International Journal of Engineering & Technology, *7 (2) (2018) 817-822*

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i2.11065
Research paper

Optimized artificial neural network for classification of biological data

N Senthilselvan[1]*, S Rajarajan[1], V Subramaniyaswamy[1]

[1]School of Computing, SASTRA Deemed University, India
**Corresponding author E-mail:senthilselvan@cse.sastra.edu*

Abstract

Biological data suffers from the problem of high dimensionality which makes the process of multi-class classification difficult and also these data have elements that are incomplete and redundant. Breast Cancer is currently one of the most pre-dominant causes of death in women around the globe. The current methods for classifying a tumour as malignant or benign involve physical procedures. This often leads to mental stress. Research has now sought to implement soft computing techniques in order to classify tumours based on the data available. In this paper, a novel classifier model is implemented using Artificial Neural Networks. Optimization is done in this neural network by using a meta-heuristic algorithm called the Whale Swarm Algorithm in order to make the classifier model accurate. Experimental results show that new technique outperforms other existing models.

Keywords:*Breast Cancer Classification; Artificial Neural Network; Whale Swarm Optimization; Classifier.*

1. Introduction

Cancer is the term given to the family of disease that are characterised by abnormal cell growth [1-5]. It has been estimated and reported in the cancer facts and figures for 2018 by the American cancer society that 1,735,350 new cases of cancer will be reported and 609,640 deaths because of cancer will be reported in the United States. By the year 2030, the number of people suffering from cancer will be nearly 22 million [6-10]. Lung, Prostate and Breast related cancers are one of the most common cancers. Amongst these, breast cancer is one of the leading causes of death among middle-aged women [11-15]. The American Cancer Society has reported that nearly 252,000 new cases of cancer have been reported in the year 2017. This breast cancer can be detected by regular tests or testing after a lump has been developed. Not all lumps or tumours are cancers. Tumours are of two types, Benign and Malignant [16-20]. Malignant tumours are the ones that are categorised as cancers. The current process of finding out whether a tumour is benign or malignant involves Mammogram Imaging, screening exams or biopsy of the cancer tissue [21-23]. All of these tests are invasive and require strain the patient physically and mentally. They may also lead to unwanted diagnosis. To overcome all of these, an expert system is required it is capable of classifying the tumour from records of relevant data [24-28]. This is the main reason for applying machine learning and soft computing techniques in the research of medical data [29-34].

Many soft computing techniques have been implemented on the research of classifying breast cancer data. One of the most recent research works has been in the field of neural networks. Artificial Neural Network (ANN) is a computer system that mimics the working of the nervous systems and the way the biological brain processes data [35-36]. These networks are arranged as layers consisting of artificial neurons which are interconnected with each other. Neural networks are used because of their ability of adaptive learning, where they learn to detect patterns from a set of training data. These networks also self-organise the data according to themselves. The network thus built will then be applied on a set of testing data to carry of binary classification [37-40]. Hence build an ANN for the classification breast cancer data into two classes: Malignant and Benign. But one disadvantage of the network is that it tends to get stuck at the local minima and not reach the global minima. It also has low convergence. To overcome these disadvantages, a meta-heuristic algorithm is used to optimise certain key parameters of the network. In this paper, Whale Swarm Algorithm was incorporatedit’s a novel meta-heuristic algorithm proven to have higher convergence and avoidance of the local optimum. Build an artificial neural network and then optimize it with whale swarm algorithm to improve the accuracy of the classifier.

2. Related works

Machine learning has been one of the biggest fields of research from the turn of the century. Many notable methods have been developed to build classifier models and classify datasets to obtain notable insights. Over the last two decades many researchers have built expert systems using various models. In the year 1996, Sentino[14] developed neural networks for classifying breast cancer data. Since then many research works have been carried out in the field of neural networks. Nauck et al. built a neuro-fuzzy model in the year 1999. In 2003, Abonyi developed a classifier system using the fuzzy clustering method employing supervised learning. He was able to achieve 97% accuracy with the classifier. Ubeyli[14] carried forward the research on neural nets when he submitted four different variations of neural networks in the year 2007.

Polat and Gunes built a novel classifier using Least Square Support Vector Machine and achieved a classifier accuracy of nearly 98%. In the very same year, Akay also proposed a classifier model that was based on Support Vector Machine but combined it with feature selection technique to increase the performance. Polat and

Gunes also built a fuzzy artificial immune system that employed the nearest neighbour classifier model in 2007. In 2011, Paulin created an expert system using a feed forward mechanism based artificial neural network. In the very same year, Chen et al. proposed a method using support vector machines where they employed the rough set feature selection technique.

In the year 2014, Dheeba et al. suggested a image classifier that worked on wavelet neural network that was optimised using the particle swarm optimization technique. This research was carried out on Mammogram Image Database. In 2014, Senapathi proposed two variants of a classifier that was based on Radial Basis Functional Neural Networks. One variant used the K- Particle swarm optimization technique and the other variant used the Recursive Least Square filter for optimization. In the year 2015, Mert et al. proposed a classifier model with Radial Basis Functional Neural Networks but now used independent component analysis along with it.

