International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **5(A)**, May **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 3163–3172

# A TWO-STEP SUPERVISED LEARNING ARTIFICIAL NEURAL NETWORK FOR IMBALANCED DATASET PROBLEMS

Asrul Adam[1], Zuwairie Ibrahim[2], Mohd Ibrahim Shapiai[1], Lim Chun Chew[3]
Lee Wen Jau[3], Marzuki Khalid[1] and Junzo Watada[4]

[1]Faculty of Electrical Engineering
Universiti Teknologi Malaysia
81310 UTM Johor Bahru, Malaysia
asruldm@fkegraduate.utm.my; ibrahim@fke.utm.my; marzuki@utm.my

[2]Faculty of Electrical and Electronic Engineering
Universiti Malaysia Pahang
26600 Pekan, Pahang, Malaysia
zuwairie@fke.utm.my

[3]ATTD-a APAC Path Finding
INTEL Malaysia
{ chun.chew.lim; wen.jau.lee }@intel.com

[4]Graduate School of Information and Systems
Waseda University
2-7 Hibikino, Wakamatsu, Kita-Kyushu 808-0135, Japan
junzo.watada@gmail.com

Received September 2010; revised September 2011

Abstract. *In this paper, a two-step supervised learning algorithm of a single layer feedforward Artificial Neural Network (ANN) is proposed for solving imbalanced dataset problems. Levenberg Marquart backpropagation learning algorithm is utilized in the first step learning, while the second step learning mechanism is introduced by optimizing the decision threshold of the step function at the output layer of ANN using particle swarm optimization (PSO). After all the steps learning are accomplished, the best weights and decision threshold value are obtained to be used for testing process. Several imbalanced datasets, which are available in UCI Machine Learning Repository, are chosen as case study. The prediction performance is assessed by Geometric Mean (G-mean), which is a standard measure to indicate the efficiency of classifier for imbalanced datasets. Based on the experimental results, the proposed method is able to provide good G-mean value compared with the conventional ANN approaches.*

**Keywords:** Artificial neural network, Imbalanced dataset problem, Particle swarm optimization, Machine learning, Single layer feedforward neural network, Decision threshold, Two-class classification

1. **Introduction.** An imbalanced dataset can be defined as a dataset that consists of several inputs and outputs (classes), where one of the classes (minority class) is significantly less than other classes (majority class). The problem of imbalanced dataset is due to the imbalanced class ratio. The problem is more difficult to be solved if the class ratio is highly imbalanced and lack of representative data. In recent years, learning from imbalanced datasets has become a crucial problem in machine learning and usually found in many applications such as computer security [1], biomedical [2,3], remote-sensing [4], engineering [5,6], and manufacturing industries [7].

Most of conventional ANN classifiers perform poorly and are not able to efficiently learn from imbalanced datasets because the classifier is designed for balance datasets. A study

carried out by Murphey et al. [8] highlighted the imbalanced datasets problems, which are the overwhelming training instances of the majority class, and the network tends to ignore the minority class and then treats it as noise.

Therefore, the existing learning algorithms for imbalanced dataset problems have been proposed to improve the conventional ANN classifiers. Giang et al. [5] have modified four main training algorithms for feedforward ANN, namely, gradient descent (GD), gradient descent with momentum and variable learning rate (GDMV), resilient back propagation (RPROP), and Levenberg-Marquardt (LM). This finding showed that the modified training algorithms are able to achieve better classification accuracy than conventional ANN training algorithms.

By utilizing a single layer feedforward ANN, Anand et al. [9] have proposed a modified version of the conventional backpropagation algorithm. The algorithm focuses on calculating the direction of the weight changes, which decrease the error for each class. Next, a study was conducted by using three different ANN architectures, which are Fuzzy ARTMAP, multi-layered backpropagation, and Radial Based Functions (RBF) [10]. For each of the three network architectures, three training methods were used: simple train-test, duplicate training samples of the minority class, and the Snowball method. In conclusion, the authors suggested that the Fuzzy ARTMAP has the potential to give robust performance for imbalanced dataset problems.

