http://dx.doi.org/10.14257/ijhit.2016.9.11.19

Classification of Medium Resolution Satellite Imageries using Artificial Neural Network and Swarm Intelligence

Atijosan A[1]*, Badru R[2], Babalogbon A[3] and Alaga T[4]

[1,2,3,4]Cooperative Information Network, National Space Research and Development Agency, Obafemi Awolowo University Campus, Ile-Ife, Osun State, Nigeria

[1]bimbo06wole@yahoo.com, [2]sunkanmi@yahoo.com, [3]titaniumdenjin@yahoo.com, [4]alagayomi@yahoo.com*

Abstract

This work proposes the application of a hybrid Particle Swarm Optimization (PSO) with Levenberg Marquardt Back-Propagation (LMBP) algorithm to train Artificial Neural Networks (ANNs) for classification of medium resolution multispectral satellite imageries. ANNs have been widely used in satellite image classification and have been shown to outperform traditional classifiers in many situations. However the Back Propagation (BP) algorithm traditionally used in training ANN suffer the problem of local minima entrapment, thus affecting the accuracy and performance of the ANN classifier. A hybrid combination of PSO and LMBP algorithm is applied to resolve the aforementioned problem and enhance the accuracy and performance of the ANN classifier. To investigate the performance of the proposed method, medium resolution multispectral satellite imagery was classified using the proposed classifier and its performance compared with that of conventional LMBP and Scaled Conjugate Back-Propagation (SCBP) trained ANN classifier. Results obtained shows that the hybrid PSO and LMBP trained ANN classifier out performs the conventional LMBP and SCBP trained ANN classifier and achieves ≈95% accuracy on the test medium resolution satellite imagery.

Keywords: *Satellite Image, Artificial Neural Networks, and Particle Swarm Optimization*

1. Introduction

Thematic maps derived from remotely-sensed satellite images are invaluable sources of crucial information for various applications such as agriculture, modelling of environmental variables, understanding habitats distribution, planning, monitoring and management of natural resources, (Gomez *et al*, 2016; Ahmen and Al-Noman, 2015). Image classification, aims at categorizing all pixels in the satellite image into various land cover classes which can then be used to produce thematic maps, like land use/cover present in the satellite image (Agrawal and Bawane, 2015). The accuracy and efficiency of classification techniques used to produce these thematic maps are crucial (Gomez *et al*, 2016; Anchang 2016; Meher 2015), as these maps, provide the basis for deciding and implementing policies and plans for sustainable development at the local, regional and global scale.

Artificial Neural Network, are being widely used as an alternative to traditional statistical models because they have the notable ability to derive meaning from complicated or imprecise data and can be used to extract patterns and detect trends that are too complicated to be recognized by either humans or traditional computing techniques (Chen *et al* 2015). They have been reported to yield comparable or superior accuracy compared to statistical classifiers (Agrawal and Bawane, 2015). They have found applications in wide variety of problems such as data processing, classification, regression analysis, time series prediction and pattern recognition, arising from a variety of disciplines, including mathematics, computer science, and engineering (Melo and Watada, 2016; Chen *et al* 2015; Jacovides *et al* 2015).

Feedforward Neural Networks (FNNs) are the most popular ANNs used in practical applications (Chen *et al* 2015) and have been found suitable for supervised classification of multispectral satellite images (Agrawal and Bawane, 2015). The training process is an important aspect of a FNN model and performance of FNNs are dependent on the success of the training process. The aim of the training phase is to minimize a cost function defined as a mean squared error (MSE), or a sum of squared error (SSE), between its actual and target outputs. This is achieved by adjusting the FNNs weights and biases. The method most commonly used for finding the optimum weight and biases combination of FNNs is the Back Propagation algorithm (BP) (Melo and Watada, 2016; Das *et al*, 2014; Rumelhart *et al*, 1986). Although, the Levenberg Marquardt (LM) algorithm is very efficient in speeding up the convergence rate of BP algorithms, but being a gradient based algorithm it still suffers from local minima entrapment which may lead to failure in finding a global optimal solution (Chen *et al* 2015; Nawi *et al*, 2014; Zhang *et al*, 2007).

From literature, the use of novel heuristic optimization methods (global optimization) or evolutionary algorithms is a popular solution to enhance the problems of BP-based learning algorithms (Cao *et al*, 2016). Global search optimization techniques have the ability to adjust the weights for neural networks to avoid the local minima problem (Melo and Watada, 2016; Garro and Vázquez, 2015; Das *et al*, 2014). Although there is no one size fits all solution, Particle Swarm Optimization (PSO) algorithms have been found to be efficient and practical in finding the optimal weights for a given network, thereby reducing local minimal entrapment (Chen *et al* 2015; Das *et al*, 2014; SeyedAli *et al*, 2012) and enhancing the accuracy of the classification process. This study therefore applies a hybrid PSO-LMBP algorithm to train a FNN for medium resolution multispectral satellite image classification. This is done with the aim of enhancing the accuracy of information obtained from thematic maps produced from medium resolution multispectral satellite imageries. To the best of our knowledge, application of hybrid PSO-LMBP algorithm for the classification of medium resolution satellite imageries has not been reported in literature.

The rest of the paper is organized as follows: Section 2 and 3 present a brief introduction to ANN's and PSO algorithms. Section 4 discusses the hybrid PSO-LMBP training of FNN for medium resolution satellite image classification. Section 5 discusses the preparation of the datasets used for the simulation. Simulation results are provided in section 6. Finally, section 7 concludes the paper.