Nahato proposed a classifier model with Back Propagation neural network in 2015 and the very next year Zaher optimised it using RIW. Zaher also proposed a Deep Belief Network as a classifier. Table 1 shows authors and their methodologies they used for developing classifiers. It also highlights the dataset that was used and the year the research was conducted.

Table 1: Existing Classifier Models

Work	Technique	Dataset	Evaluation Method	Year
Nauck and Kruse[1]	Neuro-Fuzzy	WBCD	Accuracy	1999
Abonyi and Szeifert[2]	Supervised Fuzzy Clustering	WBCD, Wine Data	Accuracy	2003
Polat and Gunes[3]	Least Square – Support Vector Machine, Fuzzy-Airtificial Immune System	WBCD	Accuracy, sensitivity, Specificity	2007
Akay[4]	Support Vector Machine cominbed with feature selection	WBCD	Accuracy, Specificity, Sensitivity	2009
Paulin[5]	Feed Forward Artificial Neural Network	WBCD	Accuracy	2011
Hui-Ling Chen et al.[6]	Rough Set based Support Vector Machine	WBCD	Accuracy	2011
Dheeba et al.[7]	Particle Swarm Optimized Wavelet Neural Network	Mammogram Database	ROC curve, specificity, sensitivity	2014
M.R.Senapathi et al.[8]	RBFNN-KPSO, RBFNN using extended kalman filter.	WBCD	Accuracy	2014
Mert et al.[9]	RBFNN using Independent component analysis	WBCD	Accuracy	2015
Nahato et al.[10]	RS-BPNN	Hepatitis, WBC, Statlog Heart Disease Datasets	Accuracy	2015
Zaher et al.[11]	RIW-BPNN, DBN-NN	WBCD	Accuracy, Specificity, Sensitivity	2016
HarikumarRajaguru[12]	Bayesian Linear Discriminant Analysis	82 Breast Cancer Patient Details	Accuracy, Specificity, Sensitivity	2017
Tan et al.[13]	MCCNN	Mini-MIAS database	Accuracy, Specificity, Sensitivity	2017

3. Proposed methodology

3.1. Building an ANN

The neural network is built as a series of layers where some are hidden and some are visible. The input layer and output layer is visible whereas the rest are hidden layers. The data is compiled and changed into a '.csv' file, in order to load it into the system. Once the data is ready, we build the model of our network. This neural network follows a sequential model where we define the input layer followed by four layers that are hidden and ending with the output layer. The number of hidden layers is not defined as a fixed constant; it can vary according to the problem in hand. We will build a fully interconnected network of neurons. We define the neurons in each of the layers individually with respect to the weight and activation functions. The output layer will have only a single neuron in order to enable binary classification. All the neurons have uniform weight distribution. These layers have individual activation functions which determine the behaviour of neuron in a given layer. The input layer will have 9 neurons because the dataset has 9 features. This is hence the input layer. The four hidden layers will have 12, 9, 6 and 3 neurons respectively. We also define the output layer. The model is shown in Figure 1. The model is then compiled. During the compilation the loss function is defined. The goal is to ultimately minimise the value returned by the loss function. An optimizer is defined which handles the working of the neural network taking care of the gradient descent and the rate of convergence. We provide the metric that will be used to access this classifier model. Once the model in complied, we fit the data[16] in the neural network. We divide the data into training and testing data. The network uses both of these datasets to learn and adapt. The metric is evaluated for both the datasets. This evaluation phase returns the accuracy of the classifier and the predictions of the network on the testing data.