Moreover, Fu et al. [11] have proposed a modified training algorithm for the RBF neural network to improve the prediction performance of the conventional RBF neural network. The modified training algorithm focuses on improving the accuracy of minority class with maintaining the overall classifier performance. Zhou et al. [12] have applied cost-sensitive learning in backpropagation ANNs. This study also determines the effects of data sampling (undersampling and oversampling) and threshold-moving during training. Instead of utilizing a data sampling technique, this approach finds an optimum decision threshold using a move-threshold algorithm at the output layer of the network to obtain the best prediction performance. In the end, the authors concluded that the threshold-moving is the best training algorithm for a cost-sensitive ANN.

Alejo et al. [13] have proposed a method to improve the classifier performance of RBF and Multilayer Perceptron (MLP) for imbalanced datasets. In this method, the RBF and the MLP are applied with a filtering technique in data preprocessing that is based on a Nearest Neighbour rule. Consequently, this method improved the performance of RBF classifier. Unfortunately, the performance is worst when this method employs to MLP classifier. Another existing approach called Modular Neural Network (MNN) [14], which is based on divide-and-conquer concept [14], is adopted as a novel decomposition technique. The network is combined with several integration methods, such as averaging and Genetic Algorithm (GA), for combining the decisions by each network.

Overall, the strategies to handle these problems can be categorized into two different approaches, which are data level and algorithm level [15]. At the data level, features selection and re-sampling techniques, such as resolving over sampling and under sampling, can be used to minimize the imbalanced effect [13]. On the other hand, the algorithm level involves internal modification of learning algorithm.

The investigation approach proposed in this paper focuses on ANN as a tool to classify imbalanced datasets. In order to improve feedforward ANN for imbalanced datasets, this paper proposes a two-step supervised learning, which includes particle swarm optimization (PSO), as a tool to tune the decision threshold value of the step function at the output layer of ANN. Recently, PSO has been utilized in various applications such as DNA sequence design [16], genetic programming [17], and stock portfolio selection [18].

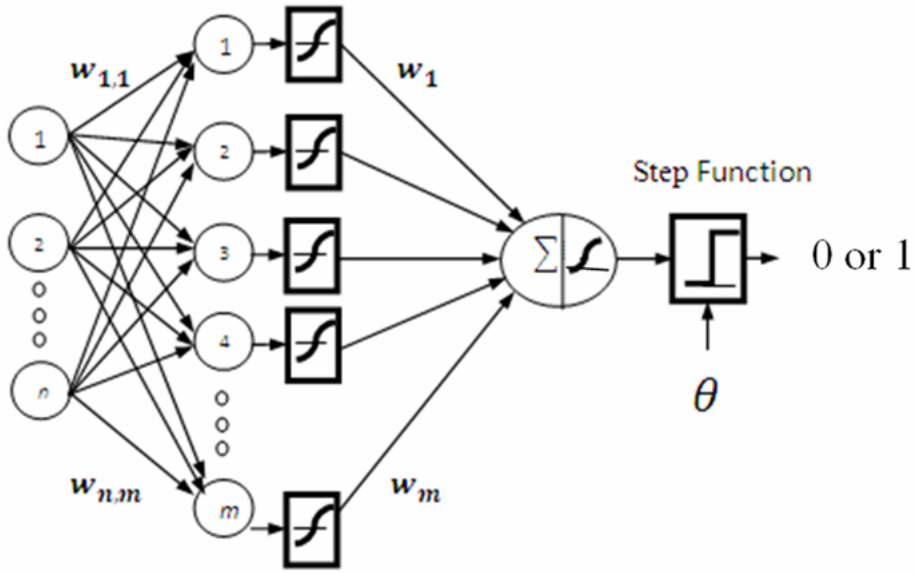


FIGURE 1. An architecture of ANN for binary classification

The rest of this paper is organized as follows. Section 2 focuses on a conventional ANN and performance measure for imbalanced dataset problems. Section 3 describes the proposed two-step learning approach based on ANN for imbalanced dataset problems. In Section 4, several experimental results are presented and discussed. Finally, conclusions are given in Section 5.

