Design and application of a wavelet neural network program for evaluation of goodwill value in corporate intellectual capital

Bei Yuan[1,*], Fusheng Wang[1], Di Bao[2]

1. School of Management, Harbin Institute of Technology, Harbin 150001, China

2. School of Economic and Management, Hunan University of Science and Engineering, Yongzhou 425199, China

yuanbei1989@163.com

ABSTRACT. *Considering the poor applicability of existing wavelet neural network (WNN) methods in the termination decision of R&D projects, this paper applies the WNN in the evaluation of the value of goodwill in corporate intellectual capital (CIC) through computer programming. Specifically, the author designed a WNN program that combines the merits of both neural network and wavelet analysis. Then, evaluation of CIC goodwill value was elaborated in details from the aspects of wavelet transform and multi-resolution analysis, the learning algorithm and training process, as well as the non-stationary time series analysis and prediction. The comparison of the predicted curve and the original series curve shows that the WNN-based program outperformed the traditional analytical methods in the accuracy of CIC goodwill value evaluation. The research findings shed new light on the application of the WNN program in the setting of the index system for goodwill value evaluation.*

RÉSUMÉ. *Compte tenu de la faible applicabilité des méthodes de réseau de neurones en ondelettes (WNN, le sigle de « wavelet neural network » en anglais) existantes dans la décision de clôture aux projets de R&D, cet article applique le WNN à l'évaluation de la valeur de l'écart d'acquisition dans le capital intellectuel des entreprises (CIC, le sigle de « corporate intellectual capital » en anglais) par le biais de la programmation informatique. En particulier, l'auteur a conçu un programme de WNN qui combine les avantages de l'analyse par réseau de neurones et par ondelettes. Ensuite, l'évaluation de la valeur des écarts d'acquisition de CIC a été élaborée en détail à partir des aspects de la transformation en ondelettes et de l'analyse multi-résolution, de l'algorithme d'apprentissage et du processus de formation, ainsi que de l'analyse et de la prédiction de séries chronologiques non stationnaires. La comparaison de la courbe prédite et de la courbe de la série d'origine montre que le programme basé sur le réseau de neurones en ondelettes a surpassé les méthodes analytiques traditionnelles en termes de précision de l'évaluation de la valeur des écarts d'acquisition de CIC. Les résultats de la recherche ont jeté un nouvel éclairage sur l'application du programme de WNN dans l'établissement du système d'indexation pour l'évaluation de la valeur des écarts d'acquisition.*

KEYWORDS: *wavelet neural network (WNN), corporate intellectual capital (CIC), goodwill value.*

MOTS-CLÉS: réseau de neurones à ondelettes (WNN), capital intellectuel des entreprises (CIC), valeur des écarts d'acquisition.

DOI:10.3166/ISI.23.5.185-200 © 2018 Lavoisier

1. Introduction

With the development of science and technology, more and more attention has been paid to corporate intellectual capital (CIC). One of the most important aspects of the CIC is goodwill, which refers to the favorable business conditions like superior geographical location, famous trade name, superb process technology and good reputation. The value of goodwill is the total amount of economic resources attached to an enterprise under the combined effect of the various factors affecting the goodwill of the enterprise. The existing evaluation methods for the value of goodwill mainly focus on qualitative analysis, failing to yield comprehensive results. For example, when the value of goodwill is evaluated by the cost method, one of the three internationally accepted asset valuation approaches along with the market method and income method, the analyst needs to estimate the current cost incurred to recreate the components in the intangible asset of the target goodwill.

The wavelet neural network (WNN) provides a possible solution to the said problem. Inspired by biological neural networks, the WNN is a distributed and parallel information processing model. By adjusting the relationship between numerous internal nodes, this network system can complete complex logical operations and realize highly nonlinear relationships. As an artificial neural network, the WNN provides a non-programmed algorithm model capable of processing information adaptively. Since it was proposed by Prof. Benvenist in 1992, the WNN has developed rapidly into a popular mathematical modeling method. As a feedforward neural network, the WNN essentially integrates such two models as wavelet transform and neural network via wavelet decomposition by replacing neurons with wavelets. The merits of the two models are combined in the WNN, thanks to the introduction of two new parameters that enable efficient function approximation, pattern recognition and fault tolerance (Linhares *et al.*, 2015; Mikhailov, 2003; 2004; Paugam, 2011; Pih, 2014; Porpora, 2016).