2. Artificial Neural Networks

An ANN is a system that performs a mapping between input and output patterns that represents a problem (Garro and Vázquez, 2015). Typically, ANN structures are composed of three different layers: input, hidden (one or more) and output layers (Martinel *et al* 2015). These layers are interconnected by links called weights. The operations of a FNN can be explained as follows; from figures 1 and 2, the input layer is composed of input units $(x_i = 1, \ldots\ldots n)$, these values are transferred to the hidden layer units where the input unit values are multiplied by the weights that connect that unit with the hidden unit. Summation of all the weights connected to the hidden unit minus its threshold θ_j is shown in equation 1:

$$z_{inj} = \sum_{i=0}^{n} x_i w_{ij} - \theta_j \tag{1}$$

where w_{ij} is the weight of the ith unit in input layer jth, the threshold is denoted as θ_j and the ith input unit is x_i. The activation function $f(z_{inj})$ is calculated by using Eqs. 2

and 3, where z_{inj} is the sum of all the input weights multiplied by the input unit value in the hidden layer:

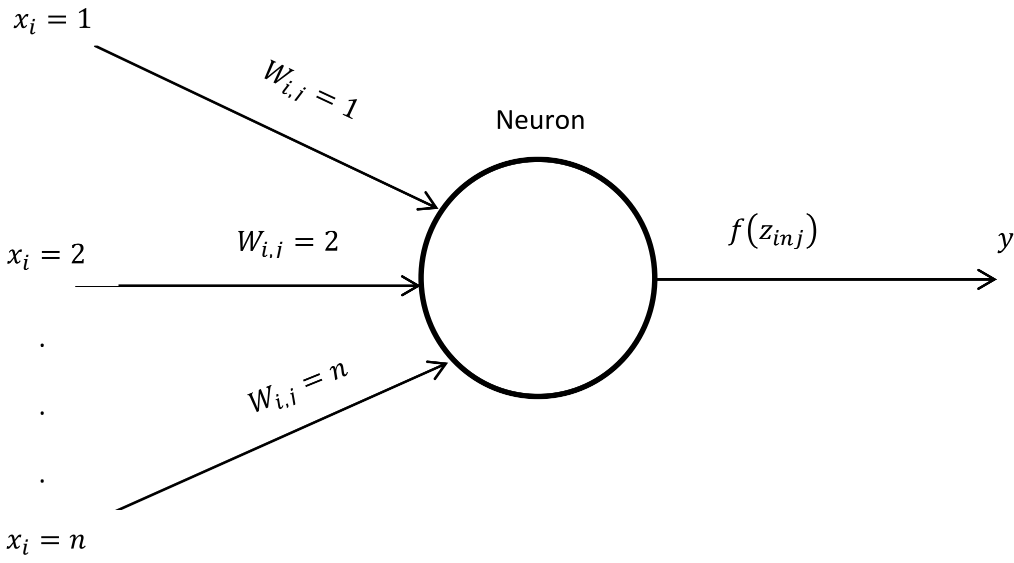

Figure 1. Typical Neuron in a Feed Forward Network

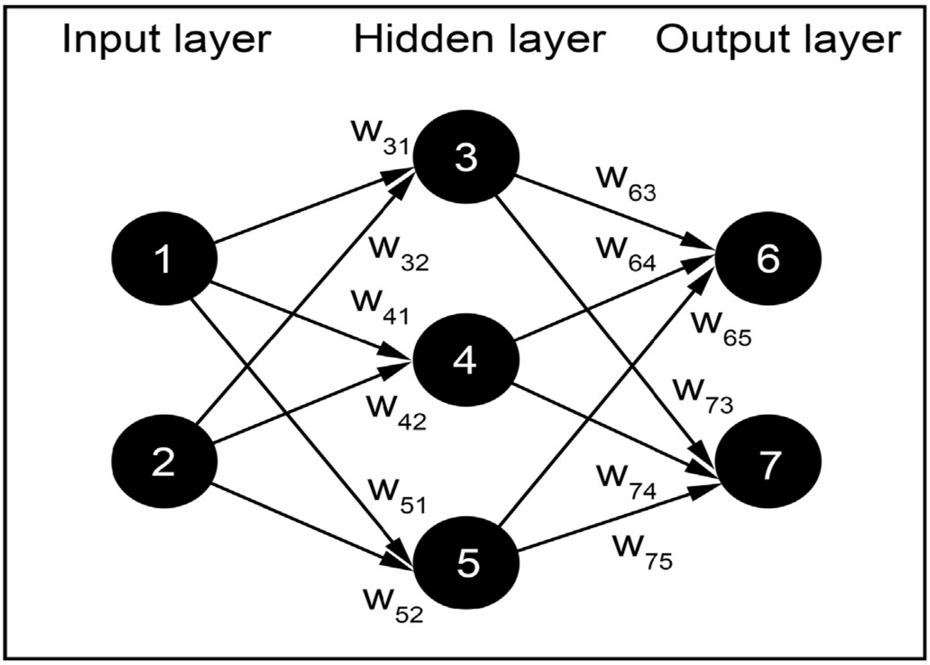

Figure 2. Neural Network Structure (Source: Melo and Watada, 2016)

$$f(z_{inj}) = \frac{1}{1+\exp(-(z_{inj}))} \qquad (2)$$

$$z_j = f(z_{inj}) \qquad (3)$$

The summation of the hidden units value (z_{inj}, $j = 1, \ldots., H$) multiplied by its corresponding weights w_{jk} minus its threshold θ_k is calculated using equation (4):

