Available online at www.sciencedirect.com
SCIENCE DIRECT®

NEUROCOMPUTING

Neurocomputing 61 (2004) 85–97

www.elsevier.com/locate/neucom

Decision support systems using hybrid neurocomputing

Cong Tran[a], Ajith Abraham[b,*], Lakhmi Jain[a]

[a] *School of Electrical and Information Engineering, University of South Australia, Australia*
[b] *Department of Computer Science, Oklahoma State University, USA*

Abstract

This paper suggests a decision support system for tactical air combat environment where not much prior information is available about the decision regions. We proposed a combination of unsupervised learning for clustering the data (to develop decision regions) and a feed forward neural network to classify the decision regions accurately. The clustered data is used as the inputs to the multi-layered feed forward neural network, which is trained using several higher order learning paradigms. Experiment results reveal that the proposed method is efficient.

Keywords: Decision support system; Concurrent hybrid system; Neurocomputing

1. Introduction

Several decision support systems have been developed mostly in various fields including medical diagnosis, business management, control system, command and control of defence, air traffic control and so on [12,19,16]. Usually previous experience or expert knowledge is often used to design decision support systems. Several adaptive learning frameworks for constructing intelligent decision support systems have been proposed [1,3]. To develop an intelligent decision support system, we need a holistic view of the various tasks to be carried out including data and knowledge management (reasoning techniques) [9,18]. The focus of this paper is to develop a tactical air combat decision support system (TACDSS) with minimal prior knowledge, which could also provide optimal decision scores. This paper is an extension of our previous

* Corresponding author.
E-mail addresses: cong.tran@postgrads.unisa.edu.au (C. Tran), ajith.abraham@ieee.org (A. Abraham), lakhmi.jain@unisa.edu.au (L. Jain).

0925-2312/$ - see front matter
doi:10.1016/j.neucom.2004.03.006

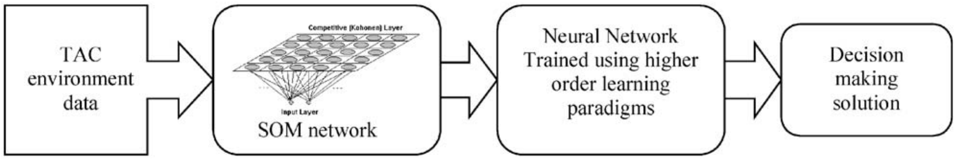


Fig. 1. Concurrent unsupervised and supervised training of neural network for TACDSS.

work where we had implemented evolutionary-fuzzy system [6] and different fuzzy inference methods using different learning techniques [4,5]. As shown in Fig. 1, we propose a concurrent unsupervised neural network to cluster the decision regions and a supervised feed forward neural network trained using different higher-order learning paradigms to automatically generate the decision scores. Section 2 presents the problem of decision making in tactical air combat system. In Section 3, we introduce some theoretical concepts of self-organization map (SOM) followed by feed forward neural network learning paradigms namely Levenberg–Marquardt (LM), quasi-Newton (QN) and scale conjugate gradient (SCG) algorithms [2]. Experimentation results are provided in Section 5 and some conclusions are also provided towards the end.

2. The tactical air combat environment (TACE)

The goal of the project is to develop the decision support system for tactical environment. This section will explain the way to develop the decision support system with the support of cognitive work analysis (CWA) [21]. Asset management in TACE is an important task of the airborne early warning command and control (AEW& C). Its task will be to regularly collect and update information of the environment such as weather, speed and direction of wind for an aircraft. These will create the large amount of information that being processed by operators in a dynamic environment. Other tasks include providing information about the state of assets such as fuel and weapons, information of enemy aircraft/vehicle/ships (quantity and type). Additional procurement of assets may be initiated in a timely manner such that replacement could be achieved immediately when required. Tasks also include coordination with other controllers to ensure that there is sufficient asset at the ground base. There are more tasks that the mission system operator or commander will be required to perform to manage effectively especially when it becomes overwhelming during a real tactical air combat environment as explained by the AEW& C [7].