In 1994, Kreinovich V. *et al.* proved the WNN as the best approximator for univariate functions. The proof further accelerates the development of the WNN. Pati and Krisnaprasad explored the relationship between wavelet analysis and neural network, and created a wavelet network model called discrete affine, which essentially includes discrete wavelet transform into the neural network. Szu *et al.* designed two adaptive wavelet networks: one of the networks, focusing on function approximation, mainly represents signals; the other network, focusing on the feature extraction of suitable wavelets, essentially looks for a set of best wavelet bases from the wavelet feature space. Baskshi and Stephanopolous chose the orthogonal wavelet function as the activation function of neurons, and then proposed an orthogonal multi-resolution analysis method based on the WNN. Based on the multi-resolution analysis theory, their WNN includes both scale function and wavelet function, and was trained by step-by-step learning (Carvalho *et al.*, 2016; Casta *et al.*, 2010; Chauhan *et al.*, 2009;

Christopher *et al.*, 2016).

Despite its good local time-frequency features, the wavelet theory has only been applied in small scale, owing to the extremely high cost in the construction and storage of wavelet basis functions for the large-scale application of wavelet analysis. Fortunately, the neural network, good at handling large-scale problems, can make up for this defect. Thus, the two techniques can be combined into a WNN. When the WNN is adopted for the termination decision of R&D projects, the most difficult step lies in the setting of an appropriate index system, including selecting the future state of the target project and quantifying its feature factors. The success of WNN application hinges on the selection of the correct indices. This calls for more attention to the construction of the index system. The existing studies on index system construction mostly stops at the discriminant methods and aims to verify the discriminant models or methods. There is little report on the neural network models or neural network applications. What is worse, the previous research concentrates on statistical analysis, which cannot be directly adopted to solve practical problems. As a result, this paper attempts to apply the WNN program into practical problems, and combine the program calculation with the evaluation of goodwill value in the CIC (hereinafter referred to as the CIC goodwill value) (Schenten *et al.*, 2016; Shoaib *et al.*, 2016; Wen & Stephen, 2016).

Based on the WNN method, the numerical information (0 and 1) of the evaluation object was taken as the input vector of the neural network, while the mean goodwill value estimated by experts was considered as the output. During the training, the wavelet coefficients and network weights were adjusted adaptively, turning the qualitative indices of the evaluation into quantitative indices. In this way, the value of the goodwill can be evaluated in a comprehensive manner. This paper provides a new strategy to overcome the limitation of the previous research of the WNN: applying the WNN program in the setting of the index system for goodwill value evaluation. Compared with the traditional analytical methods, the proposed evaluation method is accurate and practical in the evaluation of CIC goodwill value (Wójcik-Jurkiewicz, 2009; Yilmaz & Oysal, 2010; Zanoni & Vernizzi, 2014).

The remainder of this paper is organized as follows: Section 2 designs and analyzes the WNN evaluation program; Section 3 discusses and forecasts the non-stationary time series for the evaluation of CIC goodwill value, thereby verifying the feasibility of the WNN-based evaluation of CIC goodwill value; Section 4 wraps up this paper with several conclusions.