## 2. Conventional ANN and Performance Measure for Imbalanced Dataset Problems.

**2.1. ANN classifier for two-class imbalanced dataset problems.** A single layer feedforward ANN for two-class classification is shown in Figure 1, where $n$ is the number of inputs and $m$ is the number of neurons in the hidden layer. The weights, $w$, are located on the links from input layer to hidden layer and from hidden layer to output layer. The hyperbolic tangent function is used in the hidden layer. Levenberg-Marquardt (LM) algorithm is used in training the neural network to minimize the Mean Squared Error (MSE) between the actual outputs of the network and the desired outputs.

A sigmoid function, $f(x)$, shown in Equation (1), is used at the output layer to calculate the limit value of the desired output between 0 and 1.

$$f(x) = (1/(1 + \exp(-x))) \tag{1}$$

$$g(f(x)) = \begin{cases} 1 & \text{if } f(x) > \theta \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $x$ is the total weight values after summation at output layer. Then, a step function, $g(f(x))$, shown in Equation (2), is used to clamp the $f(x)$ value, which is either 0 or 1, based on the threshold value, $\theta$, as the decision threshold. For this network, the decision threshold value is set to 0.5.

**2.2. Performance measure for imbalanced datasets.** Geometric Mean (G-mean) is one of standard performance measures used in an imbalanced dataset classifier. The reason of using G-mean is to balance the ratio of prediction between majority and minority class. The percentage of G-mean indicates that how good an imbalanced dataset classifier

predicts the classes. The G-mean is calculated as follows:

$$G - mean = \sqrt{(TNR \times TPR)} \tag{3}$$

where,

$$TNR = \frac{TN}{TN + FP} \tag{4}$$

$$TPR = \frac{TP}{TP + FN} \tag{5}$$

where $TP$, $FN$, $FP$ and $TN$ can be defined as follows. True Positive ($TP$) refers to correctly prediction of the majority class. False Negative ($FN$) refers to wrongly prediction of the minority class as majority class. False Positive ($FP$) refers to wrongly prediction of majority class as minority class. True Negative ($TN$) refers to correctly prediction of minority class.

3. **The Proposed Two-Step Supervised Learning of Artificial Neural Network.** Figure 2 shows an overview of the proposed approach. The proposed approach can be divided into three phases; first-step learning, second-step learning, and testing. At first, the dataset is randomly divided for training and testing processes. The class ratio of the training and testing dataset must be exactly similar to the class ratio in the dataset.

