learned mineral) is presented to the network, the output for it will not fall on an already existing line. This is an indication that the presented sample is a novel mineral. In

such case, further samples needs to be presented and the system allowed to learn the new sample.

The ANFIS network learns and store knowledge as rule base.

This rule base can be variously visualized as shown in fig21 or as rule surfaces that shows how inputs combine with one another as shown in Fig 22. Also the shape of the membership functions change as the system learns as depicted in Fig 23.

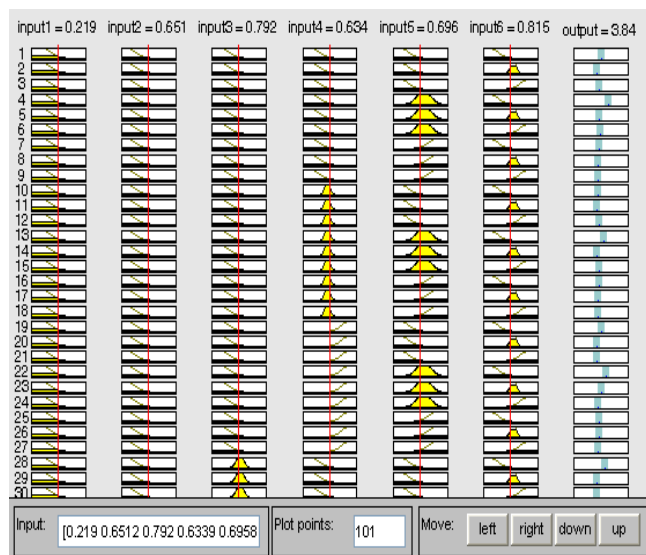


Fig 21: Rule base of the trained network

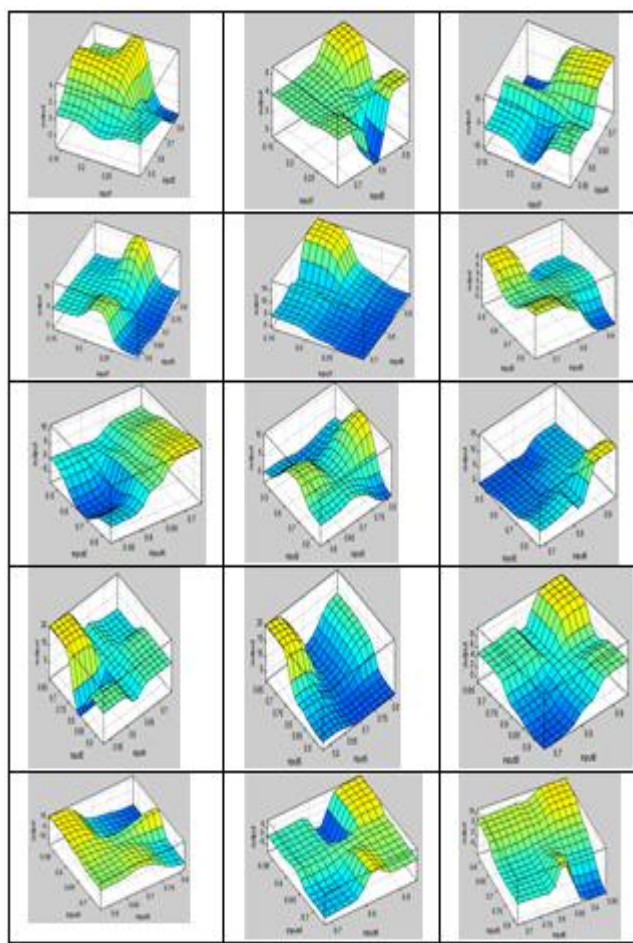


Fig 22: Rule surfaces of the trained network

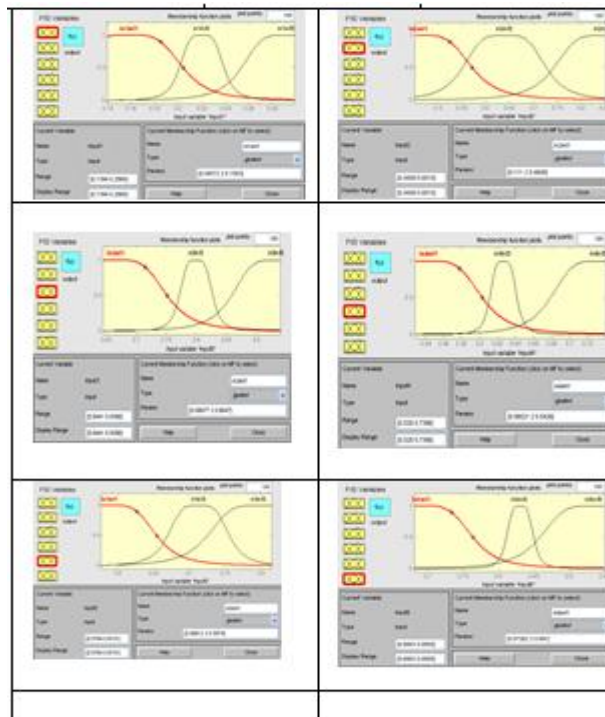


Fig 23: Membership functions of the trained network

1) Mineral reserve estimation

Subsequent to the detection and identification of a particular class of mineral, it is desirable to estimate the level of availability of such mineral in the given area. In particular the abundance of such mineral in the given area needs to be measured. This information is important as it will form the basis for further prospecting works in the given area. The initial estimate must justify the investment required for further search for the mineral. Thus our task is to develop an algorithm capable of estimating the level of abundance of the detected mineral.

In this task, we proceed as follows: the class of mineral represented by each pixel is available from the classification and identification data. Thus, one can count the number of pixels that belong to a particular class of mineral in a given area. The count is then taken as a percentage of the total number of pixels in the given area multiplied by the given area. This algorithm will work even in case of volume data as the number of pixel in given volume data is proportional to the volume of the sought mineral.

$$\text{Reserve estimate} = \text{No. of pixel in a class} / \text{Total no. of pixel} \times \text{area}$$