2. WNN-based evaluation model for CIC goodwill value

2.1. Wavelet transform and multi-resolution analysis

(1) Wavelet transform

The CIC goodwill value network consists of such three layers as the input layer, the hidden layer and the output layer. Here, the influencing factors of CIC goodwill

value are taken as the inputs, the mean goodwill value estimated by experts is adopted as the output, and the Morlet wavelet is selected as the excitation function for hidden layer neurons:

$$h(t) = \cos(1.75t)\exp(-\frac{t^2}{2}) \tag{1}$$

The hidden layer neurons were divided into wavelet base neurons (ψ) and scale function neurons (ϕ). Here, the mother wavelet function $\psi(x)$ in the function space $L^2(R)$ is assumed to satisfy the following conditions:

$$C_\psi = \int_R \frac{|\widehat{\psi}(w)|^2}{|w|} dw < \infty \tag{2}$$

where $\hat{\psi}(w)$ is the Fourier transform of $\psi(x)$. Let a and b be the expansion and translation coefficients of $\psi(x)$, respectively. Then, the input layer function, which is a basic wavelet function, it can be expressed as $\psi_{a,b}(x)$. What's more, the output layer function can be expressed as y_k:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi(\frac{x-b}{a}) \quad (a,b \in R^2) \tag{3}$$

$$y_k = \sum_{j=1}^{n} r_j h\left(\frac{\sum_{i=0}^{m} w_{ij} x_k(i) - b_j}{a_j}\right)$$

For any $f(x) \in L^2(R)$, i.e. any influencing factor $X(x_1, x_2, x_3...x_p)$ of the CIC goodwill value, let p be the number of input samples. Then, the wavelet transform can be defined as:

$$W_f(a,b) = \frac{1}{\sqrt{C_\psi}} \int_R f(x)\psi_{a,b} dx \tag{4}$$

The inversion formula of equation (4) can be expressed as:

$$f(x) = \frac{1}{C_\psi} \int_0^{+\infty} W_f(a,b)\psi_{a,b} da db \tag{5}$$

(2) Multi-resolution analysis

After the wavelet transform, the input layer, hidden layer and output layer functions of CIC goodwill value evaluation were subjected to discretization, i.e. multi-layer multi-resolution analysis (Davanipoor *et al.*, 2012). Specifically, each function F was described as a limit of a set of approximation functions in the function space. In this set, the approximation functions, each being a smooth approximation of function F, are ranked in ascending order of fineness. Since the approximations are obtained at different levels of resolution, the analysis process is called a multi-resolution analysis. This analysis approach provides the basic idea of seeking wavelet filter: the bases of wavelet could be constructed from the subspaces, and then expanded by simple transform into the entire function space. The initial purpose of wavelet analysis is to construct the orthogonal basis of $L^2(R)$. The said bases can be obtained by translation and expansion of the same function.

Let $a=2^m$ and $b=k2^m$ according to the binary system. Then, the wavelet transform formula can be converted into:

$$\psi_{m,k}(x)=\frac{1}{\sqrt{2^m}}\psi(2^{-m}x-k) \qquad (m,k\in Z^2) \tag{6}$$

Then, the continuous integrable function $f(x)\in L^2(R)$ approximates the resolution 2^m for multi-resolution analysis:

$$f_m(x)=\sum_{k=-\infty}^{+\infty}a_{mk}\phi_{mk}(x) \tag{7}$$

Using the scaling and translation coefficients, $\phi_{mk}(x)$ can be constructed from the scale function $\phi(x)$:

$$\phi_{mk}(x)=\frac{1}{\sqrt{2^m}}\phi(2^{-m}x-k) \qquad (m,k\in Z^2) \tag{8}$$

Let d_k be the expected output of CIC goodwill value k, y_k be the evaluation output of CIC goodwill value, and P be the number of training samples. Thus, the network coefficients can be optimized by the MSE function:

Therefore, the coefficients of the network can be optimized by minimizing the error function:

$$MSE=\frac{1}{2}\sum_{k=1}^{P}(y_k-d_k)^2 \tag{9}$$

Once the network parameters enter the learning process, the evaluated CIC goodwill value can be calculated by the weight coefficients of the influencing factors.