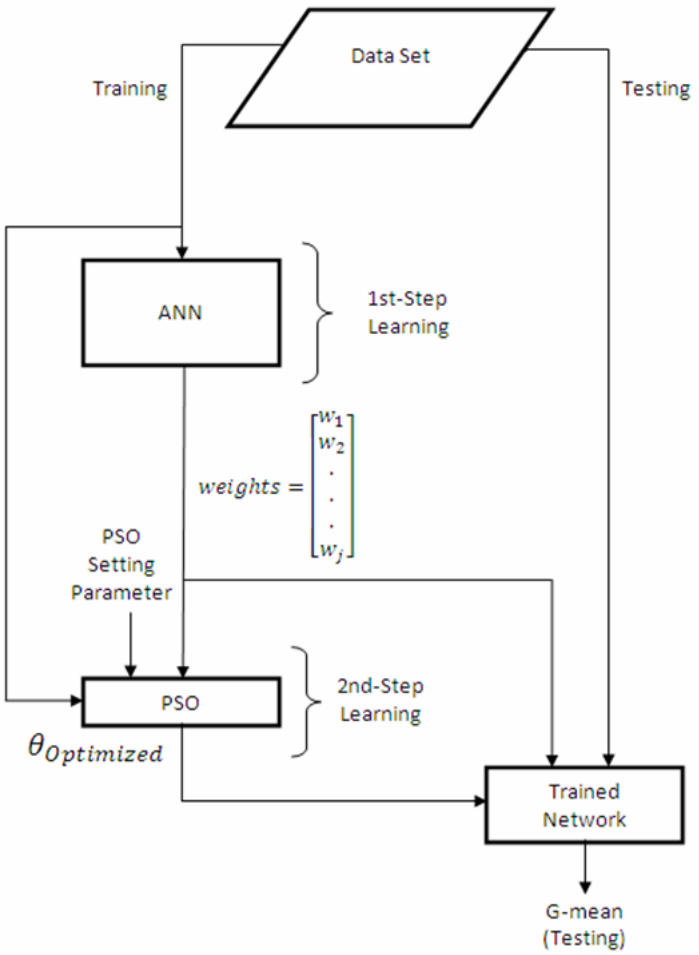


FIGURE 2. The architecture of the proposed two-step supervised learning of ANN for imbalanced dataset problems

3.1. **First step learning.** This section describes the first step learning mechanism of the proposed ANN classifier. The first step of the learning mechanism is similar to the conventional ANN learning algorithm in Section 2. The best weight values, $w_j$, which are obtained from training process, are used as the input to the second step learning and testing process. Note that $j$ is the total number of best weights in the designed network.

3.2. **Second step learning.** The objective of the second step learning mechanism is to further optimize the network. In this study, PSO [19] with dynamic inertia weight (DIW), Equation (6) is employed to find the best value of decision threshold, $\theta_{Optimized}$ during the second step learning, as shown in Figure 3. The PSO parameters that are employed in the second step learning is shown in Table 1. G-mean is used as the fitness function to tune the decision threshold, $\theta$.

$$\omega = \omega_{\max} - \left( \frac{\omega_{\max} - \omega_{\min}}{Maximum\ Iteration} \times Current\ Iteration \right) \tag{6}$$

3.3. **Testing.** As shown in Figure 2, the output of testing dataset is used after the first and the second learning steps are accomplished. Finally, G-mean is calculated to evaluate the classifier's performance for imbalanced dataset.