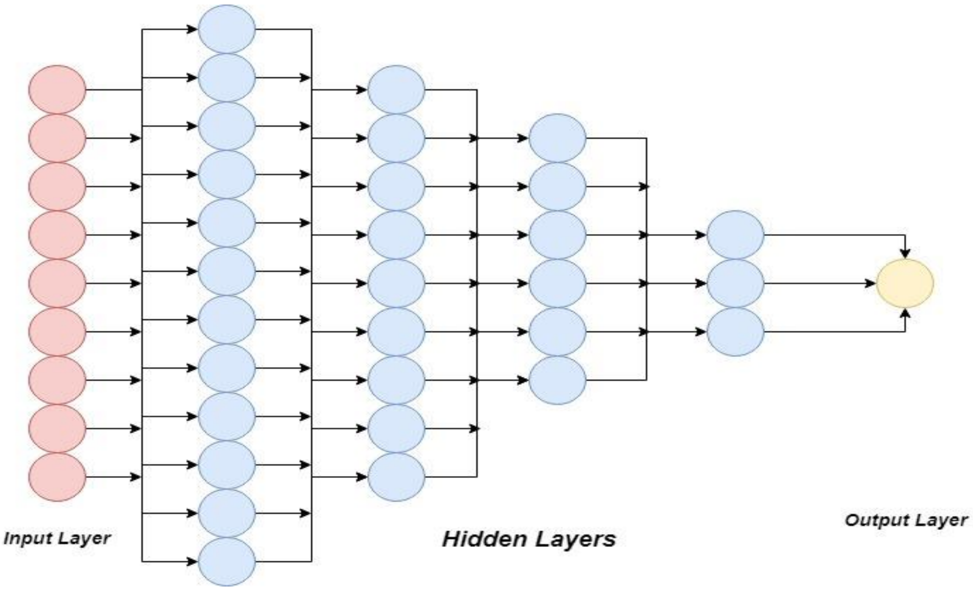


Fig. 1: Deep Artificial Neural Network.

3.2. Optimization using whale swarm method

Whale swarm algorithm is a meta-heuristic algorithm proposed by Mirajili and Lewis and mimics the hunting pattern of humpback whales. Humpback whales hunt their prey by either swimming around them in the shape of a shrinking circle or by selecting a random prey. The algorithm is thus implemented in two phases namely the exploitation phase and the exploration phase which mimics each of the hunting patterns.

The exploitation phase mimics the pattern of encircling and the spiral bubble net attacking method and is modeled mathematically as

$$\mathrm{Qw}(\mathrm{t}+1)=\mathrm{X}.\,\mathrm{e}^{\mathrm{bn}}.\cos(2\pi\mathrm{n})+\mathrm{Hp},$$

where $\mathrm{X}=|\mathrm{Hp}(\mathrm{t})-\mathrm{Hw}(\mathrm{t})|$ is the distance between the whale $\mathrm{Hw}(\mathrm{t})$ and its prey $\mathrm{Hp}(\mathrm{t})$ in the iteration t, b is a constant defining the shape of the spiral and n is a random number in [-1,1].

The Encircling method updates the position as

$$\mathrm{Qw}(\mathrm{t}+1)=\mathrm{Hp}(\mathrm{t})-\mathrm{A}.\mathrm{X}$$

$\mathrm{X}=|\mathrm{C}.\,\mathrm{Hp}(\mathrm{t})-\mathrm{Hw}(\mathrm{t})|$, where D is the distance between the whale and prey.

A and C are computed as

$$\mathrm{C}=2\mathrm{l}$$

$$\mathrm{A}=2\mathrm{a}.\,\mathrm{l}-\mathrm{a},$$

Where one is a random vector and a value is decreased from 2 to 0 in the subsequent iterations. The whole process is shown as

$$\mathrm{Qw}(\mathrm{t}+1)=\begin{cases}\mathrm{Hp}(\mathrm{t})-\mathrm{A}.\mathrm{X}, pr<0.5\\ \mathrm{X}.\,\mathrm{e}^{\mathrm{bn}}.\cos(2\pi\mathrm{n})+\mathrm{Hp}, pr\geq 0.5\end{cases}$$

Wherepr is the probability that it can choose either to encircle the prey or spiral around it. In the exploration phase, a random whale is selected to guide the search. It is mathematically represented as

$$\mathrm{Hw}(\mathrm{t}+1)=\mathrm{Hrand}(\mathrm{t})-\mathrm{A}.\mathrm{X}$$

$$\mathrm{X}=|\mathrm{C}.\,\mathrm{Hrand}(\mathrm{t})-\mathrm{Hw}(\mathrm{t})|$$

Where $\mathrm{Hrand}(\mathrm{t})$ specifies the random whale chosen from the population and A is a random vector with value greater than 1 or less than -1. $\mathrm{Hp}(\mathrm{t})$is continuously updated using the best solution. The algorithm is shown in figure 2.

```
BEGIN
Initialize agents
Find current best
global best = current best
FOR t = 0 : number of iterations
    FOR each agent
        find better and nearest
        IF Exists
            move current agent in direction of its better and nearest
        END IF
    find current best
    IF current best better than global best
        SET global best to current best
    END IF
END FOR
Save golobal best
END
```

Fig. 2: Whale Swarm Algorithm.