2.2. Learning algorithm and training process

(1) Learning algorithm

To speed up the convergence, the parameters of the evaluation model for CIC goodwill value were subjected to learning. Firstly, the functions were optimized by the Levenberg–Marquardt (LM) algorithm, which is expressed in the matrix form as $z_k=[w_{ik},r_k,a_k,b_k]$, with k being the number of iterations. Meanwhile, the output layer function of the evaluation model is defined as $F(z)$.

$$z_{k+1} = z_k - A_k^{-1} g_k \tag{10}$$

where $A_k=\nabla^2 F(z)|_{Z=Zk}$ a Hessian matrix; $g_k=\nabla F(z)|_{Z=Zk}$ is the gradient of the output layer function of the evaluation model. Let $F(z)=v(z)^T v(z)$, with $v(z)$ being the error vector. Then, the gradient of the output layer function can be expressed as:

$$\nabla F(z) = 2J^T(z)v(z) \tag{11}$$

The Jacobian matrix $J(z)$ can be expressed as:

$$J(z) = \begin{bmatrix} \frac{\partial v_1(z)}{\partial z_1} & \frac{\partial v_1(z)}{\partial z_2} & \cdots & \frac{\partial v_1(z)}{\partial z_n} \\ \frac{\partial v_2(z)}{\partial z_1} & \frac{\partial v_2(z)}{\partial z_2} & \cdots & \frac{\partial v_2(z)}{\partial z_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial v_N(z)}{\partial z_1} & \frac{\partial v_N(z)}{\partial z_2} & \cdots & \frac{\partial v_N(z)}{\partial z_n} \end{bmatrix}$$

The approximation function of the Hessian matrix is:

$$\nabla^2 F(z) \approx 2J^T(z)J(z) \tag{12}$$

Combining equations (10) and (11), we have:

$$z_{k+1} = z_k - \left[J(z)^T J(z) \right]^{-1} J(z)^T v(z) \tag{13}$$

After being converted to an identity matrix, the gradient of the output layer function in the evaluation model can be expressed as:

$$z_{k+1} = z_k - \left[J(z)^T J(z) + \mu_k I \right]^{-1} J(z)^T v(z) \tag{14}$$

When μ_k=0, the above algorithm is a simplified Newton algorithm; with the growth in the value of μ_k, it can be viewed as a small step gradient algorithm (Dhibi *et al.*, 2016). If the iterative training is successful, the learning can be accelerated by reducing the value of μ_k; otherwise, it can be slowed down by increasing that value.

The LM learning algorithm is essentially a quasi-Newton algorithm (Efendi *et al.*, 2016), whose convergence speed lies between the Newton algorithm and the gradient reduction method. The algorithm can be implemented in the following steps:

Step 1) Randomly initialize the weight vector w, set the target error ε, and let k=1.

Step 2) Calculate the Jacobian matrix J for the weight vector W according to the n-variate equation set in equation (1).

Step 3) Search according to the following formula:

$$w^{(k+1)}=w^{(k)}-\eta_k(J^T(w^{(k)})J(w^{(k)})+\mu_k I)J^T(w^{(k)})F(w^{(k)}) \quad (15)$$

Step 4) If E(W(k))≤ε, terminate the algorithm and obtain the weight vector that satisfies the accuracy requirement; otherwise, go to Step 5)

Step 5) If

$$|F(w^{(k+1)})-F(w^{(k)})|\leq 0, \text{ let } \mu_k=\mu_k/4 \quad (16)$$

Otherwise, let $\mu_k=\mu_k\times 4$, and go to Step 3).

The program code of the LM learning algorithm is shown in Figure 1 below.

