

Research Article

Wavelet Neural Network Using Multiple Wavelet Functions in Target Threat Assessment

Gaige Wang,^{1,2} Lihong Guo,¹ and Hong Duan³

¹Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China

²Graduate School of Chinese Academy of Sciences, Beijing 100039, China

³School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China

Correspondence should be addressed to Lihong Guo; guolh@ciomp.ac.cn

Received 13 December 2012; Accepted 15 January 2013

Academic Editors: J. Bajo and Q. Zhao

Copyright © 2013 Gaige Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Target threat assessment is a key issue in the collaborative attack. To improve the accuracy and usefulness of target threat assessment in the aerial combat, we propose a variant of wavelet neural networks, MFWFNN network, to solve threat assessment. How to select the appropriate wavelet function is difficult when constructing wavelet neural network. This paper proposes a wavelet mother function selection algorithm with minimum mean squared error and then constructs MFWFNN network using the above algorithm. Firstly, it needs to establish wavelet function library; secondly, wavelet neural network is constructed with each wavelet mother function in the library and wavelet function parameters and the network weights are updated according to the relevant modifying formula. The constructed wavelet neural network is detected with training set, and then optimal wavelet function with minimum mean squared error is chosen to build MFWFNN network. Experimental results show that the mean squared error is 1.23×10^{-3} , which is better than WNN, BP, and PSO_SVM. Target threat assessment model based on the MFWFNN has a good predictive ability, so it can quickly and accurately complete target threat assessment.

1. Introduction

With the development of science and technology, the requirement of information is increasingly improving in modern warfare. To adapt to this change, many countries have begun the research of multisensor information fusion from the 1970s. After years of research, the United States, Britain, and other military powers have developed a number of information fusion systems which can be used for combat. Target threat assessment belongs to the third level in information fusion model and is a kind of high-level information fusion. The target threat assessment is the essential basis for the allocation of force and fire in C4ISR system.

The traditional methods to solve threat assessment are Bayesian inference [1, 2], multiattribute decision-making theory [3], GSOBP [4], Elman_AdaBoost [5], analytic hierarchy process [6], Dempster-Shafer theory [7], Hypothesis-drive [8], and so forth. These methods are based on constant weight

vector and must rely on expertise available, which makes these methods significantly increase subjective factor of threat assessment, and it is highly possible to evaluate inaccurate results, making the complex relationship between evaluation indicators not effectively reflected. In addition, the models created by these methods have a fatal drawback; that is, these models do not have the self-learning and adaptive capacity, so it is difficult to adapt to change as the change of the enemy air attack weapons performance and tactical means brought to the weights of each factor changes. As the neural network has many advantages, such as strong learning ability and adaptability, it is adept at working out the target threat assessment compared with the above-mentioned methods. BP is a mature and effective method, with the advantages of rigorous derivation process, solid theoretical basis, strong versatility, and clear physical concepts. Literature [7, 8] studies threat assessment using BP networks, and achieve good results. However, with the dimension of training

data increasing, the convergence of BP slows down, and the network performance deteriorates; moreover, in the training process, it is easy to fall into local minimum solution. Wavelet neural network (WNN) has many advantages compared with other neural networks; for example, the parameters (hidden nodes and weight) are more easily determined than the radial basis function (RBF) neural networks; it requires smaller training amount than multilayer perceptron network; also, wavelet neural network has a fast convergence. In the same approximation quality, the wavelet neural network requires fewer nodes. For WNN, one of the biggest drawbacks is the difficulty of the choice of mother wavelet function, so this paper proposes an algorithm for selecting optimal mother wavelet function. And then, we construct the (Multiple Wavelet Function Wavelet Neural Networks) MWFNN using the above method to select optimal mother wavelet function for threat assessment.

2. MWFNN Network

2.1. Wavelet Theory. Firstly proposed by Grossman and Morlet in the 1980s, wavelet theory [9] is a mathematical theory and analysis method to make up the shortages of Fourier transform. In the field of signal processing, the most widely used analysis method is the Fourier transform, but it has obvious deficiency that the Fourier transform has no distinguishable ability in the time domain, because the time information is not included in the results of Fourier transform. Wavelet is special waveform with the mean 0 and the limited length.