The next step is to calculate the output unit value $(y_k, k = 1, \ldots.,0)$ by applying a sigmoid function (equation 5) using equation 6:

$$y_k = f(y_{ink}), \tag{5}$$

$$y_k = f(y_{ink}) = \frac{1}{1+exp^{(-y_{ink})}} \tag{6}$$

The training error E can be calculated by the difference between the target value that corresponds to the input unit value and the actual value at each output unit:

$$E = \frac{1}{m}\sum_{i=0}^{m}(y_{ik} - Y_{ik})^2, \tag{7}$$

where m is the number of categories (number of units in the output layer), y_{ik} is the current output of an output unit k and Y_{ik} is the proper output of this unit. The sum of net partial errors for the whole training set provides the total error E of the network. The error is then back-propagated and weights are altered to minimize it. This process is repeated until the computed error drops below a predetermined value or the number of iterations exceeds a predefined maximum. This is where the chosen training algorithm is vital, because the training algorithm helps to finding an optimal set of weights and biases that will give desired values at the network's output when presented with different patterns at its input (Chen *et al*, 2015; Xu and Zhang, 2014).

3. Particle Swarm Optimization

PSO is a population based evolutionary computation technique proposed by Kennedy and Eberhart (Mirjalili *et al*, 2012). The algorithm is inspired by observations of social and collective behaviour of bird flocking and fish schooling in search of food and survival (Garro and Vazquez, 2015, Das *et al* 2014). The algorithm works by initializing a flock of birds randomly over the searching space, where every bird is called a particle. These particles fly with a certain velocity and find the global best position after some iteration. At each iteration, each particle can adjust its velocity vector, based on its momentum and the influence of its best position p_i(cognitive component) as well as the best position of its neighbours p_g (social component) and then compute a new position that the particle is to fly to (Zhang, 2007).

PSO can be modelled mathematically as follows (Garro and Vazquez, 2015):

$$v_i(t+1) = wv_i(t) + c_1r_1\big(p_i(t) - x_i(t)\big) + c_2r_2\left(p_g(t) - x_i(t)\right) \tag{8}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{9}$$

where v_i is the velocity of particle i at iteration t,; c_1 and c_1 are acceleration coefficients; r_1 and r_2 are uniformly distributed random numbers between (0,1); p_i(cognitive component) is the particle own best position; p_g(social component) is the best position of a particle in a population; x_i is the current position of particle i at iteration t; w is a weighting function or inertia weight that determines the influence of the current velocity on the subsequent velocity and can be represented as (Sallama, 2014)

$$w = w_{max} - \frac{w_{max}-w_{min}}{iter_{max}} \times iter_i \tag{10}$$

where w_{max} is initial weight; w_{min} is final weight; $iter_{max}$ is maximum iteration; $iter_i$ is current iteration.

The first part of (8) $wv_i(t)$ provides exploration ability for PSO, the second and third parts, $c_1r_1\big(p_i(t) - x_i(t)\big)$ and $c_2r_2\left(p_g(t) - x_i(t)\right)$ represents private thinking and collaboration of particles respectively. The PSO starts by randomly placing the particles in a problem space. After each iteration, the velocities of the particles are calculated using equation 8. After defining the velocities, the positions of particles can be updated using equation 9. The process of changing particles positions will continue until an end criterion is met (Mirjalili, *et al*, 2012).

4. Hybrid PSO-LMBP Algorithms for Training FNN's

Generally, when training FNNs using a heuristic algorithm, the heuristic algorithms are used for finding a combination of weights and biases which provide the minimum error for the network (Mirjalili *et al*, 2012). When using the PSO algorithm for training ANN's, every particle represents a set of weights and biasis and the PSO algorithm searches for the best combination of weights and biases that provides a minimum error for the ANN. The searching process, as adapted from Zhang *et al*, 2007 is as follows, initialize a group of random particles and update the particles using equations 8 and 9 until a new generation set of particles is generated (each particle represents a possible solution). Those particles are then used to search the global best position in the solution space. Finally the LMBP algorithm is used to search around the global optimum. This enables the hybrid algorithm to find an optimum more quickly. The ideal behind the hybrid algorithm is to use PSO to search for a global optimistic result. Thereafter the LMBP is used to find a local optimistic result among the global optimistic results. In this way both algorithms complement each other; PSO is good for global search but poor for local searches, while LMBP has a strong ability to find local optimistic results but its ability to find global optimistic results is weak (Zhang *et al*, 2007). In order to design the PSO-LMBP for a FNN the following elements need to be defined. First, a fitness function using the error of the FNN should be defined to evaluate particle fitness. Second, an encoding strategy should be defined to encode the weights and biases of the FNN. The elements are described as follows;

4.1. Fitness Function

Fitness function of the *i*th training sample is defined as (Melo and Watada, 2016; Zhang *et al*, 2007)

$$fitness(X_i) = E(X_i) \tag{11}$$

E is the training error as earlier defined in section 2.