4. Experimental setup

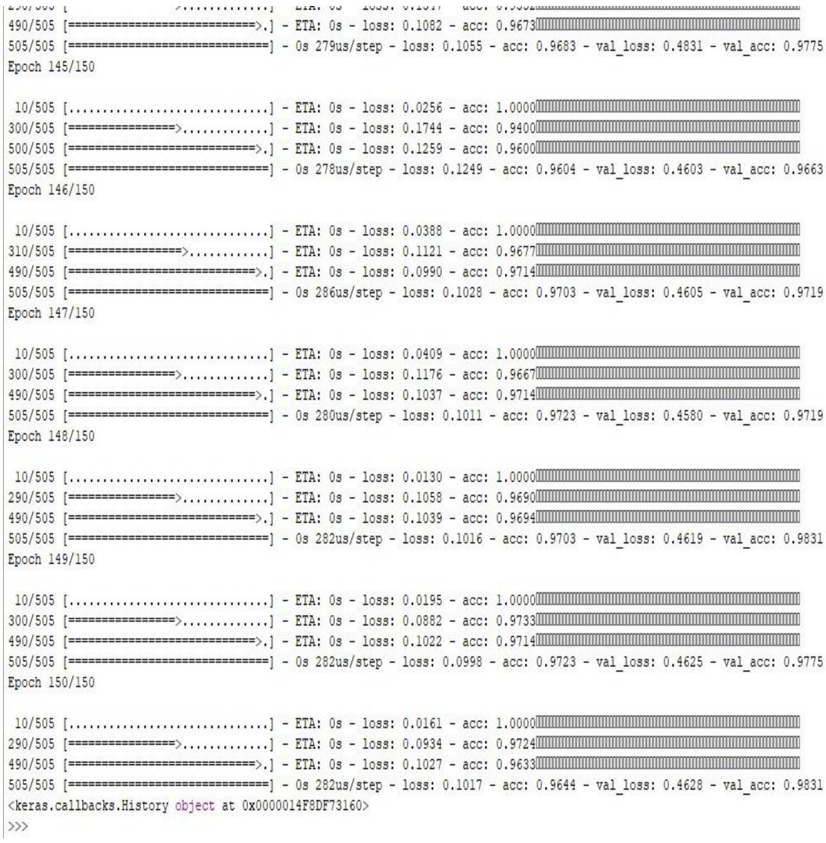
In this paper we will work with Keras, an API that can be used to create and work with neural networks. Keras is based on python programming. It also uses the Tensorflow, another API to carry out the backend processing. The system is first installed with the latest version of the python and all the necessary modules in python namely numpy, scipy, matplotlib and pandas are added to the system. Then the Tensorflow API is installed upon which the Keras system is installed. Once the system is completely functional, the neural network is created and tested. The algorithm is shown in figure 3.

```
BEGIN
Import Sequential model into Keras
Import Numpy into Keras
Import Dense Layers from Keras
Train_Data = Load Training Data from CSV file.
Test_Data = Load Testing Data from CSV file.
X= Partition Train_Data and set to first 9 columns.
Y= Partition Train_Data and set to last column.
TX= Partition Test_Data and set to first 9 colums.
TY= Partition Test_Data and set to last column.
Create first Layer with 9 neurons
Create second Layer with 12 neurons
Create third layer with 9 neurons
Create fourth layer with 6 neurons
Create fifth Layer with 3 neurons
Create last Layer with 1 neuron.
Compile the model.
Optimize the Value of Epochs using Whale Swarm Optimization.
Fit the Model
Evaluate the Model
END
```

Fig. 3: Optimized Deep Neural Network.