The decision support system not only requires being intelligent but also should incorporate human machine interaction and consider human as the integral part of the system. The CWA is a system design technique to provide corporation between the human and computing system. CWA is suitable to analyze complex systems that has high-level of cognitive input from human operators, which contributes to the strong success during unpredictable situations and assist hard decision-making. The decision support system should have high level of automation and information integration with a role of operation shift to high task level that involves problem solving, hard decision making, conceptual understanding, planning and workload management.

2.1. The activity context of TACE

There are various activities (work functions) in the TACE system. The work functions could be broadly divided into 4 functions namely (a) mission planning and reporting (b) system set up, configuration and shutdown (c) surveillance and (d) asset control. The major classes of mission context are namely (a) on ground and not in aircraft (b) on ground in aircraft (c) on way to station (d) on station (e) returning to base (f) on ground and in aircraft and (g) on ground not in aircraft [17]. The activity context of AEW& C will be concerned about the crew at each stage of the mission, and their changing preoccupations as the mission progresses. For example, the mission planning is a preoccupation in the earlier phases of a mission and mission reporting is a preoccupation at the later stages of mission. To achieve these objectives, the TACE must incorporate all the priorities and be capable of long-term understanding of the patterns of regional activity, real-time understanding of the current tactical situation, coordinate and maintain safety of assets under its control, and self preservation. The purpose-related functions must be able to provide information related to the work domain including development of the tactical picture, evaluation of the tactical situation, communication, and implementation of protective measures. Finally, AEW& C will be equipped with physical devices such as radar for gathering information from the environment and a mission data-processing computer for storing and processing information. TACE work domain analysis provides physical device solutions at the higher-level work domain functions. Thus if the ability to exchange information and communicate is compromised through deficiencies in the equipment design, then the ability to establish, update, and disseminate the tactical picture and to exercise control over friendly assets is also compromised. However, if the deficiency is in the equipment functioning by reduction of electronic and radio emissions, the presence of platforms will be communicated less effectively, which could result in the reduction of protection level of platform, sensors, and information systems from friend and enemy.

2.2. The simple scenario of tactical air combat environment

Fig. 2 presents a typical scenario of air combat tactical environment. The Airborne Early Warning and Control (AEW& C) is performing surveillance in a particular area of operation. It has two hornets (F/A-18s) under its control at the ground base as shown "+" in the left corner of Fig. 2. An air-to-air fuel tanker (KB707) "□" is on station and the location and status are known to the AEW& C. Two of the hornets are on patrol in the area of Combat Air Patrol (CAP). Sometime later, the AEW& C on-board sensors detects 4 hostile aircrafts (Mig-29) shown as "O". When the hostile aircrafts enter the surveillance region (shown as dashed circle) the mission system software is able to identify the enemy aircraft and its distance from the Hornets in the ground

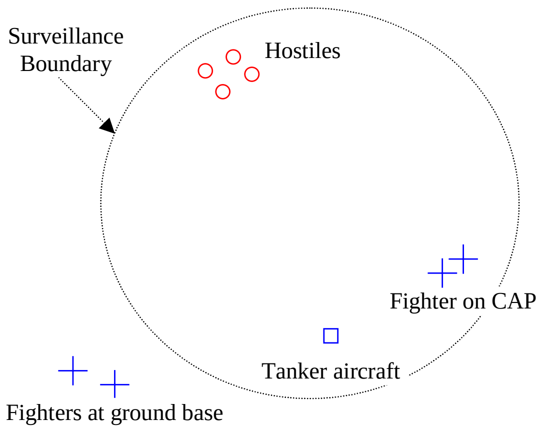


Fig. 2. A simple scenario of the air combat.

base or in the CAP. The mission operator has few options to make a decision on the allocation of hornets to intercept the enemy aircraft.