4.2. Encoding Function

Encoding strategy: this is used to represent the weights and biases of the FNN. At this stage every particle is encoded as a matrix. Decoding using this method is highly efficient (Mirjalili *et al*, 2012; Zhang, 2007). An example of this encoding for the FNN of Fig. 3 is provided as follows:

$$particle(:,:,i) = [W_I, B_1, W_2', B_2] \tag{12}$$

$$W_1 = \begin{bmatrix} w_{13} & w_{23} \\ w_{14} & w_{24} \\ w_{15} & w_{25} \end{bmatrix},\ B_1 = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}, W_2' = \begin{bmatrix} w_{36} \\ w_{46} \\ w_{56} \end{bmatrix}, B_2 = [\theta_4]$$

Where W_1 is the hidden layer weight matrix, B_1 is the hidden layer bias matrix, W_2' is the transpose of W_2, W_2 is the output layer weight matrix and B_1 is the hidden layer bias matrix.

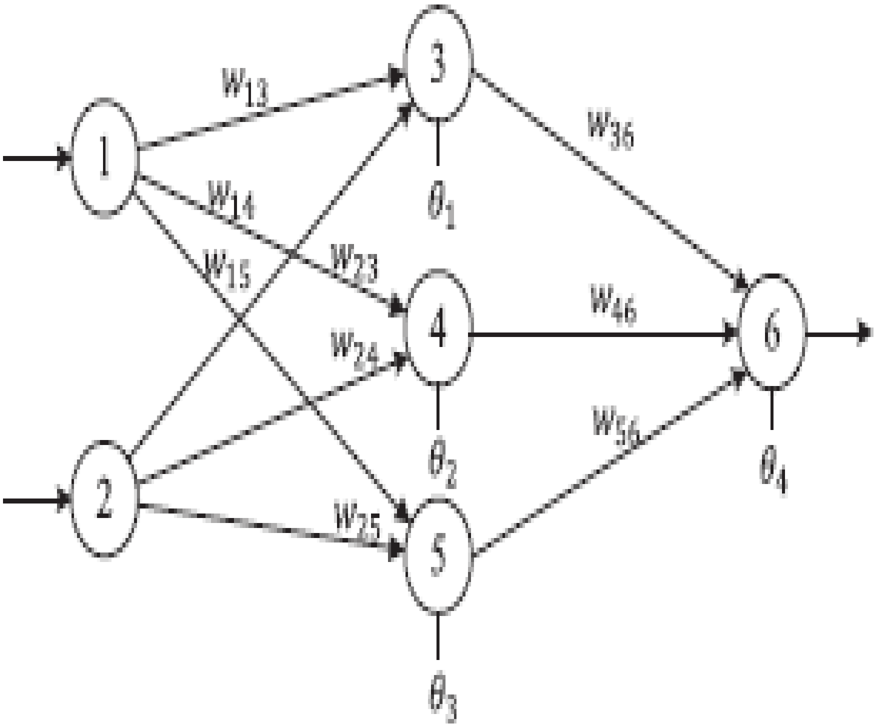

Figure 3. FNN with A 2-3-1 Structure (Source: Mirjalili *Et al*, 2012)

The Pseudo code for the PSO-LMBP algorithm interface is as follows;

- Step 1: The particles positions and velocities are initialized from a uniformly distributed random probability in the problem search space within the range of [0 1]
- Step 2: Evaluate each initialized particle's fitness value (difference in error between the target output and the actual output of the FFN). If the fitness value is better than its local best, the best position p_g is evaluated from the current particles positions.
- Step 3: If the stop criteria or maximum iteration is reached go to Step 8, else go to Step 4.
- Step 4: The particle with the best fitness value p_g is selected. The positions and velocities of all the particles are updated according to Eqs. 8 and 9. The coefficients $c_1$and c_2 are updated by sampling a Gaussian distribution. The boundaries for velocity and position are checked. If the new position or velocity is beyond the boundaries, the new value is set to be the minimum or maximum.
- Step 5: Evaluate each particles fitness value and the worst particle is replaced by the stored best particle. Update the best particle p_g if the new particle's fitness p_{ib} is better than p_g
- Step 6: Reduce the inertia weights w
- Step 7: If the current p_g is unchanged for a specified number of generation, then go to Step 8; else, go to Step 3.
- Step 8: Use the LMBP algorithm to search around p_g for some epochs, if the search result is better than p_g output the current search result. Else output p_g

4.3. Classification

Classification is carried out as follow;

i. The optimal network formed from Step 8 (optimum combination of weights and biases as encoded in the best particle structure) is trained using the training dataset.

ii. Thereafter, use the trained network to classify the test dataset

5. Data

The data was generated from a 2007 NigeriaSat1 image (resolution of 32m) covering part of Obafemi Awolowo University (O.A.U), Ile-Ife. The dataset was prepared using the procedure used in preparing the UCL repository LandSat dataset (Keogh *et al*, 1998). The dataset was generated by taking a small section from the original data. One frame of the imagery dataset consists of three digital images of the same scene in different spectral bands. Two of these are in the visible region (corresponding approximately to green and red regions of the visible spectrum) and one is in the (near) infra-red region. Each pixel is an 8-bit binary word, with 0 corresponding to black and 255 to white. The spatial resolution of a pixel is 32m x 32m. Each line contains the pixel values in the three spectral bands (converted to ASCII) of each of the 9 pixels in the 3x3 neighbourhood and a number indicating the classification label. The baseline classification for comparison purpose was done manually using historical high resolution image and local knowledge of the area. The training dataset contains a total of 841 pixels (using a 3x3 neighbourhood). The test dataset contains a total of 361 pixels (using a 3x3 neighbourhood). In each line of data the three spectral values for the top-left pixel are given first followed by the three spectral values for the top-middle pixel and then those for the top-right pixel, and so on with the pixels read out in sequence left-to-right and top-to-bottom. Table 1 describes the dataset while Figures 4 and 5 show the image of the test dataset and baseline classification respectively. Output of the base line classification is in 3x3 neighbourhood blocks (*i.e.* each block consist of 3x3 pixels).