Wavelet function is constructed through a series of basic transformation with a mother wavelet function. Not all functions can be used as wavelet mother function if a wavelet function is to be available and then develop into a good wavelet transform function, it must satisfy many conditions. Therefore, it is difficult to find the practical wavelet function. In the practical wavelet functions, some of them do not have expressions.

Let $\varphi(t)$ be a square integrable function, that is, $\varphi(t) \in L^2(R)$. If its Fourier transform $\Psi(\omega)$ can satisfy the following compatibility condition:

$$\int_R \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty \quad (1)$$

then $\varphi(t)$ is called a basic wavelet or mother wavelet function. We make translation and scale for wavelet function, the translation factor τ , and the scale factor (also known as the expansion factor) a , so that we get function $\varphi_{a,\tau}(t)$:

$$\varphi_{a,\tau}(t) = a^{1/2} \Psi\left(\frac{t-\tau}{a}\right), \quad a > 0, \tau \in R. \quad (2)$$

As the translation factor τ and the scale factor a are continuous variables, their value can be positive or negative; so $\Psi_{a,\tau}(\omega)$ is called continuous wavelet function (also called the mother wavelet function).

Wavelet transform calculates the inner product between the signal $x(t)$ with mother wavelet function

$$f_x(a, \tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \varphi^*\left(\frac{t-\tau}{a}\right) dt. \quad (3)$$

Equivalent expression in time domain is given as

$$f_x(a, \tau) = \frac{\sqrt{a}}{2\pi} \int_{-\infty}^{\infty} X(\omega) \Psi^*(a\omega) e^{j\omega\tau} d\omega, \quad (4)$$

where $a > 0$, $\tau \in R$, $X(\omega)$ and $\Psi(\omega)$ are the Fourier transform of $x(t)$ and $\varphi(t)$, respectively.

The conclusion can be drawn from (3) and (4) that is wavelet analysis can analyze the local characteristics of the signal through the mother wavelet function transformation; therefore, wavelet theory is considered to be the breakthrough for the Fourier transform, and the theory has been successfully applied to image processing, optical devices detection, and signal analysis and other fields.

2.2. Wavelet Neural Network. Wavelet transform has time-frequency localization property and focal features and neural network (NN) has self-adaptive, fault tolerance, robustness, and strong inference ability. How to combine the advantages of wavelet transform and NN to solve practical problems has been one of the hot spots. So-called wavelet neural network (WNN) or wavelet network (WN) is a variety of two techniques and inherits the advantages of the neural network and wavelet transformation. Proposed by Q. Zhang in 1992 [10], WNN uses the wavelet function as the activation function instead of the Sigmoid activation function.

For WNN, its topology is based on BP network; the transfer function of hidden layer nodes is the mother wavelet function; and the network signal is prior to transmission while error is backpropagation in the training process. The network topology is shown in Figure 1. In Figure 1, x_1, x_2, \dots, x_n is the input vector; y_1, y_2, \dots, y_l is the predicted output; ω_{ij} and ω_{kj} are the weights connecting every layer; and h_j is mother wavelet function.

For the input signal sequence $x = (x_1, x_2, \dots, x_n)$, the output of the hidden layer is calculated as

$$h(j) = h_j \left[\frac{\sum_{i=1}^n \omega_{ij} x_i - b_j}{a_j} \right], \quad j = 1, 2, \dots, m, \quad (5)$$

where $h(j)$ is output value for the node j in the hidden layer; h_j is the mother wavelet function; ω_{ij} is weight connecting the input layer and hidden layer; b_j is the shift factor, and a_j is the stretch factor for h_j .

Currently, the choice of mother wavelet functions has not yet formed a standard theory; commonly used wavelet functions are Morlet, Haar, Daubechies (dbN), Symlet (symN), Meyer, Coiflet, Biorthogonal wavelets, and so on.