- Send the Hornet directly to the spotted area and intercept.
- Call the Hornet in the area back to ground base and send another Hornet from the ground base.
- Call the Hornet in the area to refuel before intercepting the enemy aircraft.

The mission operator will base his decisions on a number of decision factors, such as:

- Fuel used and weapon status of hornet in the area.
- Interrupt time of Hornet in the ground base and the Hornet at the CAP to stop the hostile.
- The speed of the enemy fighter aircraft and the type of weapons it posses.
- The information of enemy aircraft with type of aircraft, weapon, number of aircraft.

From the above simple scenario, it is evident that there are several important decision factors of the tactical environment that might directly affect the air combat decision. In the simple tactical air combat, the four decision factors that could affect the decision options for calling the Hornet in the CAP or the Hornet at the ground base are the following:

- "*fuel status*"—quantity of fuel available to perform the intercept,
- "*weapon possession status*"—quantity of weapons available in the Hornet,
- "*interrupt time*"—time required by the hornet to interrupt the hostile, and
- "*danger situation*"—information of the Hornet and the hostile in the battlefield.

Table 1
Decision factors for the tactical air combat

Fuel used	Time intercept	Weapon status	Danger situation	Decision
Full	Fast	Sufficient	Very dangerous	Good
Half	Normal	Enough	Dangerous	Acceptable
Low	Slow	Insufficient	Endanger	Bad

Each factors has difference range of unit such as the *fuel status* (0–1000 l), *interrupt time* (0–60 min), *weapon status* (0–100%) and *danger situation* (0 to 10 points). We used the following two expert rules for developing the rule-based inference system.

- The decision selection will have small value if the *fuel status* is low, the *interrupt time* is slow, the hornet has low *weapon status*, and the *danger situation* is high.
- The decision selection will have high value if the *fuel status* is full, the *interrupt time* is fast, the hornet has high *weapon status* and the *danger situation* is low.

In the air combat environment, decision-making is always based on all states of decision factors. But sometimes, a mission operator or commander could make a decision based on an important factor, such as the fuel used is too low, the enemy has more powerful weapons, quality and quantity of enemy aircraft and so on. Table 1 shows some typical scores (decision selection point) taking into account of the various tactical air combat decision factors.

3. Self-organizing feature maps

Kohonen's self-organizing map (SOM) algorithm introduces the fundamental idea of unsupervised, competitive learning, self-organization, and global ordering [10,11]. Adjusting weights of neurons in a local neighborhood around the winning neuron leads to global ordering through continuous learning. This operation of the SOM algorithm shows the ability of biological neurons that perform global ordering based on local interactions. This global order leads to the creation of natural structures and biologically motivated configurations and shapes, which are created according to laws of minimum energy, time, or complexity [22]. In the SOM learning algorithm, the weight adjustments are provided not only for a winning neuron but also for its *neighbors*. A Competitive learning of the SOM algorithm will find the best matching neuron as the *winner* for a given input vector x. The winning neuron's weight adjustment will also be shared by its neighbors in a certain neighborhood. Thus, the neighboring neurons will learn from the same input vector x. The "*learning rate*" is also called the *neighborhood kernel*. It is a function of both time (iteration step) and the winning neuron spatial neighborhood $N_j(k)$. This spatial neighborhood is a time-dependent function that defines the set of neurons that are topographically close to the winning neuron. The neurons in the spatial neighborhood adjust their weights according to the same learning rule but with amount depending on their position with respect to the winner. After adjusting the weight of the neighborhood neurons of the winning neuron, the SOM algorithm will continue to adjust the weight of the neighbor neurons until the iterations are completed. As a result of weight adjustment, a group of neurons are obtained forming a *cluster* and the algorithm is repeated to search another winning neuron until all the input data is processed. SOM algorithm is used to learn the decision regions (clusters).