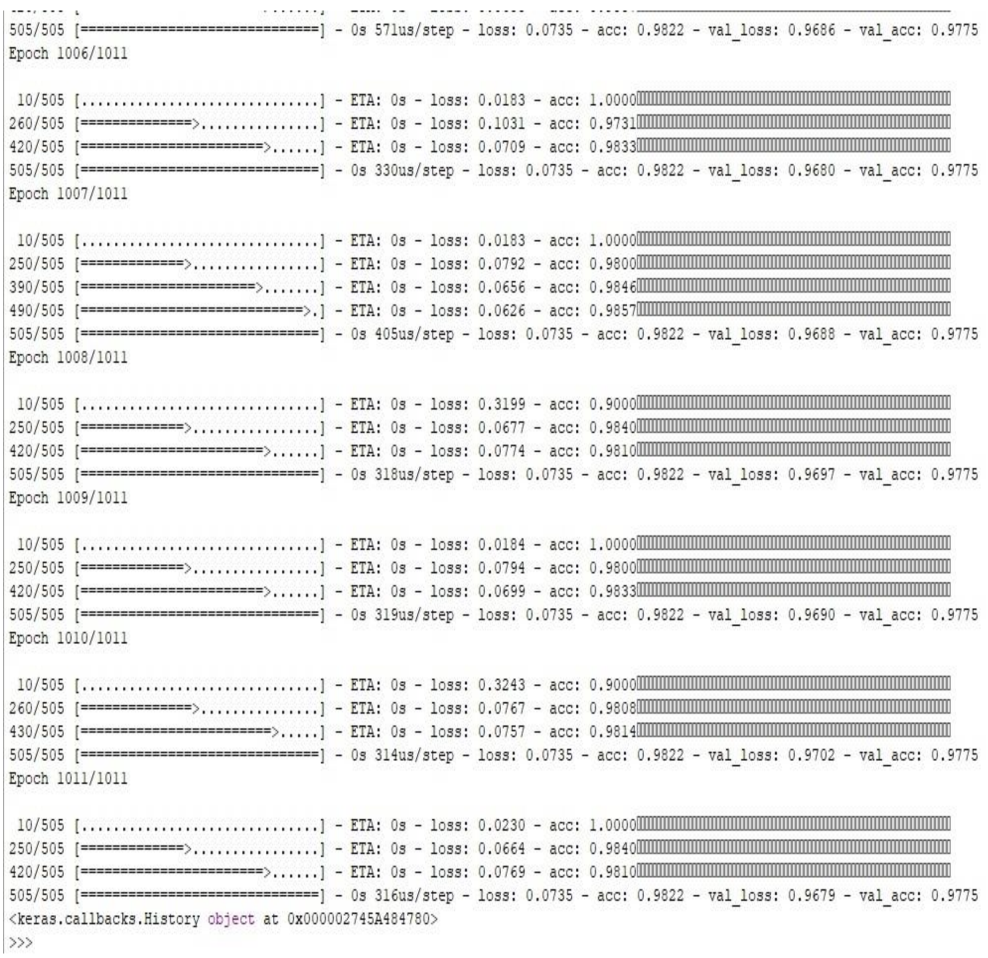
5. Experimental results

The program was successfully executed according to the proposed algorithm. The accuracy of the model before the optimization was applied stood at 96.44%. This is shown in figure 4. Once the optimization has been added, the accuracy becomes 98% which is shown in figure 5. The classifier model which uses artificial neural network and is optimised using whale swarm optimization has an accuracy of 98.22% with a validation accuracy of 97%. The classifier model built using artificial neutral network and optimised using Whale swarm optimisation has shown paramount results. This classifier model has better performance than many other similar classifier models as shown in table 2.



```
290/505 [===============>..............] - ETA: 0s - loss: 0.1317 - acc: 0.9552
490/505 [============================>.] - ETA: 0s - loss: 0.1082 - acc: 0.9673
505/505 [==============================] - 0s 279us/step - loss: 0.1055 - acc: 0.9683 - val_loss: 0.4831 - val_acc: 0.9775
Epoch 145/150

 10/505 [..............................] - ETA: 0s - loss: 0.0256 - acc: 1.0000
300/505 [===============>..............] - ETA: 0s - loss: 0.1744 - acc: 0.9400
500/505 [============================>.] - ETA: 0s - loss: 0.1259 - acc: 0.9600
505/505 [==============================] - 0s 278us/step - loss: 0.1249 - acc: 0.9604 - val_loss: 0.4603 - val_acc: 0.9663
Epoch 146/150

 10/505 [..............................] - ETA: 0s - loss: 0.0388 - acc: 1.0000
310/505 [================>.............] - ETA: 0s - loss: 0.1121 - acc: 0.9677
490/505 [============================>.] - ETA: 0s - loss: 0.0990 - acc: 0.9714
505/505 [==============================] - 0s 286us/step - loss: 0.1028 - acc: 0.9703 - val_loss: 0.4605 - val_acc: 0.9719
Epoch 147/150

 10/505 [..............................] - ETA: 0s - loss: 0.0409 - acc: 1.0000
300/505 [===============>..............] - ETA: 0s - loss: 0.1176 - acc: 0.9667
490/505 [============================>.] - ETA: 0s - loss: 0.1037 - acc: 0.9714
505/505 [==============================] - 0s 280us/step - loss: 0.1011 - acc: 0.9723 - val_loss: 0.4580 - val_acc: 0.9719
Epoch 148/150

 10/505 [..............................] - ETA: 0s - loss: 0.0130 - acc: 1.0000
290/505 [===============>..............] - ETA: 0s - loss: 0.1058 - acc: 0.9690
490/505 [============================>.] - ETA: 0s - loss: 0.1039 - acc: 0.9694
505/505 [==============================] - 0s 282us/step - loss: 0.1016 - acc: 0.9703 - val_loss: 0.4619 - val_acc: 0.9831
Epoch 149/150

 10/505 [..............................] - ETA: 0s - loss: 0.0195 - acc: 1.0000
300/505 [===============>..............] - ETA: 0s - loss: 0.0882 - acc: 0.9733
490/505 [============================>.] - ETA: 0s - loss: 0.1022 - acc: 0.9714
505/505 [==============================] - 0s 282us/step - loss: 0.0998 - acc: 0.9723 - val_loss: 0.4625 - val_acc: 0.9775
Epoch 150/150

 10/505 [..............................] - ETA: 0s - loss: 0.0161 - acc: 1.0000
290/505 [===============>..............] - ETA: 0s - loss: 0.0934 - acc: 0.9724
490/505 [============================>.] - ETA: 0s - loss: 0.1027 - acc: 0.9633
505/505 [==============================] - 0s 282us/step - loss: 0.1017 - acc: 0.9644 - val_loss: 0.4628 - val_acc: 0.9831
<keras.callbacks.History object at 0x0000014F8DF73160>
>>>
```