The output of the output layer is calculated as

$$y(k) = \sum_{i=1}^m \omega_{ik} h(i), \quad k = 1, 2, \dots, l, \quad (6)$$

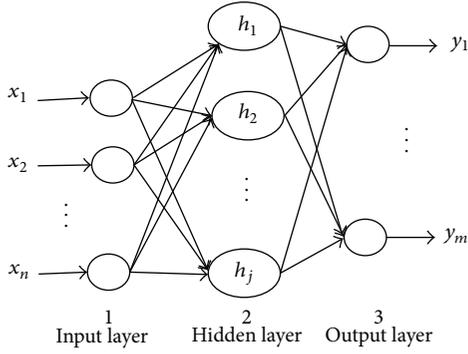


FIGURE 1: Topology of wavelet neural network.

where $h(i)$ is the output value for node i in the hidden layer; ω_{ik} is weight connecting the hidden layer and output layer; l and m are the number of nodes for output layer and the hidden layer, respectively.

For WNN, the updating weight algorithm is similar to BP network; the gradient method is used to update mother wavelet function parameters and connection weights between the layers, making the prediction output closer and closer to the desired output. The weights of WNN and the parameters of wavelet function are updated as follows.

(1) Calculating the prediction error of WNN

$$e = \sum_{k=1}^m yn(k) - y(k), \quad (7)$$

where, $y(k)$ is the predicted output value, $yn(k)$ is the expected output value for the network.

(2) Updating the weights of WNN and the parameters of wavelet function according to the prediction error e

$$\begin{aligned} \omega_{n,k}^{(i+1)} &= \omega_{n,k}^{(i)} + \Delta\omega_{n,k}^{(i+1)}, \\ a_k^{(i+1)} &= a_k^{(i)} + \Delta a_k^{(i+1)}, \\ b_k^{(i+1)} &= b_k^{(i)} + \Delta b_k^{(i+1)}, \end{aligned} \quad (8)$$

where $\Delta\omega_{n,k}^{(i+1)}$, $\Delta a_k^{(i+1)}$, and $\Delta b_k^{(i+1)}$ are calculated by the network prediction error:

$$\begin{aligned} \Delta\omega_{n,k}^{(i+1)} &= -\eta \frac{\partial e}{\partial \omega_{n,k}^{(i)}}, \\ \Delta a_k^{(i+1)} &= -\eta \frac{\partial e}{\partial a_k^{(i)}}, \\ \Delta b_k^{(i+1)} &= -\eta \frac{\partial e}{\partial b_k^{(i)}}, \end{aligned} \quad (9)$$

where η is the learning rate.

The process of training WNN is as follows

(1) Data preprocessing: first, the original data is quantified and normalized, and then the data is divided into training set and testing set for network training and testing, respectively.

- (2) Initializing WNN: connection weights ω_{ij} and ω_{jk} , translation factor b_k , and scale factor a_k are randomly initialized, and the learning rate η is set.
- (3) Training network: input the training set into WNN, compute network predicted output values, and calculate the error e between output and the expected value.
- (4) Updating the weights: update mother wavelet function parameters and network weights according to the prediction error e , making the predictive value of the network as close to actual values.
- (5) If the results satisfy the given conditions, use the testing set to test the network, otherwise, return to Step 3.

2.3. MFWWNN. MFWWNN will be provided in this section.

2.3.1. MFWWNN Algorithm

Step 1. Initializing: initialize mother wavelet function library $\text{waveFunction} = \{\text{waveFunction}_i\}$, $i = 1, 2, \dots, K$, $K = \|\text{waveFunction}\|$ is the number of elements waveFunction included, the variance δ_i for mother wavelet functions waveFunction_i , and optimal wavelet function $\text{waveFunction}_{\text{best}} = \text{waveFunction}_1$ and its variance $\delta_{\text{best}} = \delta_1$.

Step 2. Choosing the best mother wavelet function: for each mother wavelet function,

- (i) Update the weights and parameters of wavelet function waveFunction_i according to (7)–(9).
- (ii) if $\delta_i < \delta_{\text{best}}$

$$\begin{aligned} \delta_{\text{best}} &= \delta_i \\ \text{waveFunction}_{\text{best}} &= \text{waveFunction}_i \end{aligned} \quad (10)$$

End

Step 3. Constructing MFWWNN using $\text{waveFunction}_{\text{best}}$ as mother wavelet function.

Step 4. Testing constructed MFWWNN network in Step 3 using the testing set.

Step 5. Analyzing results.