4. Learning decision regions using supervised learning paradigms

Once the clusters are defined using SOM, the next step is to apply the supervised learning to train a neural network to learn the different decision regions for the given input data. The supervised learning of neural networks can be viewed as a function optimization problem, wherein higher-order optimization methods using gradient information are used to improve the rate of convergence. In our experiments, we studied the performance of Levenberg–Marquardt (LM) [8], Quasi-Newton (QN) [13] and the scale conjugate gradient (SCG) [15] algorithms.

4.1. Levenberg–Marquardt (LM)

When the performance function has the form of a sum of squares, then the Hessian matrix H can be approximated to $H = J^{\mathrm{T}}J$; and the gradient can be computed as $g = J^{\mathrm{T}}e$, where J is the Jacobian matrix, which contains first derivatives of the network errors with respect to the weights, and e is a vector of network errors. The Jacobian matrix can be computed through a standard backpropagation technique that is less complex than computing the Hessian matrix. The LM algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$x_{k+1} = x_k - [J^{\mathrm{T}}J + \mu I]^{-1} J^{\mathrm{T}} e.$$

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. As Newton's method is more accurate, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. By doing this, the performance function will always be reduced at each iteration of the algorithm [14].

4.2. Quasi-Newton (QN)

The quasi-Newton method was derived from quadratic objective function. The inverse of the Hessian matrix $B = H^{-1}$ is used to bias the gradient direction, B_{t-1} is an old value of B. In the quasi-Newton training method, the weights are updated using,

$$W_{t+1} = W_t - \eta B_t g_t.$$

B matrix here is not computed. It is successively estimated employing rank 1 or rank 2 updates, following each line search in a sequence of search directions,

$$B_t = B_t + \Delta B_t.$$

The two major formulas to compute ΔB_t as follows:

$$\Delta B_t = \frac{dd^{\mathrm{T}}}{d^{\mathrm{T}}\Delta g} - \frac{B_t - 1\Delta g \Delta g^{\mathrm{T}} B_t - 1}{\Delta g^{\mathrm{T}} B_t - 1\Delta g} \tag{1}$$

or

$$\Delta B_t = \left(1 + \frac{\Delta g^{\mathrm{T}} B_t - 1\Delta g}{d^{\mathrm{T}}\Delta g}\right)\frac{dd^{\mathrm{T}}}{d^{\mathrm{T}}\Delta g} - \frac{d\Delta g^{\mathrm{T}} B_t - 1 + B_t - 1\Delta g d^{\mathrm{T}}}{d^{\mathrm{T}}\Delta g}, \tag{2}$$

where $d = w_t - w_{t-1}$ and $\Delta g = g_t - g_{t-1}$.

Eq. (1) is called the DFP (Davidon–Fletcher–Powell) formula and (2) is called the BFGS (BroydenFletcher–Goldfarb–Shannon) method.

4.3. Scale conjugate gradient (SCG) algorithm

The conjugate gradient methods are originally derived from quadratic minimization and the minimum of the objective function E can be efficiently found within N iterations. With initial gradient $g_{\mathrm{initial}} = \delta E/\delta w|_{W=W_{\mathrm{initial}}}$, and direction vector $d_{\mathrm{initial}} = -g_{\mathrm{initial}}$, the conjugate gradient method recursively constructs two vector sequences as follows:

$$g(t+1) = g(t) + \lambda(t)Hd(t),$$

$$d(t+1) = -g(t+1) + \gamma(t)d(t),$$

where

$$\lambda(t) = \frac{g(t)^{\mathrm{T}} g(t)}{d(t)^{\mathrm{T}} Hd(t)}$$

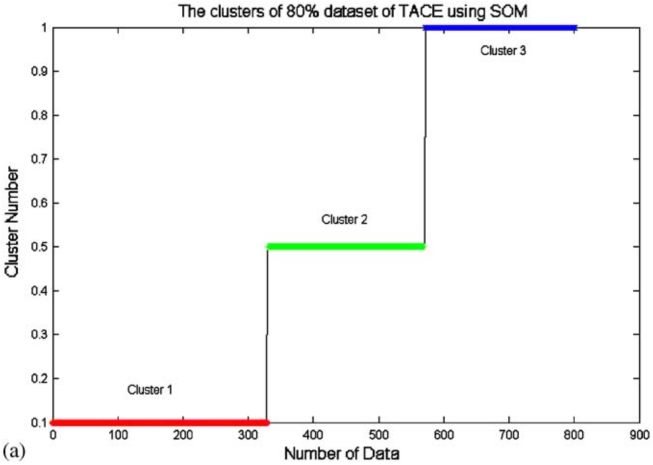
and

$$\gamma(t) = \frac{g(t+1)^{\mathrm{T}} g(t+1)}{g(t)^{\mathrm{T}} g(t)},$$

where t is the time instant and $t+1$ is next iteration time, d is called the conjugate direction and H is the Hessian matrix of the objective function E. To reduce the computational time, a scaled conjugate gradient (SCG) algorithm was developed by Moller [15]. Moller's approach avoids the line search per learning iteration by using Levenberg–Marquardt way of scaling the step size.

5. Experiment results and performance analysis

The master data set is in accordance with Table 1 comprises of 1000 datasets. Each record consists of the TACE data namely '*fuel status*', '*interrupt time*', '*weapon*



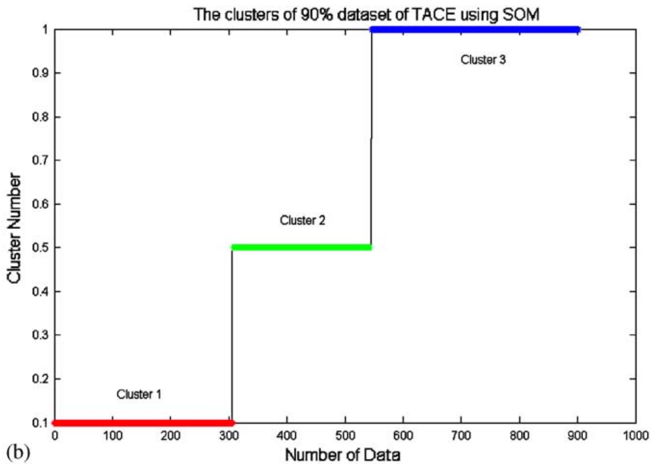


Fig. 3. Developed TACE clusters using SOM (a) 80% and (b) 90% data.

status' and '*danger situation*'. To avoid any bias on the data, from the master dataset, we randomly created two sets of training (Dataset A—90% and Dataset B—80%) and test data (10% and 20%), respectively. All the experimentations were repeated three times and the average errors are reported.

5.1. Unsupervised training of SOM

We used the Matlab SOM toolbox developed by Vesanto [23,24]. SOM analysis provided three clusters: C_1, C_2 and C_3. The developed clusters for the two data sets A and B are shown in Figs. 3 (a) and (b).