Algorithm 1 Levenberg-Marquardt (LM) algorithm

```
initialize k = 0, ν = 2, z = z_0;
A = J(z)^T J(z), g = J(z)^T v(z);
found = (||g||_∞ ≤ ε_1), μ = τ * max{a_ij};
while not found and k < k_max do
    k = k + 1;
    solve (A + μI)h_lm = −g;
    if ||h_lm|| ≤ ε_2(||z||) + ε_2 then
        found = true
    else
        z_new = z + h_lm
        ρ = (F(z) − F(z_new))/(L(0) − L(h_lm))
        if ρ > 0 then
            z = z_new
            A = J(z)^T J(z), g = J(z)^T v(z)
            found = (||g||_∞ ≤ ε_1)
            μ = μ * max{1/3, 1 − (2ρ − 1)^3}, τ = 2
        else
            μ = μ * τ, τ = 2 * τ
        end if
    end if
end while
```

Figure 1. Source code of the LM algorithm

The adoption of the LM algorithm in learning ensured a good convergence speed, but did not solve the local convergence problem. Therefore, the global optimization algorithm genetic algorithm (GA) was introduced to determine the initial state of the neural network (Ergur & Oysal, 2015), so as to minimize the fitting error in convergence. The flow chart and program code of the GA are presented in Figures 2 and 3, respectively.

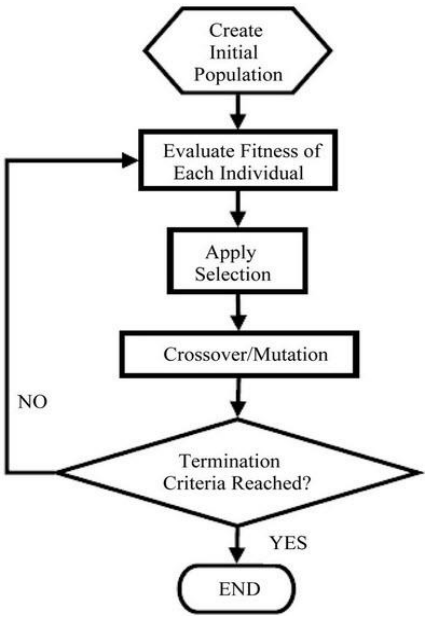

Figure 2. The flow chart of the GA

Algorithm 2 Genetic Algorithm (GA)

Input: population P;
Output: best fit individuals.

initialize $P(t=0)$;
evaluate $P(t=0)$;
while termination condition is not met **do**
  $P_p(t) = P(t).selectParents()$;
  $P_c(t) = reproduction(P_p)$;
  $mutate(P_c(t))$;
  $P(t+1) = buildNextGenerationFrom(P_c(t), P(t))$;
  $t = t+1$;
end while
return best fit individuals

Figure 3. Source code of the GA

(2) Training process

The GA and the LM were combined to train the evaluation model for CIC goodwill value. Firstly, the initial state of the CIC goodwill value network was determined by the GA; then, the fitting error was minimized by the LM, fulfilling the purpose of optimization. The program code of the GA-LM is displayed in Figure 4 below.

Algorithm 3 GA-LM WNN algorithm

initialize parameter matrices $\mathbf{w}, \mathbf{r}, \mathbf{a}, \mathbf{b}$ using Algorithm 2;
input $\{x_k(i), d_k\}^P$ to compute output $y_k(i)$ and error e;
while $e \geq MSE$ **do**
  calculate matrix z_k by $\mathbf{w}, \mathbf{r}, \mathbf{a}, \mathbf{b}$;
  compute matrix $\mathbf{J}$;
  $\Delta z = -[\mathbf{J}^T(z)\mathbf{J} + \mu_k \mathbf{I}]^{-1}\mathbf{J}^T(z)v(z)$;
  $z_k = z_k + \Delta z$;
  calculate error e';
  if $e' < e$ **then**
    $\mu_{k+1} = \mu_k / \alpha$
    update parameters a, b, w_{ij}, r_j and $e = e'$
  else
    $\mu_{k+1} = \mu_k * \alpha$
  end if
end while

Figure 4. Program code of the GA-LM training program

In addition, another training algorithm, the particle swarm optimization (PSO) was adopted to further train the evaluation model. Compared with the GA-LM, the PSO is easy to implement and does not need to adjust too many parameters. As an iteration-based optimization algorithm, the PSO initializes the system as a set of random solutions and iteratively looks for the optimal solution, which is coded in real numbers.