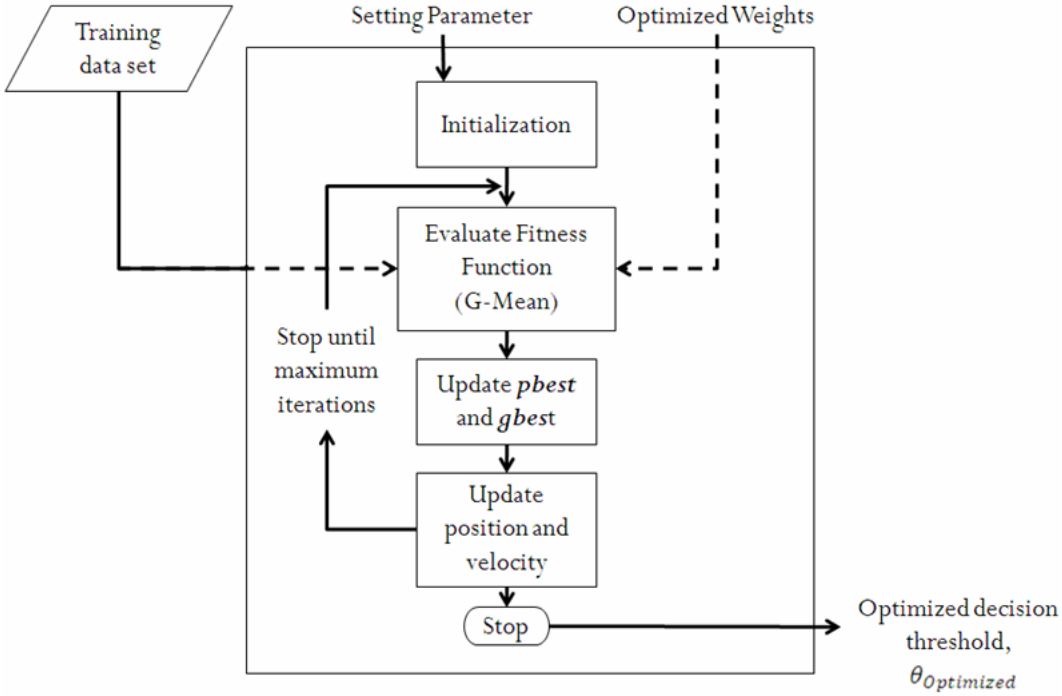


FIGURE 3. Implementation of PSO algorithm during the second-step learning mechanism of the proposed ANN

TABLE 1. PSO setting parameter

| | |
|---|---|
| Number of particles | 20 |
| Dynamic inertia weight, $\omega$ | $(\omega_{\max} \sim \omega_{\min}) = 0.9 \sim 0.4$ |
| Cognitive coefficient, $C_1$ | 1.42 |
| Social coefficient, $C_2$ | 1.42 |
| $r_1$ and $r_2$ | Random $[0, 1]$ |
| Maximum iteration | 200 |

3.4. **Experimental setup.** The experiments were performed using a Pentium (R) Dual-Core 2.60GHz computer with 4GB of RAM. To validate the classifier, several benchmark datasets were used such as Haberman Survival dataset, German Credit dataset, Pima Indian dataset, and Liver Disorder dataset.

In each execution, all of the datasets are divided randomly into the proportions of 60 percent as training samples and 40 percent as testing samples from the entire dataset. The classifier is run 50 times and the average results together with the standard deviation (STDEV) are recorded.

For the ANN classifier, an experimental setup is required to build the architecture. The main structure of the ANN classifier includes the number of neurons in the hidden layer and the activation function inside the neurons at the hidden and output layers. In this study, the numbers of neurons are selected using a trial and error method. The hyperbolic tangent $[-1, 1]$ is used as an activation function at the hidden layer for normalization, while a sigmoid function $[0, 1]$ is located inside the neuron at the output layer.

Other settings for the ANN classifier, such as the number of neurons in the input layer and the total number of weights, are dependent on the dataset. 10 neurons at the hidden layer are set for all datasets. Other parameters, which are shown in Table 3, have been suggested by other researchers. To find the maximum number of iterations requires an investigation. The investigation will set and run for a specific number of iterations and then observing whether the system has converged by at certain iteration number. For example, the experiment is executed 10 times and runs for about 1000 iterations. The result shows that the particles converged and the number of iterations needed for convergence is noted. After 10 times execution, the maximum with which the particles converge by a certain iteration number will determine the maximum number of iterations.

## 4. Experimental Results and Discussion.

4.1. **Haberman's survival dataset.** Table 2 shows a Haberman's survival dataset that is taken from UCI Machine Learning Repository [20]. This dataset consists of three numerical inputs. *Numerical 1* values ranges from 30 to 83, *Numerical 2* values ranges from 58 to 69, and *Numerical 3* values ranges from 0 to 52. The output is in categorical form, which is either 0 or 1. There are 306 collected samples in this data set. 73.5 percent of samples are 0 (majority class) whereas 26.5 percent of samples are 1 (minority class).