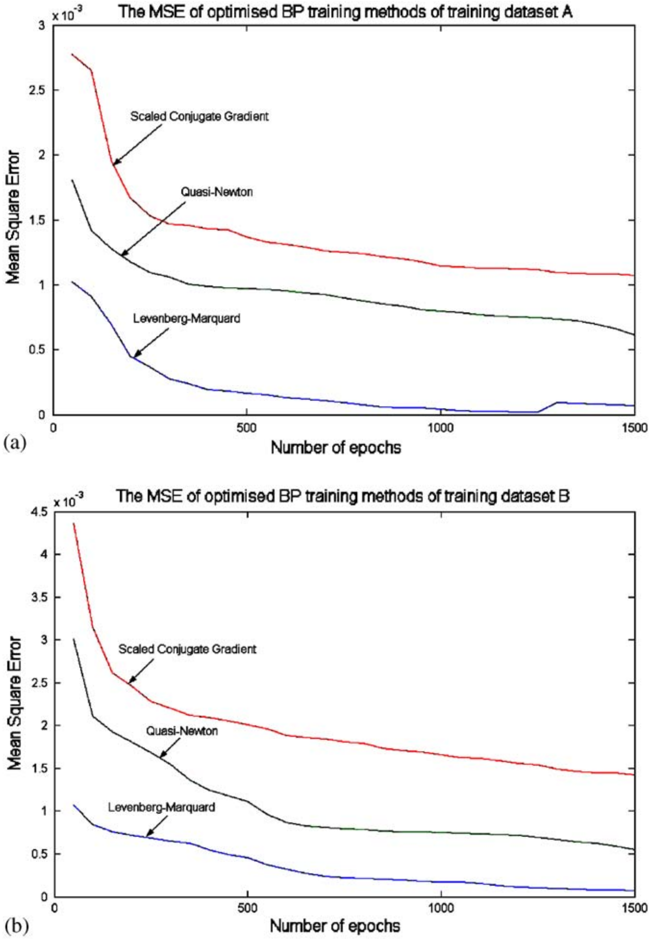


Fig. 4. The MSE of training dataset A and B for the three training methods.

5.2. Learning the decision regions

We used a feed-forward neural network trained [20] using LM, QN and SCG algorithms. Given input data, the network was supposed to learn the three clustered decision regions. The three clusters were assigned numerical values of 0.1, 0.5 and 1, respectively. We adopted a trial and error approach to decide the number of hidden layers. Figs. 4 (a) and (b) illustrate the convergence of mean squared error for (a) dataset A and (b) dataset B using LM, QN and SCG approaches.

The comparison of actual and target output for both Datasets A and B using the three approaches are illustrated in Figs. 5 and 7 (training) and Figs. 6 and 8 (testing). Figs. 5–8 Table 2 shows the RMSE of training and test performance. Empirical results

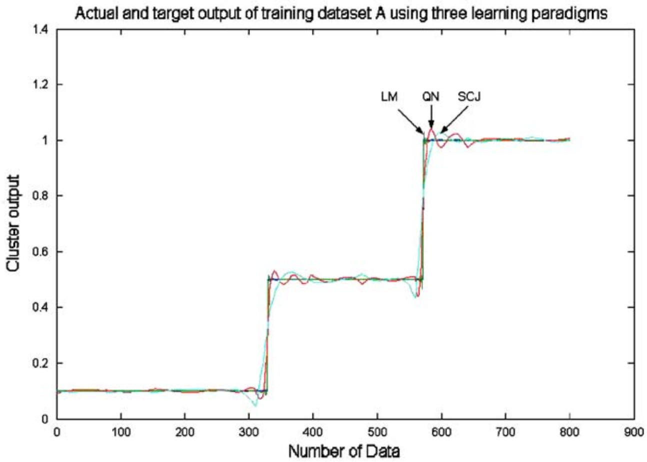


Fig. 5. Actual and target output of training dataset A using three learning paradigms.

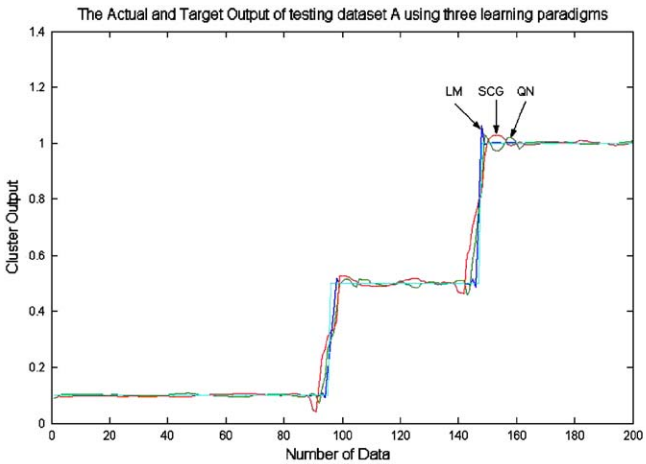


Fig. 6. Actual and target output of test dataset A using three learning paradigms.

reveal that even though LM gave the best training error for both datasets, SCG gave the best generalization performance.