The PSO training of the evaluation model is summed up as follows:

Let N be the number of input samples in a D-dimensional space, $X_i=(x_{i1},x_{i2},\ldots, x_{id},\ldots,x_{iD},)$, $V_i=(v_{i1},v_{i2},\ldots,v_{id},\ldots,v_{iD},)$ and $P_i=(p_{i1},p_{i2},\ldots,p_{id},\ldots,p_{iD},)$ be the position, the velocity and the ideal position of the i-th sample, respectively, and P_g be the optimal position of all input samples. Then, the velocity of the i-th sample in the next iteration can be calculated as:

$$v_{id}(t+1) = W \times v_{id}(t) + c_1 \times rand() \times (p_{id}(t) - x_{id}(t)) + c_2 \times rand() \times (p_{gd}(t) - x_{id}(t)) \quad (17)$$

where c_1 and c_2 are acceleration coefficients; *rand*() is a random variable in the range [0, 1]; W is the inertia factor. A large W is suitable for the search in a large solution space, and the inverse is also true. In equation (17), the inertia factor is made up of three parts: the product between the velocity and inertia factor in the previous iteration,

the unique feature of the input sample, and the difference between the sample and the swarm.

Let w_{ij} be the weight from the input layer to the hidden layer, w_j be the weight from the hidden layer to the output layer, and a_j and b_j be the scaling and translation coefficients, respectively. Then, the position vector of the i-th input sample can be defined as:

$$present(i) = [w_1, w_2, ..., w_k, a_1, a_2, ..., a_k, b_1, b_2, ..., b_k], i = 1,2,...,n \tag{18}$$

where k is the number of neurons in the hidden layer.

The expected output of the sample can be denoted as y_l(l=1,2,…,L). Note that L is the number of input samples. Taking the mother wavelet function as the basis function, we have:

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{19}$$

where d is the number of iterations. The position and velocity of each input sample can be determined by equations (17) and (18). Then, the optimal position p_g^d of all input samples can be derived from the optimal position p_i^d of each input sample. The minimum mean square error (MMSE) is the difference between the actual output $\hat{y}$ and the expected output y:

$$MMSE = \frac{1}{2}\sum_{i=1}^{L}(y_l - \hat{y}_l)^2 \tag{20}$$

The parameters of the PSO are set as follows.

The number of input samples is generally set to 20~40, or 100~200 if the problem is extremely hard or falls into a specific category. Here, 31 input samples are selected because there are 31 factors affecting the CIC goodwill value.

Next, the convergence of the PSO was further improved by controlling the relationship between inertia weight W and the number of iterations as a linearly negative correlation. The inertia weight at the i-th iteration can be expressed as:

$$W_i = W_{max} — (W_{max} — W_{min}) / Enter\_max \times i \tag{21}$$

where W_{max} is the maximum weighting efficient; W_{min} is the minimum weighting coefficient. Note that $W_{min} \leq W \leq W_{max}$.

The program code of the PSO is shown in Figure 5.

Algorithm 1 WNN-PSO algorithm

repeat
  input training instances and expected outputs
  initialize particles and their positions, speed
  for each particle i **do**
    for each dimension d **do**
      $v_{id}(t+1) = W \times v_{id}(t) + c_1 \times rand() \times (p_i d(t) - x_{id}(t)) + c_2 \times rand() \times (p_g d(t) - x_{id}(t))$
      $x_{id}(t+1) = x_{id}(t) + v_{id}(t+1)$
    end for
    $p_i = x_i$
    calculate $\hat{y}$ based on Equation (2)
    calculate $MMSE$ based on Equation (6)
    if $MMSE_i < MMSE_g$ **then**
      $p_g = p_i$
    end if
  end for
  $t = t + 1$
until $t > MAX\_INTERATIONS$

Figure 5. Program code of the PSO

3. Analysis and prediction of non-stationary time series

This section analyzes and predicts the non-stationary time series based on wavelet transform through the following steps. First, the original time series was decomposed into different layers by wavelet decomposition, separating the trend items, periodic items and random items. Then, each layer was analyzed and predicted separately. Finally, the results of different layers were synthetized into the predicted value of the original time series. In this way, it is possible to determine the overall trend of the CIC goodwill value, identify the existing problems and propose countermeasures.