Fig. 4: Performance without Optimization.



```
505/505 [==============================] - 0s 571us/step - loss: 0.0735 - acc: 0.9822 - val_loss: 0.9686 - val_acc: 0.9775
Epoch 1006/1011

 10/505 [..............................] - ETA: 0s - loss: 0.0183 - acc: 1.0000
260/505 [==============>...............] - ETA: 0s - loss: 0.1031 - acc: 0.9731
420/505 [=======================>......] - ETA: 0s - loss: 0.0709 - acc: 0.9833
505/505 [==============================] - 0s 330us/step - loss: 0.0735 - acc: 0.9822 - val_loss: 0.9680 - val_acc: 0.9775
Epoch 1007/1011

 10/505 [..............................] - ETA: 0s - loss: 0.0183 - acc: 1.0000
250/505 [=============>................] - ETA: 0s - loss: 0.0792 - acc: 0.9800
390/505 [======================>.......] - ETA: 0s - loss: 0.0656 - acc: 0.9846
490/505 [============================>.] - ETA: 0s - loss: 0.0626 - acc: 0.9857
505/505 [==============================] - 0s 405us/step - loss: 0.0735 - acc: 0.9822 - val_loss: 0.9688 - val_acc: 0.9775
Epoch 1008/1011

 10/505 [..............................] - ETA: 0s - loss: 0.3199 - acc: 0.9000
250/505 [=============>................] - ETA: 0s - loss: 0.0677 - acc: 0.9840
420/505 [=======================>......] - ETA: 0s - loss: 0.0774 - acc: 0.9810
505/505 [==============================] - 0s 318us/step - loss: 0.0735 - acc: 0.9822 - val_loss: 0.9697 - val_acc: 0.9775
Epoch 1009/1011

 10/505 [..............................] - ETA: 0s - loss: 0.0184 - acc: 1.0000
250/505 [=============>................] - ETA: 0s - loss: 0.0794 - acc: 0.9800
420/505 [=======================>......] - ETA: 0s - loss: 0.0699 - acc: 0.9833
505/505 [==============================] - 0s 319us/step - loss: 0.0735 - acc: 0.9822 - val_loss: 0.9690 - val_acc: 0.9775
Epoch 1010/1011

 10/505 [..............................] - ETA: 0s - loss: 0.3243 - acc: 0.9000
260/505 [==============>...............] - ETA: 0s - loss: 0.0767 - acc: 0.9808
430/505 [========================>.....] - ETA: 0s - loss: 0.0757 - acc: 0.9814
505/505 [==============================] - 0s 314us/step - loss: 0.0735 - acc: 0.9822 - val_loss: 0.9702 - val_acc: 0.9775
Epoch 1011/1011

 10/505 [..............................] - ETA: 0s - loss: 0.0230 - acc: 1.0000
250/505 [=============>................] - ETA: 0s - loss: 0.0664 - acc: 0.9840
420/505 [=======================>......] - ETA: 0s - loss: 0.0769 - acc: 0.9810
505/505 [==============================] - 0s 316us/step - loss: 0.0735 - acc: 0.9822 - val_loss: 0.9679 - val_acc: 0.9775
<keras.callbacks.History object at 0x000002745A484780>
>>>
```

Fig. 5: Performance after Optimization.

Table 2: Comparison of Classifier Accuracy

Classifier	Accuracy
PSOWNN	93.67%
ICA-RBFNN	90.37%
RBFNN-KPSO	97.85%
RBFNN- Extended Kalman filter	96.42%
BLDA	83.45%
BCDCNN	82.71%
ANN-WSA	98.22%

6. Conclusion

As seen from the table, the final classifier ANN-WSA as prescribed in this paper performs better than already existing classifier models. This work can be extended by coupling the ANN with other meta- heuristic algorithms. In this paper, we have optimized a single parameter: epochs, in order to make the network more accurate. Extension to this can be done by optimizing more than one parameter in the algorithm.

References

[1] Nauck, D., & Kruse, R. (1999). Obtaining interpretable fuzzy classification rules from medical data. *Artificial intelligence in medicine*, *16*(2), 149-169.https://doi.org/10.1016/S0933-3657(98)00070-0.

[2] Abonyi, J., &Szeifert, F. (2003). Supervised fuzzy clustering for the identification of fuzzy classifiers. *Pattern Recognition Letters*, *24*(14), 2195-2207.https://doi.org/10.1016/S0167-8655(03)00047-3.

[3] Şahan, S., Polat, K., Kodaz, H., &Güneş, S. (2007). A new hybrid method based on fuzzy-artificial immune system and k-nn algorithm for breast cancer diagnosis. *Computers in Biology and Medicine*, *37*(3), 415-423.https://doi.org/10.1016/j.compbiomed.2006.05.003.