6. Conclusion

In this paper, we proposed a hybrid unsupervised–supervised training method to develop an intelligent decision support system for a tactical air combat environment when no priori information about the decision regions are available. We also investigated the

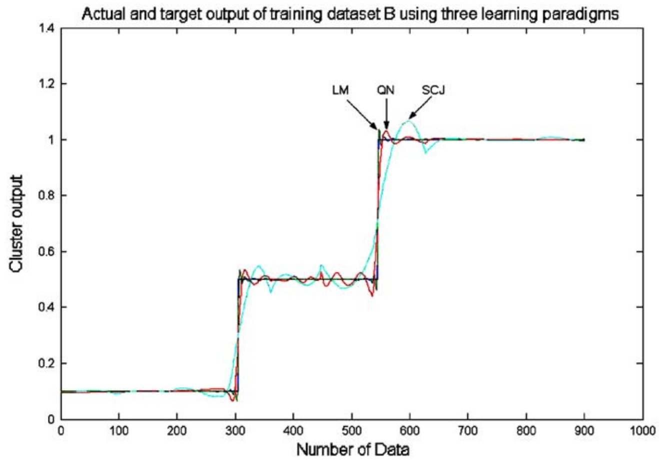


Fig. 7. Actual and target output of training dataset B using three learning paradigms.

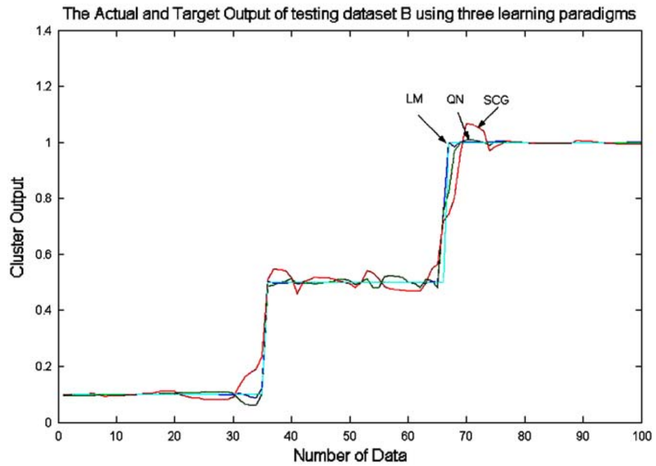


Fig. 8. Actual and target output of test dataset B using three learning paradigms.

Table 2
Comparative performance of the different learning algorithms

Learning method	Training Dataset A	Dataset B	Test Dataset A	Dataset B
LM	0.0028	0.0085	0.0253	0.0253
QN	0.0209	0.0223	0.0357	0.0328
SCG	0.0327	0.0429	0.0470	0.0484

performance of three neural network-learning paradigms to learn the different decision regions. Even though LM gave the best training performance in spite of the computational complexity, SCG gave the best generalization performance for both datasets. Our future research will incorporate several other clustering algorithms, Bayesian reasoning and hidden Markov decision models to further granulate the decision regions and improve the overall performance.