$$\text{Reserve estimate} = \text{No. of pixel in a class} / \text{Total no. of pixel} \times \text{X area}$$

Abundance Volume Estimation

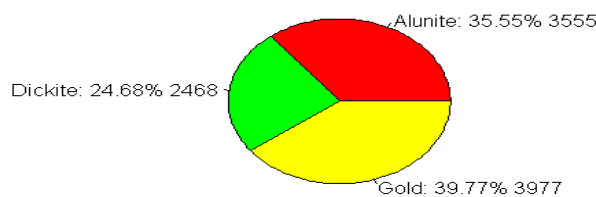


Fig 24: Abundance estimate for 3 classes

Abundance Volume Estimation

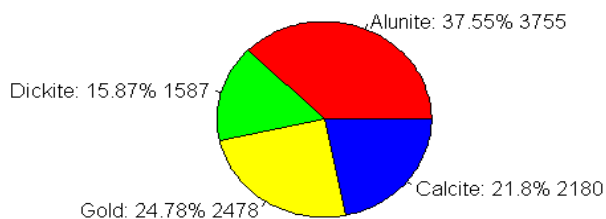


Fig 25: Abundance estimate for 4 classes

Abundance Volume Estimation

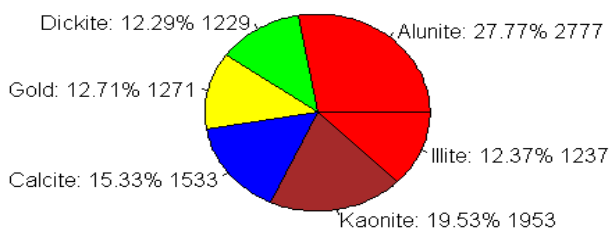


Fig 26: Abundance estimate for 6 classes

To ensure a continue supply of mineral to meet the need of a growing population so as to ensure continuity of life of man, animals and plants, there is a need for this type of intelligent system. Apart from the fact that the system will have zero impact on the earth surface during mineral prospecting, it will also go a long way to boost the mining industry.

If the mining industries are boosted, it provides the basic needs for the people; they will be good source of income for the government, individuals, parastaters e.t.c. This will also provide better job opportunities for the citizens. In fact, it will go a long run to boost the economy of the country at large.

Again, the product of this research which could be used in any part of the world where hyperspectral data is made available on the satellite justifies the huge amount of money being spent by the Government to lunch a satellite.

VIII. CONCLUSION

An intelligent system was developed for mineral prospecting. The system was developed using unsupervised and supervised learning approach. With the use of Kohonen Self Organizing Map, hyperspectral data were classified into various classes with each class representing a particular type of mineral. Adapted Neurofuzzy Inference System was used to identify each of the classes of minerals. The relative percentage of each of the mineral was also indicated by the intelligent system. The system is also capable of identifying novel minerals. This research work will go a long way to boost the mining industry in the area of mineral prospecting.