3. Target Threat Assessment Using MFWWNN

Strictly speaking, threat assessment is an NP-hard problem, belonging to the third level in the JDL information fusion model. Target threat assessment needs to consider many factors (such as geography, weather, enemy, etc.), and the relation among the various factors is not a simple linear combination and it is difficult to determine a function between the target threat value and various factors. Therefore, we must consider various factors and their relationships when studying the threat assessment. However, we consider

the following six factors in general: target type, target speed, target heading angle, target height, and target distance. We will test the performance of MFWFNN using these factors in this paper.

3.1. Target Threat Assessment Factor. We mainly consider the following six key factors when studying the target threat assessment in the paper:

- (1) Target Type: large targets (such as fighter-bombers), small targets (such as stealth aircraft, cruise missiles), and helicopters;
- (2) Target heading angle: such as 22° , 26° , and 6° ;
- (3) Target speed: such as 100 m/s, 500 m/s, and 220 m/s;
- (4) Target height: such as very low, low, medium, and high;
- (5) Target interference: such as strong, medium, and weak;
- (6) Target distance: such as 100 km, 110 km, and 220 km.

3.2. Threat Assessment Model Using MFWFNN. We design MFWFNN model according to the data characteristics. Because the data is 6-dimensional, and the output is 1-dimensional, the structure of WNN is 6-12-1. Firstly, we input six indicators that are the target type, target speed, target heading angle, target interference, target height, and the distance to the input layer. The hidden layer nodes are formed by the wavelet function, and the output layer outputs predicted target threat assessment value under the current indicators. On the basis of the above analysis, we construct the target threat assessment model based on MFWFNN with these six selected indicators, and its architecture is shown in Figure 2.

4. Model Simulation

In this section, we will test the target threat assessment model using MFWFNN proposed in Section 3.

4.1. Data Preprocessing. Part of the data used in our work is shown in Table 1. Target value is quantified using G. A. Miller's quantitative theory, which represents the degree of threat: extremely small, very small, little small, small, medium, large, little large, very large, and extremely large. The properties of the specific quantitative criteria are quantified as follows.

- (1) Target Type: helicopter, large target (such as fighter-bombers), and small targets (such as stealth aircraft, cruise missiles), are quantified by 3, 5, and 8, respectively;
- (2) Target interference: strong, medium, weak, and no are quantified by 8, 6, 4, and 2, respectively;
- (3) Target height: very low, low, medium, and high are quantified 8, 6, 4, and 2, respectively;
- (4) Target speed: 0 m/s~1800 m/s equal interval (200 m/s) is quantified by 9 to 1.

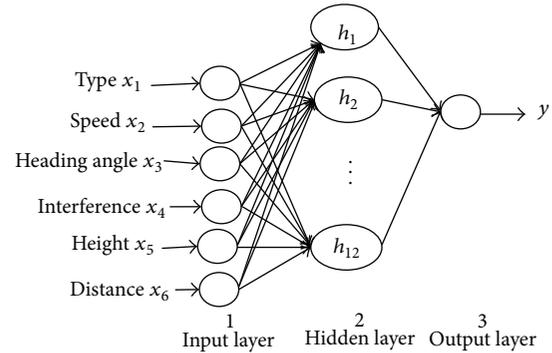


FIGURE 2: Architecture for the model of target threat assessment based on MFWFNN.

- (5) Target heading angle: $0^\circ \sim 36^\circ$ equal interval (4°) is quantified by 9 to 1.
- (6) Target distance: 0 km~450 km equal interval (50 km) is quantified by 9 to 1.
- (7) Determining the target output: firstly normalize the various factors from air combat situation and then put them into the WMF_WNN proposed in Section 3.2. At last, the WMF WNN outputs threatening assessment value.

After quantifying the data, we can normalize the training set and testing set using the following expression:

$$f: x \longrightarrow y = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \quad (11)$$

where $x, y \in R^n$, $x_{\min} = \min(x)$, $x_{\max} = \max(x)$. The values are converted into the range $[0, 1]$ through the normalization, that is, $x_{\max} \in [0, 1]$, $i = 1, 2, \dots, n$.