Table 2. Haberman's survival dataset

| Unit ID | Numerical 1 | Numerical 2 | Numerical 3 | Output |
|---|---|---|---|---|
| 1 | 30 | 64 | 1 | 0 |
| 2 | 30 | 62 | 3 | 0 |
| 3 | 30 | 65 | 0 | 0 |
| ... | ... | ... | ... | ... |
| 306 | 83 | 58 | 2 | 1 |

4.2. **Implementation of the conventional ANN to Haberman's survival dataset.** The conventional ANN has been implemented to examine Haberman's survival dataset based on the configuration explained in Section 2. In the investigation, the conventional feedforward ANN used $\theta = 0.5$. This result shows that the conventional ANN did not perform well for imbalanced dataset when the decision threshold is fixed to 0.5. Further investigation is done to analyze the contribution of decision threshold, $\theta$, to the G-mean value and the result is shown in Figure 4. Based on this investigation, the best value of decision threshold, $\theta$, is between 0.53 and 0.55. Therefore, in order to obtain the

best value of decision threshold, $\theta$, the second step learning is proposed and the use of an optimization technique, such as PSO, is beneficial in the second step learning of the proposed approach.

4.3. **Implementation of the proposed approach to Haberman's survival dataset.** In general, the high level implementation of the proposed approach to Haberman's survival dataset is shown in Figure 5. The classifier has three inputs, which are the age of patient at time of operation, the patient's year of operation (year-1900), and the number of positive auxiliary nodes detected. The output represents the survival status of each patient, whether the patient died or survived. Based on this dataset, 70 percent of patients survived (majority class) and the remaining were not survived (minority class). The ratio of patient survived and not survived can be written as $0.7 : 0.3$. Similarly, the proposed two-step learning ANN was executed 50 times and the results are shown in Table 3.

In Table 3, *G-mean Train* is the G-mean value obtained after the training whereas *G-mean Test* is the G-mean value obtained after the testing. Table 3 shows that the proposed approach provides better prediction performance than the conventional feedforward ANN. In particular, about 20 percent improvement can be achieved. Table 3 also shows that the proposed approach provides more consistent result, as indicated by smaller value of standard deviation.

Similar pattern can be seen in Table 4, where the investigation was expanded to three other benchmark imbalanced datasets. The proposed two-step learning mechanism of ANN outperforms the conventional ANN by 30 percent to 40 percent improvement. Based on the standard deviation that was recorded in Table 4, the proposed two-step learning performs with inconsistent results for all datasets. Example of convergence curve of second step learning is shown in Figure 6. The convergence curve shows some improvements of G-mean by iterations.

TABLE 3. Comparison of the average G-mean using the conventional ANN and the proposed two-step learning ANN based on Haberman's survival dataset

| Classifier | Conventional ANN, $\theta = 0.5$ | | The proposed ANN | |
|---|---|---|---|---|
| Measurement | G-mean Train (%) | G-mean Test (%) | G-mean Train (%) | G-mean Test (%) |
| Average | 38.87 | 36.04 | 71.26 | 58.67 |
| Maximum Score | 64.63 | 59.16 | 80.16 | 70.47 |
| Standard Deviation | 12.78 | 11.35 | 4.39 | 4.91 |

5. **Conclusions.** This study investigated the contribution of decision threshold to the ANN's performance when solving imbalanced dataset problems. Then, a two-step learning mechanism of ANN is proposed, which consists of the parameter tuning of ANN's weights based on a conventional algorithm and tuning of the decision threshold using PSO. Using several benchmark datasets, the proposed approach provides better classification compared to the conventional ANN. Current investigation includes a combination of the proposed approach with existing deterministic classifier to enhance further the classification performance of the two-step learning ANN.

**Acknowledgment.** This work is financially supported by the UTM-INTEL Research Fund (Vote 73332), Fundamental Research Grant Scheme (FRGS) (Vote 78645), which was awarded by the Ministry of Higher Education to Universiti Teknologi Malaysia, and