[4] Akay, M. F. (2009). Support vector machines combined with feature selection for breast cancer diagnosis. *Expert systems with applications*, *36*(2), 3240-3247.https://doi.org/10.1016/j.eswa.2008.01.009.

[5] Paulin, F., &Santhakumaran, A. (2011). Classification of breast cancer by comparing back propagation training algorithms. *International Journal on Computer Science and Engineering*, *3*(1), 327-332.

[6] Chen, H. L., Yang, B., Liu, J., & Liu, D. Y. (2011). A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis. *Expert Systems with Applications*, *38*(7), 9014-9022.https://doi.org/10.1016/j.eswa.2011.01.120.

[7] Dheeba, J., Singh, N. A., &Selvi, S. T. (2014). Computer-aided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach. *Journal of biomedical informatics*, *49*, 45-52.https://doi.org/10.1016/j.jbi.2014.01.010.

[8] Senapati, M. R., Panda, G., & Dash, P. K. (2014). Hybrid approach using KPSO and RLS for RBFNN design for breast cancer detection. *Neural Computing and Applications*, *24*(3-4), 745-753.https://doi.org/10.1007/s00521-012-1286-6.

[9] Mert, A., Kılıç, N., Bilgili, E., & Akan, A. (2015). Breast cancer detection with reduced feature set. *Computational and mathematical methods in medicine*, *2015*.https://doi.org/10.1155/2015/265138.

[10] Nahato, K. B., Harichandran, K. N., &Arputharaj, K. (2015). Knowledge mining from clinical datasets using rough sets and backpropagation neural network. *Computational and mathematical methods in medicine*, *2015*.https://doi.org/10.1155/2015/460189.

[11] Abdel-Zaher, A. M., &Eldeib, A. M. (2016). Breast cancer classification using deep belief networks. *Expert Systems with Applications*, *46*, 139-144.https://doi.org/10.1016/j.eswa.2015.10.015.

[12] Rajaguru, H., &Prabhakar, S. K. (2017, October). Bayesian linear discriminant analysis for breast cancer classification. In *Communication and Electronics Systems (ICCES), 2017 2nd International Conference on* (pp. 266-269). IEEE.https://doi.org/10.1109/CESYS.2017.8321279.

[13] Tan, Y. J., Sim, K. S., & Ting, F. F. (2017, November). Breast cancer detection using convolutional neural networks for mammogram imaging system.In *Robotics, Automation and Sciences (ICORAS), 2017 International Conference on* (pp. 1-5).IEEE.https://doi.org/10.1109/ICORAS.2017.8308076.

[14] Arya, C., &Tiwari, R. (2016, January). Expert system for breast cancer diagnosis: a survey. In *Computer Communication and Informatics (ICCCI), 2016 International Conference on* (pp. 1-9). IEEE.https://doi.org/10.1109/ICCCI.2016.7479940.

[15] UCI Machine Learning Repository. [http://archive.ics.uci.edu/ml/]. Irvine, CA: University of California, Center for Machine Learning and Intelligent Systems.

[16] Logesh, R., Subramaniyaswamy, V., Vijayakumar, V., Gao, X. Z., &Indragandhi, V. (2017). A hybrid quantum-induced swarm intelligence clustering for the urban trip recommendation in smart city. Future Generation Computer Systems, 83, 653-673. https://doi.org/10.1016/j.future.2017.08.060.

[17] Subramaniyaswamy, V., &Logesh, R. (2017). Adaptive KNN based Recommender System through Mining of User Preferences. Wireless Personal Communications, 97(2), 2229-2247.https://doi.org/10.1007/s11277-017-4605-5.

[18] Logesh, R., &Subramaniyaswamy, V. (2017). A Reliable Point of Interest Recommendation based on Trust Relevancy between Users. Wireless Personal Communications, 97(2), 2751-2780. https://doi.org/10.1007/s11277-017-4633-1.

[19] Logesh, R., &Subramaniyaswamy, V. (2017). Learning Recency and Inferring Associations in Location Based Social Network for Emotion Induced Point-of-Interest Recommendation. Journal of Information Science & Engineering, 33(6), 1629–1647.

[20] Subramaniyaswamy, V., Logesh, R., Abejith, M., Umasankar, S., &Umamakeswari, A. (2017). Sentiment Analysis of Tweets for Estimating Criticality and Security of Events. Journal of Organizational and End User Computing (JOEUC), 29(4), 51-71. https://doi.org/10.4018/JOEUC.2017100103.

[21] Indragandhi, V., Logesh, R., Subramaniyaswamy, V., Vijayakumar, V., Siarry, P., &Uden, L. (2018). Multi-objective optimization and energy management in renewable based AC/DC microgrid. Computers & Electrical Engineering.

[22] Subramaniyaswamy, V., Manogaran, G., Logesh, R., Vijayakumar, V., Chilamkurti, N., Malathi, D., &Senthilselvan, N. (2018). An ontology-driven personalized food recommendation in IoT-based healthcare system. The Journal of Supercomputing, 1-33. https://doi.org/10.1007/s11227-018-2331-8.