4.2. Analysis of Simulation Results. In this paper, we implement the MFWFNN algorithm by MATLAB R2009a with the CPU Pentium (R) 4 3.06 GHz, 1 G memory ($2 * 512$ M). The network prediction is compared with WNN, PSO_SVM and BP neural network. The results show that the proposed network is superior to the WNN, PSO_SVM and BP neural network.

4.2.1. Creating the Mother Wavelet Function Library. Because many mother wavelet functions have no specific expression, we cannot work out their derivatives. Therefore, in this work, we only use the following seven mother wavelet functions to build a library waveFunction. Their expressions are as follows:

- (1) Haar wavelet function:

$$\Psi(t) = \begin{cases} 1, & 0 \leq t \leq \frac{1}{2} \\ -1, & \frac{1}{2} \leq t \leq 1 \\ 0, & \text{other.} \end{cases} \quad (12)$$

TABLE 1: Part of data.

No.	Type	Velocity (m/s)	Heading angle (°)	Inference	Height	Distance (km)	Threat value
1	Large	450	8	Medium	Low	300	0.5843
2	Large	400	3	Strong	High	100	0.5707
3	Large	450	16	Medium	Low	200	0.5333
4	Large	800	4	Strong	High	100	0.6895
5	Large	800	12	Strong	Low	320	0.6896
6	Small	530	6	Strong	Medium	230	0.6056
7	Small	650	8	Strong	Medium	200	0.7425
8	Small	700	12	Strong	Low	320	0.7336
9	Small	750	15	Medium	Very low	400	0.7541
10	Small	640	18	Strong	Medium	280	0.6764
11	Helicopter	90	12	Weak	Very low	320	0.3937
12	Helicopter	110	3	No	Medium	100	0.3927
13	Helicopter	100	9	No	Medium	260	0.3351
14	Helicopter	120	15	No	Low	160	0.3586
15	Helicopter	80	6	Weak	High	180	0.3471

TABLE 2: Predicting results of different mother wavelet function.

Wavelet function	Haar	Gaussian	Morlet	Mexihat	Shannon	Meyer	GGW
MSE	2.10×10^{-2}	1.03×10^{54}	1.23×10^{-3}	1.27×10^{-3}	1.11×10^{55}	3.90×10^{57}	1.60×10^{-2}
Running time (s)	4.74	4.76	4.85	4.88	4.84	5.24	4.88

(2) Gaussian wavelet function:

$$\Psi(t) = \frac{t}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right). \tag{13}$$

(3) Morlet wavelet function:

$$\Psi(t) = \cos(1.75t) \exp\left(-\frac{t^2}{2}\right). \tag{14}$$

(4) Mexican Hat (Mexihat) wavelet function:

$$c = \frac{2}{\sqrt{3}}\pi^{-1/4}, \tag{15}$$

$$\Psi(t) = c(1-t^2) \exp\left(-\frac{t^2}{2}\right).$$

(5) Shannon wavelet function:

$$\Psi(t) = \frac{\sin \pi(t-1/2) - \sin 2\pi(t-1/2)}{\pi(t-1/2)}. \tag{16}$$

(6) Meyer wavelet function (approximate formula):

$$\Psi(t) = 35t^4 - 84t^5 + 70t^6 - 20t^7. \tag{17}$$

(7) Wavelet function GGW constructed by the authors:

$$\Psi(t) = \sin(3t) + \sin(0.3t) + \sin(0.03t). \tag{18}$$

We use the following parameters to initialize the network: the input layer nodes $M = 6$, the hidden layer nodes $n = 12$, the output nodes $N = 1$, and parameter learning rates $lr_1 = 0.01$ and $lr_2 = 0.001$, respectively.

We construct wavelet neural network using each wavelet function in wavelet function library and then input training set into network and train the network. The MSEs are as follows (from small to large): Morlet < Mexihat < GGW < Haar < Gaussian < Shannon < Meyer, as shown in Table 2. From Table 2, we can draw the conclusion that running time and MSE of Morlet and Mexihat are slightly different, but the MSE of Gaussian, Shannon, and Meyer is extremely great, completely divorced from reality. The results we got in this paper are consistent with the fact that most papers adopt Mexihat or Morlet as mother wavelet functions to construct wavelet neural network. So, we can use wavelet functions Morlet or Mexihat as mother wavelet function to create MWFWNN network. In this paper, we use the Morlet wavelet function as the basic mother wavelet function to construct MWFWNN.