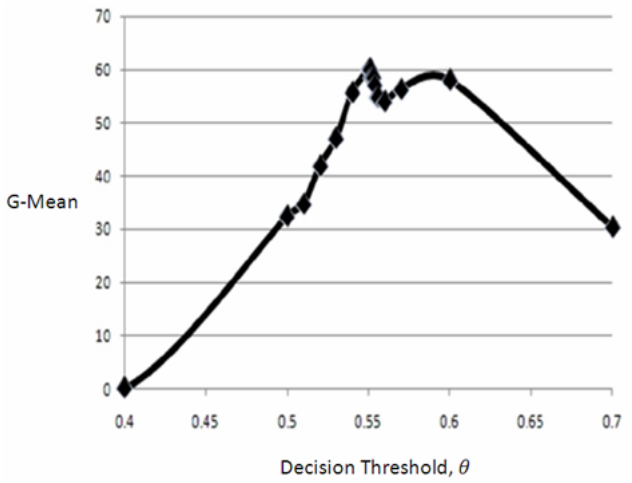


FIGURE 4. G-mean versus decision threshold

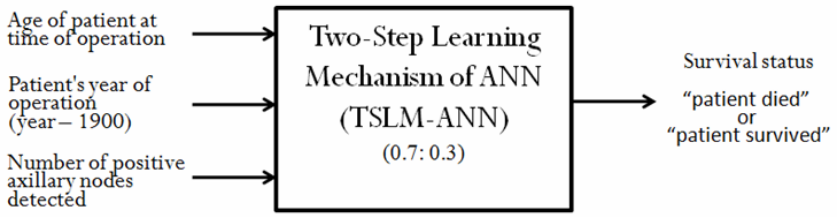


FIGURE 5. High level implementation of the proposed approach to Haberman's survival dataset

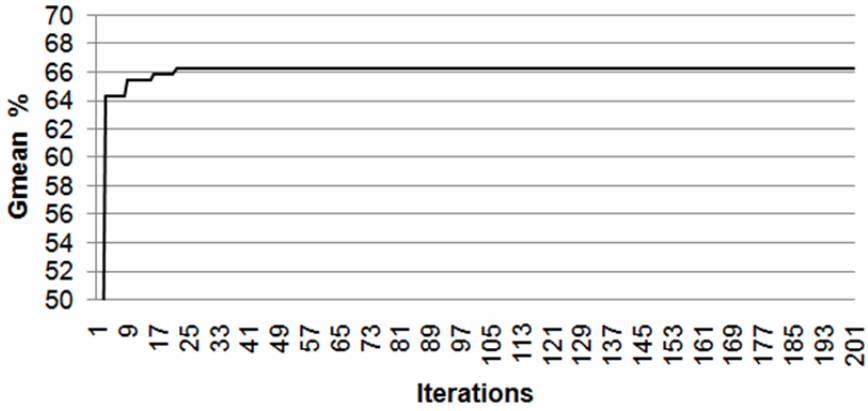


FIGURE 6. Convergence curve of second step learning: G-mean versus iterations

partly supported by UTM GUP Research Fund (Vote Q.J130000.7123.00H67). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation of this paper.

## REFERENCES

[1] D. Cieslak, N. Chawla and A. Striegel, Combating imbalance in network intrusion datasets, *Proc. of IEEE International Conference on Granular Computing*, pp.732-737, 2006.

[2] M. A. Mazurowskia, P. A. Habasa, J. M. Zuradaa, J. Y. Lob, J. A. Bakerb and G. D. Tourassib, Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance, *Advanced in Neural Networks Research: International Joint Conference on Neural Networks*, vol.21, pp.427-436, 2008.

TABLE 4. G-mean values obtained based on German Credit, Pima Indian, and Liver Disorder dataset

| Dataset | Size | #Attribute | #Classes | Class Distribution (minority : majority) | Measurement | Standard ANN, $\theta = 0.5$ | Two-Step Learning of ANN |
|---|---|---|---|---|---|---|---|
| German Credit | 1000 | 24 | 2 | 300 : 700 | Average G-mean (%) | 32.7 | 64.90 |
| | | | | | STDEV | 12.26 | 4.52 |
| Pima Indian | 768 | 8 | 2 | 268 : 500 | Average G-mean (%) | 31.64 | 72.05 |
| | | | | | STDEV | 14.32 | 3.89 |
| Liver Disorders | 345 | 6 | 2 | 145 : 200 | Average G-mean (%) | 14.44 | 63.87 |
| | | | | | STDEV | 13.45 | 5.26 |


[3] B. Anuradha and V. C. Veera Reddy, ANN for clasification of cardiac arrhythmias, *Asian Research Publishing Network Journal of Engineering and Applied Sciences*, vol.3, no.3, pp.1-6, 2008.