Table 1. Characteristics of the Dataset

Number of examples	Training dataset	841 pixels (3x3 neighbourhood)
	Test dataset	361 (3x3 neighbourhood).
Number of attributes	27 (3 spectral bands x 9 pixels in neighbourhood)	
Attributes	The attributes are numerical, in the range 0 to 255	
Decision Class	Features	Colour
	1 (water body)	Blue
	2 (woodland)	Light green
	3 (settlement)	Red
	4 (wetland)	Pink
	5 (cultivation)	Yellow

5.1. Pre-processing

Before training, the inputs and targets were scaled within a specified range [-1 to 1]. At the output the values were reverted back to their actual values.

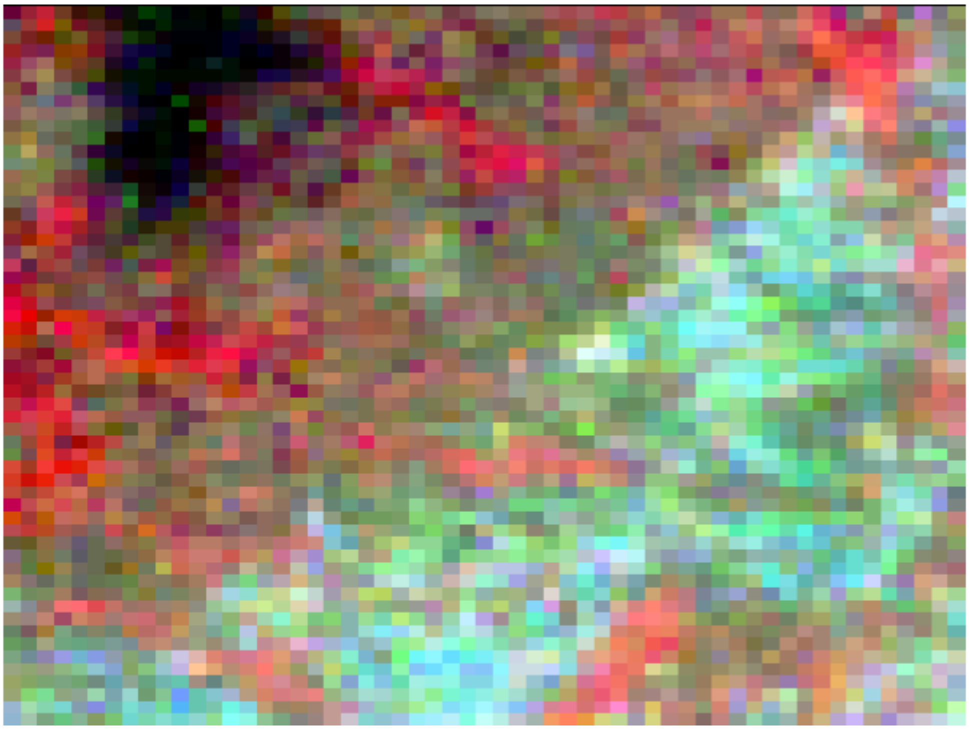

Figure 4. Nigeriasat 1 2007 Image Covering Part of O.A.U Campus

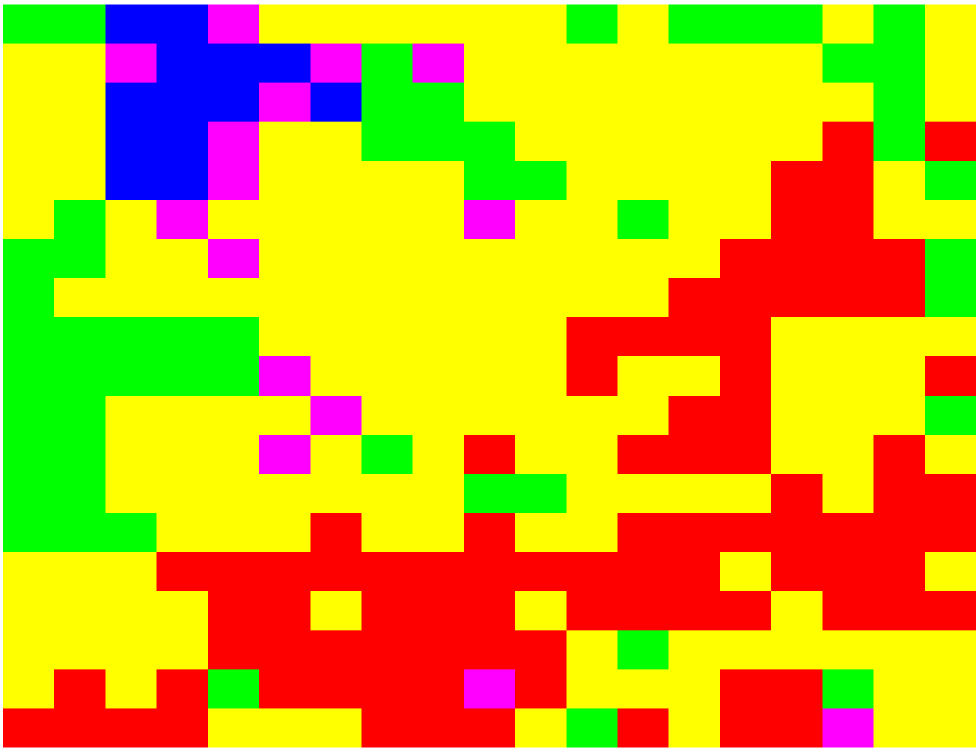

Figure 5. Baseline Classification

6. Result and Discussion

The hybrid PSO-LMBP algorithm was used in training FNNs, for the classification of medium resolution, multispectral satellite images. Table 2 lists the algorithm specific parameters used for the classification.

Table 2. Simulation Parameters

PSO parameters	
Particle Population size	60
Number of epochs	1000
Maximum inertial weight	0.9
Minimum inertial weight	0.2
Inertial weight	0.2
Acceleration constants c_1	2
Acceleration constants c_2	2
Particle velocity	0.8
ANN parameters	
Number of hidden layer	3
Activation function used for hidden layer	Tansig , Tansig , Purelin
Pre-processing for the input layer	Constrain inputs between -1 and 1
Type of network used	FFN
Performance function	MSE (mean square error)

Most of the parameters in Table 2 were determined after multiple simulation runs using different parameters to find the optimal parameters for the classification. Result obtained for PSO-LMBP over three independent runs are presented in Table 3. The average number of misclassification was 13.66, average MSE was 0.0316 and average accuracy of classification was 96.01%. Accuracy of classification was obtained using equation 12

$$Accuracy\ of\ classification = \frac{total\ no\ of\ samples - no\ of\ misclassification}{total\ no\ of\ samples} \times 100\% \quad (12)$$

Figures 6 and 7, shows the performance plot during training and regression plot for the classification with twelve misclassifications. From the performance plot in Figure 6, it can be observed that after the 600 epoch there is no significant increase in the performance of the training process. The regression plot in Figure 7 shows an almost perfect fit for the data and targets, indicating a high level of accuracy in the classification process. Figure 8 is the classified output with twelve misclassifications.

For comparative study, the classification was also done using LMBP and Scaled Conjugate (SC) BP and the results are shown in Table 4. It can be observed from Table 4 that the LM-BPNN outperforms the LMBP and SCBP trained NN’s in terms of number of misclassification, accuracy of classification and MSE.

Table 3. Number of Misclassification, Accuracy of Classification and MSE for Test Data over Three Independent Runs for PSO-LMBP Trained NN

Algorithm	No of misclassification	Accuracy of classification (%)	MSE
PSO Trained LM-BPNN (total no of samples is 361(3x3 neighbourhood pixels)	12	96.66%	0.0241
	13	96.39%	0.0324
	16	95%	0.0384
Average	13.66	96.01%	0.0316