4.2.2. Comparing with WNN and PSO_SVM, BP. Similar to MWFWNN, WNN, BP network, and support vector machine (SVM) can be used to solve the target assessment. As the choice of the support vector machine parameters c and g has no uniform standard except relying on experience; in this paper, we use particle swarm optimization (PSO) algorithm to optimize the SVM parameters c and g . Next, we use wavelet neural network, BP neural network, and PSO_SVM to solve threat assessment, and the results are compared with MWFWNN.

The structure of wavelet neural network and BP neural network is 6-12-1 according to the characteristics of the data

TABLE 3: Predicting results of BP, PSO_SVM, MFWFNN, and WNN.

Neural network	BP	PSO_SVM	MFWFNN	WNN
MSE	9.39×10^{-3}	4.80×10^{-3}	1.23×10^{-3}	4.01×10^{-3}
Running time (s)	1.86	10.6	4.88	5.10

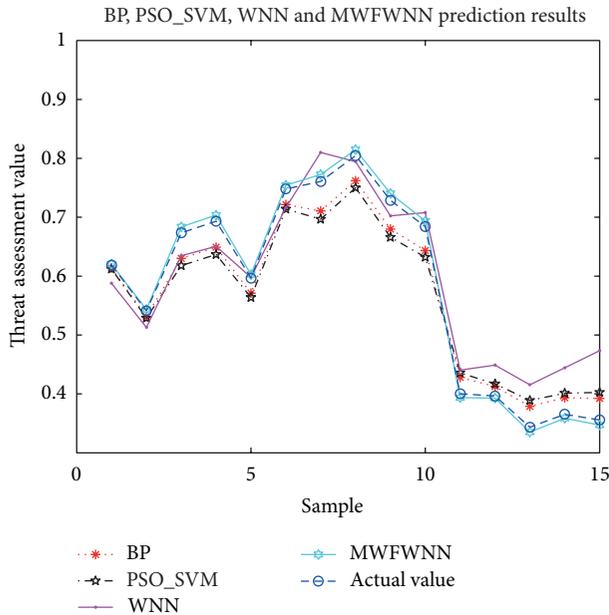


FIGURE 3: Result of target threat assessment based on WNN, BP, MFWFNN and PSO_SVM.

used. Where the WNN and other parameters are setting as shown in Section 4.2.1. PSO SVM uses LIBSVM toolbox [11] whose default is C-SVR and RBF kernel function, where $C = 1$, $\gamma = 1/n$. By default, PSO local search $c_1 = 1.5$, the global search $c_2 = 1.7$, the maximum evolution times $\text{maxgen} = 200$, the maximum population size $\text{sizepop} = 20$, the rate $k = 0.6$ between V and X ($k \in [0.1, 1.0]$), $V = kX$, SVM parameter $c \in [0.1, 100]$, and the parameter $g \in [0.01, 1000]$.

Threat assessment is predicted by the trained WNN, BP network, and PSO_SVM and their MSE are 4.01×10^{-3} , 9.39×10^{-3} and 4.80×10^{-3} , all of which are greater than MFWFNN (1.23×10^{-3}) (as shown in Table 3). For the running time, due to BP neural network calling MATLAB embedded functions, running time is least; wavelet neural networks need to call each mother wavelet function and its derivative, so it is time to consume more; MFWFNN network needs to find the optimal wavelet function from the library, so the running time is more than WNN; the PSO_SVM call PSO algorithm to optimize parameter c and g , but PSO optimization algorithm is more complex, so it implements most slowly.

Prediction error of BP, PSO_SVM, WNN, and MFWFNN is shown in Figure 3. The figure shows that, prediction errors of WNN, BP, and PSO_SVM are similar. The error is less than the true value at sample 1–10, while the error was significantly greater than the true value at sample 11–15. MFWFNN network error has similar trend, but the

error was significantly less than the WNN, BP network and PSO_SVM, prediction value closer to the expectations.