REFERENCES

- [1] Agris Nikitenko (2006). Hybrid Intelligent System Development and Implementation. *Summary of Doctoral Thesis*. RIGA TECHNICAL UNIVERSITY, Riga.
- [2] Brown, W. Groves, D. & Gedeon, T. (2010). The Use of Neural Network for Mineral Potential Mapping. *Natural Resources Research*, Vol.18, No. 10, 120-137.
- [3] Claire Heaney & Darragh Needham (2006). Swarm Intelligence Optimisation of Time Capacitated Arc Routing Problems. *UCD School of business*, College of Business and Law, University College, Dublin, Belfield.
- [4] Daoxiong Gong, Jie Yan, and Guoyu Zuo (2010). A Review of Gait Optimization Based on Evolutionary Computation. *Applied Computational Intelligence and Soft Computing*, Volume 2010 (2010), Article ID 413179, doi:10.1155/2010/413179.
- [5] Jang, J.S.R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference Systems, *Trans. Systems, Man & Cybernetics*. Vol. 23, pp.665—685
- [6] Jang, J.S.R., Sun, C.T., and Mizutani, E. (1997). Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence. *Prentice Hall*.
- [7] Jerome Perera & Bryan O'Hora (2005). The use of Neural Networks and Differential Evolution for the Prediction of Stock Returns. *UCD School of business*, College of Business and Law, University College, Dublin, Belfield.
- [8] Knox-Robinson, C.M., Groves, D. I., GoldFard, R.J., Ojala, V.J., Gardoll, S.J., Yung,G.Y., Holyland, R.W. (2003). Late-Kinematic timing of Orogenic Gold Deposits and Significance for Computer Based Exploration Techniques with Emphasis on the Yilgarn Block, Western Australia. *Ore Geology Reviews*. Vol. 17, pp. 1-38.
- [9] Kohonen, T. (1995). Self-Organizing Maps. Springer-Verlag, Heidelberg, Germany.
- [10] Masao Yokota, (2012). Aware Computing in Spatial Language Understanding Guided by Cognitively Inspired Knowledge Representation. *Applied Computational Intelligence and Soft Computing*, Volume 2012 (2012), Article ID 184103, doi:10.1155/2012/184103.
- [11] Mesia (2006). Whether Prediction using Neuro Fuzzy Method. A Seminar Paper Presented at Emmanuel Alayande College of Education, 2010. *National Conference*.
- [12] Nikola Kasabov (2007). Brain, Gene and Quantum inspired Computational Intelligence: Challenges and Opportunities. *Challenges for Computational Intelligence*. DOI 10.1007/978-3-540-71984-7_9, Vol 63, pp193-219
- [13] Paul Adeleke (2009). Intelligent System for Determining Soil for a Particular Crop Using Neural Network. *Agricultural Research*. Vol. 9, No 4, pp 75-84.
- [14] Paula Carroll (2005). Polyhedral Approach to Solving TSP Problems. *UCD School of business*, College of Business and Law, University College, Dublin, Belfield.
- [15] Saxena, D., Singh, S.N. and K.S. Verma (2010). Application of Computational Intelligence in Emerging Power Systems. *International Journal of Engineering, Science and Technology*. Vol. 2, No. 3, 2010, pp. 1-7. www.ijest-ng.com.
- [16] Simonelli, M. R (2010). Black-Scholes Fuzzy Numbers as Indexes of Performance. *Applied Computational Intelligence and Soft Computing*, Volume 2010 (2010), Article ID 607214, doi:10.1155/2010/607214.
- [17] Trina McNicholas and Niall Cassidy (2004). Forecasting Market Volatility using Kalman Filter methods. *UCD School of business*, College of Business and Law, University College, Dublin, Belfield.
- [18] Vandal Smith (2010). Application of Neural Network in Whether Prediction. "The Pacesetter ". Vol. 10, No 3, 25-35.
- [19] Xiuming, Yu, Meijing, Li, Taewook, Kim, Seon-phil Jeong, and Keun Ho Ryu (2012). An Application of Improved Gap-BIDE Algorithm for Discovering Access Patterns. *Applied Computational Intelligence and Soft Computing*, Volume 2012 (2012), Article ID 593147, doi:10.1155/2012/593147.