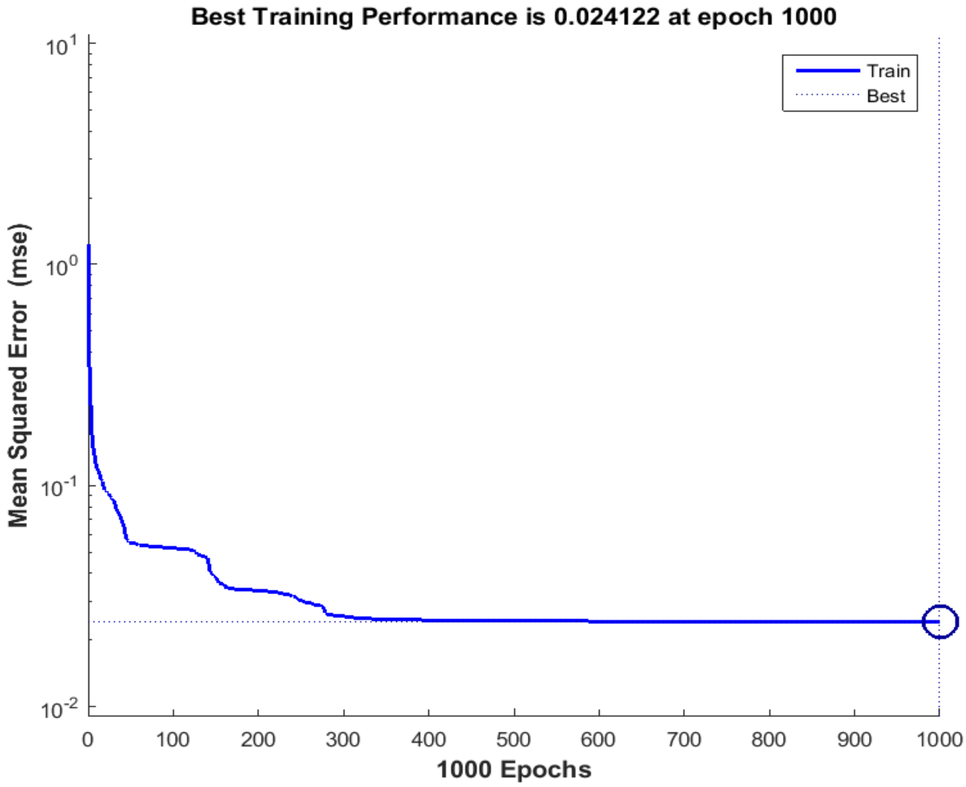

Figure 6. Performance Plot

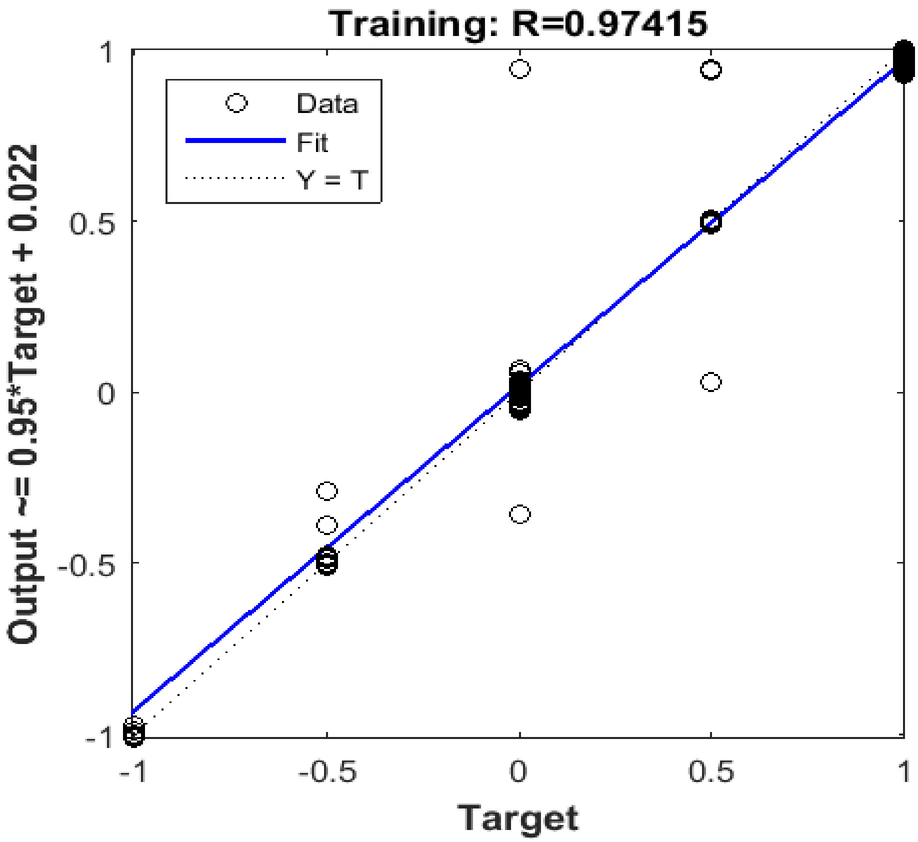

Figure 7. Regression Plot

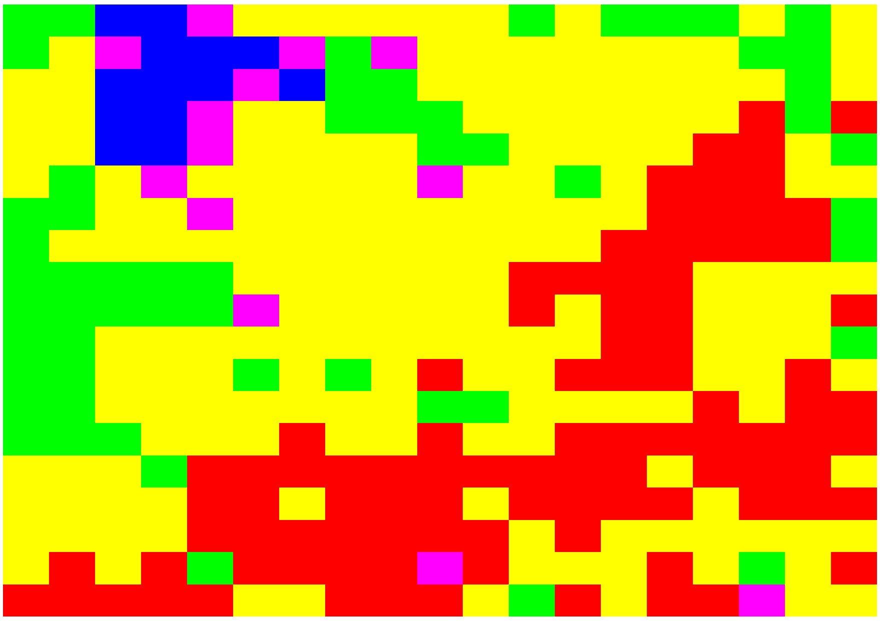

Figure 8. Classified using PSO-LMBB trained NN (12 errors)

Table 4. Number of Misclassification, Accuracy of Classification and MSE for Test Data over 3 Independent runs for PSO-LMBP, LMBP and SCG TRAINED NN'S

Algorithm	No of misclassification	Accuracy of classification (%)	MSE
PSO trained LMBP-NN	12	96.66%	0.0241
	13	96.39%	0.0324
	16	95%	0.0384
	Average = 13.66	Average = 96.01%	Average = 0.0316
LMBP trained NN	23	93.6%	0.0417
	26	92.3%	0.0430
	21	94.2%	0.0412
	Average = 23.3	Average = 93.36%	Average =0.042
Scaled conjugate gradient (SCG) BP trained NN	33	90%	0.051
	29	92%	0.045
	31	91.4%	0.041
	Average = 31	Average = 91.13%	Average = 0.0456