[4] L. Bruzzone and S. B. Serpico, A classification of imbalanced remote-sensing data by neural networks, *Pattern Recognition Letters*, vol.18, pp.1323-1328, 1997.

[5] G. H. Nguyen, A. Bouzerdoum and S. L. Phung, A supervised learning approach for imbalanced data sets, *Proc. of the 19th International Conference on Pattern Recognition*, pp.1-4, 2008.

[6] Y. Lu, H. Guo and L. Feldkamp, Robust neural learning from unbalanced data samples, *Proc. of IEEE International Joint Conference on Neural Networks, IEEE World Congress on Computational Intelligent*, vol.3, pp.1816-1821, 1998.

[7] W. K. Yip, K. G. Law and W. J. Lee, Forecasting final/class yield based on fabrication process e-test and sort data, *Proc. of IEEE International Conference on Automation Science and Engineering*, Scottsdale, AZ, pp.478-483, 2007.

[8] Y. L. Murphey, H. Wang, G. Ou and L. A. Feldkamp, OAHO: An effective algorithm for multi class learning from imbalanced data, *Proc. of IEEE International Joint Conference on Neural Networks*, Orlando, FL, USA, 2007.

[9] R. Anand, K. G. Mehrotra, C. K. Mohan and S. Ranka, An improved algorithm for neural network classification of imbalanced training sets, *IEEE Transactions on Neural Networks*, vol.4, no.6, pp.962-969, 1993.

[10] Y. Lu, H. Guo and L. Feldkamp, Robust neural learning from unbalanced data samples, *Proc. of the IEEE World Congress on Computational Intelligence*, Anchorage, AK, USA, pp.1816-1821, 1998.

[11] X. Fu, L. Wang, K. S. Chua and F. Chu, Training RBF neural networks on unbalanced data, *Proc. of the International Conference on Neural Information Processing*, Singapore, 2002.

[12] Z.-H. Zhou and X.-Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, *IEEE Transactions on Knowledge and Data Engineering*, vol.18, no.1, pp.63-77, 2006.

[13] R. Alejo, V. Garcia, J. M. Sotoca, R. A. Mollineda and J. S. Snchez, Improving the classification accuracy of RBF and MLP neural networks trained with imbalanced samples, *Proc. of the 7th International Conference on Intelligent Data Engineering and Automated Learning*, Burgos, Spain, pp.464-471, 2006.

[14] Z. Q. Zhao, A novel modular neural network for imbalanced classification problems, *Patter Recognition Letters*, vol.30, pp.783-788, 2008.

[15] S. Kotsiantis, D. Kanellopoulos and P. Pintelas, Handling imbalanced datasets: A review, *GESTS International Transactions on Computer Science and Engineering*, vol.30, no.1, pp.25-36, 2006.

[16] N. K. Khalid, Z. Ibrahim, T. B. Kurniawan, M. Khalid and N. H. Samin, Function minimization in DNA sequence design based on continuous particle swarm optimization, *ICIC Express Letters*, vol.3, no.1, pp.27-32, 2009.

[17] M. Rashid and A. R. Baig, PSOGP: A genetic programming based adaptive evolutionary hybrid particle swarm pptimization, *International Journal of Innovative Computing, Information and Control*, vol.6, no.1, pp.287-296, 2010.

[18] J.-F. Chang and K.-L. Chen, Applying new investment satified capability index and particle swarm optimization to stock portfolio selection, *ICIC Express Letters*, vol.3, no.3(A), pp.349-354, 2009.

[19] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, Wiley, 2005.
[20] A. Asuncion and D. J. Newman, *UCI Machine Learning Repository*, http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.