5. Conclusion

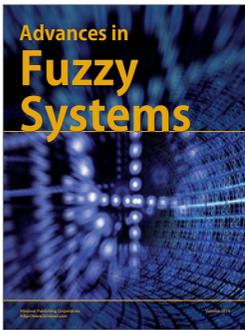
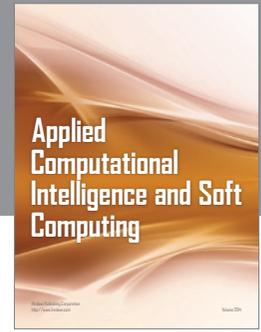
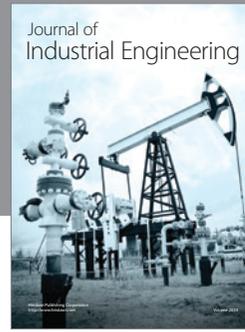
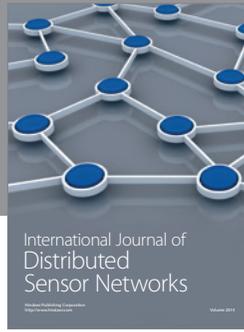
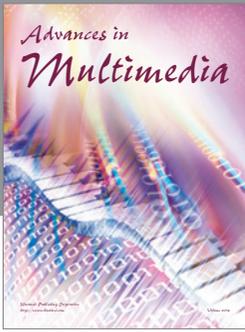
Based on requirements for quickly processing information in the modern information war, aiming to the characteristics of threat assessment in data fusion functional model, we adopt MFWFNN to solve the threat assessment under the comprehensive consideration of various factors which influence the threat degree. After constructing wavelet function library with 7 wavelet functions, we get the best performance wavelet function Morlet as mother wavelet function to construct MFWFNN network and its result is compared with the WNN, BP, and PSO_SVM networks. Simulation results show that, the MSE of MFWFNN is 1.23×10^{-3} , which is far better than WNN (4.01×10^{-3}), BP network (9.39×10^{-3}), and PSO_SVM (4.80×10^{-3}), achieving the desired goal. In our future work, we will further expand the scale of wavelet function library to find more suitable wavelet function to solve the threat assessment and other problems.

References

- [1] F. Johansson and G. Falkman, "A Bayesian network approach to threat evaluation with application to an air defense scenario," in *Proceedings of the 11th International Conference on Information Fusion (FUSION '08)*, pp. 1–7, July 2008.
- [2] J. Yang, W. Y. Gao, and J. Liu, "Threat assessment method based on Bayesian network," *Journal of PLA University of Science and Technology*, vol. 11, no. 1, pp. 43–48, 2010.
- [3] J. Li and L. Han, "Threat ordering of interference targets based on uncertain multi-attribute decision-making," *Telecommunication Engineering*, vol. 49, no. 10, pp. 62–64, 2009.
- [4] G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, "Target threat assessment using glowworm swarm optimization and BP neural network," *Journal of Jilin University*. In press.
- [5] G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, "The model and algorithm for the target threat assessment based on Elman-AdaBoost strong predictor," *Acta Electronica Sinica*, vol. 40, no. 5, pp. 901–906, 2012.
- [6] X. D. Gu, Z. X. Tong, S. J. Chai, X. W. Yuan, and Y. L. Lu, "Target threat assessment based on TOPSIS combined by IAHP and the maximal deviation," *Journal of Air Force Engineering University*, vol. 12, no. 2, pp. 27–31, 2011.
- [7] K. Sycara, R. Ginton, B. Yu et al., "An integrated approach to high-level information fusion," *Information Fusion*, vol. 10, no. 1, pp. 25–50, 2009.
- [8] A. Kott, R. Singh, W. M. McEneaney, and W. Milks, "Hypothesis-driven information fusion in adversarial, deceptive environments," *Information Fusion*, vol. 12, no. 2, pp. 131–144, 2011.
- [9] A. Grossmann and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape," *SIAM*

Journal on Mathematical Analysis, vol. 15, no. 4, pp. 723–736, 1984.

- [10] Q. Zhang and A. Benveniste, “Wavelet networks,” *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 889–898, 1992.
- [11] C. C. Chang and C. J. Lin, “LIBSVM: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