References

[1] A. Abraham, Neuro-fuzzy systems: state-of-the-art modelling techniques, in: J. Mira, A. Prieto (Eds.), Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence, Lecture Notes in Computer Science, Vol. 2084, Springer, Germany, Granada, Spain, 2001, pp. 269–276.
[2] D.J.L. Alamo, A comparison among eight different techniques to achieve an optimum estimation of electrical grounding parameters in two-layered earth, IEEE Trans. Power Delivery 8 (4) (1993) 1890–1899.
[3] R. Cattral, F. Oppacher, D. Deogo, Rule acquisition with a genetic algorithm, Proceedings of the Congress on Evolution computation, CEC99, 1, Washington, DC, 1999, pp. 125–129.
[4] C. Tran, A. Abraham, L. Jain, TACDSS: adaptation using a hybrid neuro-fuzzy system, in: J.M. Benitez, et al., (Eds.), Seventh Online World Conference on Soft Computing in Industrial Applications (WSC7), Advances in Soft Computing: Engineering Design and Manufacturing, Springer, Germany, 2002, pp. 53–62.
[5] C. Tran, A. Abraham, L. Jain, Adaptation of mamdani fuzzy inference system using neuro-genetic approach for tactical air combat decision support system, 15th Australian Joint Conference on Artificial Intelligence (AI'02), Lecture Notes Artificial Intelligence, Vol. 2557, Springer, Germany, 2002, pp. 672–679.
[6] C. Tran, L. Jain, A. Abraham, Adaptive database learning in decision support system using evolutionary fuzzy systems: a generic framework, Hybrid Information Systems, Advances in Soft Computing, Physica Verlag, Germany, 2001, pp. 237–252.
[7] C. Tran, Q. Zahid, Intelligent techniques for decision support in tactical environment Land Warfare Conference 2000, Proceedings of the Warfighting in Complex Terrain, Melbourne, published by Australia Defence Science & Technology Organisation, 2000, pp. 403–411.
[8] H. Demuth, M. Beale, Neural network Toolbox User's Guide, The Math Works Inc, Natick, MA, 2001.
[9] S.B. Eom, Decision support systems research: current state and trends, Ind. Manage. Data Systems 99 (5) (1999) 213–221.
[10] T. Kohonen, Self organized formation of topologically correct feature aps, Biol. Cybern. 43 (1982) 59–69.
[11] T. Kohonen, Self-Organizing Maps, 1st Edition, Springer, Berlin, 1995, 2nd Edition, 1997.
[12] P.A. Leal de Matos, P.L. Powell, Decision support for flight re-routing in Europe, Decision Support Systems 34 (4) (2003) 397–412.
[13] C.G. Looney, Advances in feedforward neural networks: demystifying knowledge acquiring black boxes, IEEE Trans. Knowledge Data Eng. 8 (2) (1996) 211–226.
[14] T. Masters, Advanced Algorithms for Neural Networks: a C++ Sourcebook, Wiley, New York, 1995.
[15] M.F. Moller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks 6 (1993) 523–533.
[16] G.P. Moynihan, P. Purushothaman, R.W. McLeod, W.G. Nichols, DSSALM: a decision support system for asset and liability management, Decision Support Systems 33 (1) (2002) 23–38.
[17] N. Naikar, P.M. Sanderson, Evaluating system design proposals with work domain analysis, Hum. Factors 43 (4) (2001) 529–542.
[18] H.R. Nemati, D.M. Steiger, L.S. Iyer, R.T. Herschel, Knowledge Warehouse: an architectural integration of knowledge management, decision support, Decision Support System 33 (2) (2002) 143–161.
[19] K.N. Papamichail, S. French, Explaining and justifying the advice of a decision support system: a natural language generation approach, Expert Systems Appl. 24 (1) (2003) 35–48.

[20] Parker, B. David, Learning-logic, Report TR-47, Center for Computational Research in Economics and Management Science, Massachusetts Institute of Technology, Cambridge, MA, 1985.
[21] J. Rasmussen, A. Pejtersen, L.P. Goodstein, Cognitive Systems Engineering, Wiley, New York, 1994.
[22] A.A. Saveiev, D.V. Dobrinin, Hierarchical multispectral image classification based on self-organized maps, Proceedings of IGARSS'99, Hamburg, Germany, 1999, pp. 2510–2512.
[23] J. Vesanto, SOM-based data visualization methods, in: Intelligent Data Analysis, 3(2) Amsterdam, 1999, pp. 111–126.
[24] J. Vesanto, Esa Alhoniemi, Clustering of the self-organizing map, in: IEEE Trans. Neural Networks 11(3) (2000) 586–600.