[23] Arunkumar, S., Subramaniyaswamy, V., &Logesh, R. (2018). Hybrid Transform based Adaptive Steganography Scheme using Support Vector Machine for Cloud Storage. Cluster Computing.

[24] Indragandhi, V., Subramaniyaswamy, V., &Logesh, R. (2017). Resources, configurations, and soft computing techniques for power management and control of PV/wind hybrid system. Renewable and Sustainable Energy Reviews, 69, 129-143.https://doi.org/10.1016/j.rser.2016.11.209.

[25] Ravi, L., &Vairavasundaram, S. (2016). A collaborative location based travel recommendation system through enhanced rating prediction for the group of users. Computational intelligence and neuroscience, 2016, Article ID: 1291358. https://doi.org/10.1155/2016/1291358.

[26] Logesh, R., Subramaniyaswamy, V., Malathi, D., Senthilselvan, N., Sasikumar, A., &Saravanan, P. (2017). Dynamic particle swarm optimization for personalized recommender system based on electroencephalography feedback. Biomedical Research, 28(13), 5646-5650.

[27] Arunkumar, S., Subramaniyaswamy, V., Karthikeyan, B., Saravanan, P., &Logesh, R. (2018). Meta-data based secret image sharing application for different sized biomedical images. Biomedical Research, 29.

[28] Vairavasundaram, S., Varadharajan, V., Vairavasundaram, I., & Ravi, L. (2015). Data mining-based tag recommendation system: an overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5(3), 87-112. https://doi.org/10.1002/widm.1149.

[29] Logesh, R., Subramaniyaswamy, V., &Vijayakumar, V. (2018). A personalised travel recommender system utilising social network profile and accurate GPS data. Electronic Government, an International Journal, 14(1), 90-113. https://doi.org/10.1504/EG.2018.089538.

[30] Vijayakumar, V., Subramaniyaswamy, V., Logesh, R., &Sivapathi, A. (2018). Effective Knowledge Based Recommeder System for Tailored Multiple Point of Interest Recommendation. International Journal of Web Portals.

[31] Subramaniyaswamy, V., Logesh, R., &Indragandhi, V. (2018). Intelligent sports commentary recommendation system for individual cricket players. International Journal of Advanced Intelligence Paradigms, 10(1-2), 103-117. https://doi.org/10.1504/IJAIP.2018.089492.

[32] Indragandhi, V., Subramaniyaswamy, V., &Logesh, R. (2017). Topological review and analysis of DC-DC boost converters. Journal of Engineering Science and Technology, 12 (6), 1541–1567.

[33] Saravanan, P., Arunkumar, S., Subramaniyaswamy, V., &Logesh, R. (2017). Enhanced web caching using bloom filter for local area networks. International Journal of Mechanical Engineering and Technology, 8(8), 211-217.

[34] Arunkumar, S., Subramaniyaswamy, V., Devika, R., &Logesh, R. (2017). Generating visually meaningful encrypted image using image splitting technique. International Journal of Mechanical Engineering and Technology, 8(8), 361–368.

[35] Subramaniyaswamy, V., Logesh, R., Chandrashekhar, M., Challa, A., &Vijayakumar, V. (2017). A personalised movie recommendation system based on collaborative filtering. International Journal of High Performance Computing and Networking, 10(1-2), 54-63. https://doi.org/10.1504/IJHPCN.2017.083199.

[36] Senthilselvan, N., UdayaSree, N., Medini, T., SubhakariMounika, G., Subramaniyaswamy, V., Sivaramakrishnan, N., &Logesh, R. (2017). Keyword-aware recommender system based on user demographic attributes. International Journal of Mechanical Engineering and Technology, 8(8), 1466-1476.

[37] Subramaniyaswamy, V., Logesh, R., Vijayakumar, V., &Indragandhi, V. (2015). Automated Message Filtering System in Online Social Network. Procedia Computer Science, 50, 466-475.https://doi.org/10.1016/j.procs.2015.04.016.

[38] Subramaniyaswamy, V., Vijayakumar, V., Logesh, R., &Indragandhi, V. (2015). Unstructured data analysis on big data using map reduce. Procedia Computer Science, 50, 456-465. https://doi.org/10.1016/j.procs.2015.04.015.

[39] Subramaniyaswamy, V., Vijayakumar, V., Logesh, R., &Indragandhi, V. (2015). Intelligent travel recommendation system by mining attributes from community contributed photos. Procedia Computer Science, 50, 447-455.https://doi.org/10.1016/j.procs.2015.04.014.

[40] Vairavasundaram, S., &Logesh, R. (2017). Applying Semantic Relations for Automatic Topic Ontology Construction. Developments and Trends in Intelligent Technologies and Smart Systems, 48.