7. Conclusion

In this paper, hybrid PSO with LMBP was applied in training a FNN for the classification of medium resolution multispectral satellite imageries. PSO was used to select the best global particles while LMBP algorithm was used thereafter to search around the best global particles to obtain an optimal network that will be used in training the dataset. The hybrid PSO-LMBP algorithm was evaluated by classifying a medium resolution multispectral satellite image using PSO-LMBP, LMBP and SCGBP and comparing their performances. The hybrid PSO-LMBP algorithm outperforms the conventional LMBP and SCGBP trained ANN classifier and achieves ≈95% accuracy on the test dataset. From the results obtained it can be concluded that the hybrid PSO-LMBP is suitable for classifying medium resolution multispectral satellite imageries. In future research works, we shall focus on how to apply this hybrid PSO-LMBP algorithm for classifying high resolution multispectral satellite imageries.

Acknowledgment

The authors would like to acknowledge the contributions of Mr Jasini for providing the NigeriaSat 1 image used in this work.

References

[1] R. K. Agrawal and N. G. Bawane, "Multiobjective PSO based adaption of neural network topology for pixel classification in satellite imagery", Applied Soft Computing, vol. 28, **(2015)**, pp. 217-225.

[2] B. Ahmed and M. A. Al Noman, "Land cover classification for satellite images based on normalization technique and Artificial Neural Network", Proceedings of the International Conference on Computer and Information Engineering (ICCIE), Rajshahi, Bangladesh, **(2015)**, pp. 138-141.

[3] J. Y. Anchang, E. O. Ananga and R. Pu, "An efficient unsupervised index based approach for mapping urban vegetation from IKONOS imagery", International Journal of Applied Earth Observation and Geoinformation, vol. 50, **(2016)**, pp. 211-220.

[4] J. Cao, H. Cui, H. Shi and L. Jiao, “Big Data: A Parallel Particle Swarm Optimization-Back-Propagation Neural Network Algorithm Based on MapReduce”, PloS One, vol.11, no. 6, **(2016)**, pp. 1-4.

[5] J. F. Chen, Q. H. Do and H. N. Hsieh, “Training artificial neural networks by a hybrid PSO-CS algorithm”, Algorithms, vol. 8, no. 2, **(2015)**, pp. 292-308.

[6] G. Das, P. K. Pattnaik and S. K. Padhy, “ Artificial Neural Network trained by Particle Swarm Optimization for non-linear channel equalization”, Expert Systems with Applications, vol. 41, no. 7, **(2014)**, pp. 3491-3496.

[7] B. A. Garro and R. A. Vázquez, “Designing artificial neural networks using particle swarm optimization algorithms”, Computational Intelligence and Neuroscience, vol. 2015, **(2015)**, pp. 1-5.

[8] C. Gómez, J. C. White, and M. A. Wulder, “Optical remotely sensed time series data for land cover classification: A review”, ISPRS Journal of Photogrammetry and Remote Sensing, vol. 116, **(2016)**, pp. 55-72.

[9] C. P. Jacovides, F. S. Tymvios, J. Boland and M. Tsitouri, “Artificial Neural Network models for estimating daily solar global UV, PAR and broadband radiant fluxes in an eastern Mediterranean site”, Atmospheric Research, vol. 152,**(2015)**, pp.138-145.

[10] E. Keogh, C. Blake and C. Merz, “UCI repository of machine learning databases”, **(1998)**.

[11] N. Martinel, C. Micheloni and G. L. Foresti, “Evolution of neural learning systems: A novel architecture combining the strengths of NTs, CNNs and ELMs”, System, Man and Cybernetics Magazine, vol.1, no. 3, **(2015)**, pp. 1-6.

[12] H. Melo and J. Watada, “ Gaussian-PSO with fuzzy reasoning based on structural learning for training a Neural Network”, Neurocomputing, vol. 172, **(2016)**, pp. 405-412.

[13] S. K. Meher, “Knowledge-encoded granular neural networks for hyperspectral remote sensing image classification”, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 8, no. 6, **(2015)**, pp. 2439-2446.

[14] N. M. Nawi, M. Z. Rehman, M. A. Aziz, T. Herawan and J. H. Abawajy, “An Accelerated Particle Swarm Optimization Based Levenberg Marquardt Back Propagation Algorithm”, In Neural Information Processing: C. Loo, K. Yap, K. Wong, A. Teoh, K. Huang, Eds., Springer International Publishing. vol. 8835, **(2014)**, pp. 245-253.

[15] D. Rumelhart and J. McClelland, “Parallel distributed processing”, MIT Press, Cambridge, MA, **(1986)**.

[16] A. F. Sallama, “Multi particle swarm optimisation algorithm applied to supervisory power control systems”, Doctoral dissertation, College of Engineering, Design and Physical Sciences, Brunel University, London, **(2014)**.

[17] B. C. Xu, and Y. Y. Zhang, “An improved gravitational search algorithm for dynamic neural network identification”, International Journal of Automation and Computing, vol.11, no. 4, **(2014)**, pp. 434-440.

[18] G. P. Zhang, “A neural network ensemble method with jittered training data for time series forecasting. Information Sciences”, vol. 177, no. 23, **(2007)**, pp. 5329-5346.