For the wavelet-based analysis and prediction, the decomposition and synthesis can be implemented through the binary wavelet transform and its inverse transform or multi-resolution analysis. In the former approach, the two series obtained through decomposition are of the same length as the original time series, and enjoy good smoothness. In the latter approach, each of the two series is half of the length of the original time series. Hence, the latter approach is suitable for processing long time series.

The wavelet for time series analysis and prediction should be selected depending on the specific problem. Currently, many wavelets are available, such as orthogonal wavelet, spline wavelet and double orthogonal positive wavelet, to name but a few. The orthogonal wavelet can eliminate the correlation between the decomposed layers, making them independent from each other. In this paper, the binary wavelet is selected for decomposition and synthesis and the quadratic spline function is adopted as the wavelet function.

The non-stationary time series was predicted as follows. The wavelet function has a certain smoothing effect. After 10,000 wavelet transforms, the random and periodic terms in the original time series were gradually separated. In other words, the abnormal points or random factors in the series were gradually eliminated, and constantly corrected through the transform, such that the time series became increasingly smooth. The transforms continued until the series only contains long-term trend terms. The number of decompositions were determined by plotting a point map. The trend terms were predicted by fitting the curve with the least squares method (Giovanni & Francesco, 2010).

Besides long-term trend terms, the other layers in wavelet decomposition contain periodic and random terms. The random terms are small-scale components. In most cases, the high frequency of the first wavelet decomposition contains the most random components, while the other layers contain more periodic items. For each layer of a random term, the components at different scales are generally independent and stationary, and can be processed separately. The different layers of a periodic term were discussed separately by periodic graph modeling.

Before the time series prediction and analysis, the time series was denoted as S_0 and the parameter λ was introduced as a known constant. The original time series is shown in Figure 6 below.

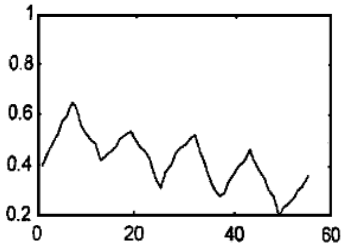

Figure 6. The original time series

The filtering (Linhares *et al.*, 2015) was carried out in the following steps. First is the wavelet decomposition. The original time series S_0 was subjected to three wavelet decompositions according to the iteration formula:

$$\begin{cases} W_{j+1} = \frac{1}{K} S_j * G_j \\ S_{j+1} = S_j * H_j \end{cases} \quad (22)$$

Thus, the binary wavelet transform series of each layer can be obtained as W_j (j=1, 2, ..., J), with J being the number of wavelet decompositions, and the series of the residual low-frequency components can be derived as S_k. The decomposition results are shown in Figure 7.

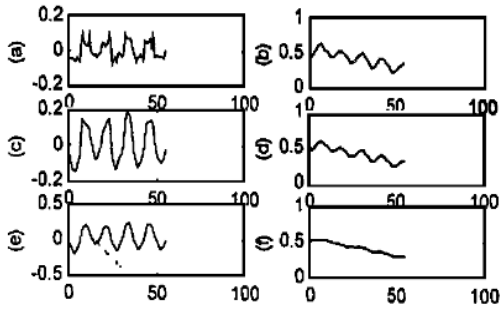

Figure 7. The decomposition results (left: high frequency: right: low frequency)

Next, the trend terms were determined. The S_k was plotted to observe if the trend terms are obvious and can be approximated with a simple model. If the trend is not obvious, the value of J (i.e. the number of decompositions) should be increased. After observing an obvious trend, the S_k was modelled to find the fitted value and the predicted value S_J. The high frequency obtained in the first decomposition was further decomposed and the results are presented in Figure 8.

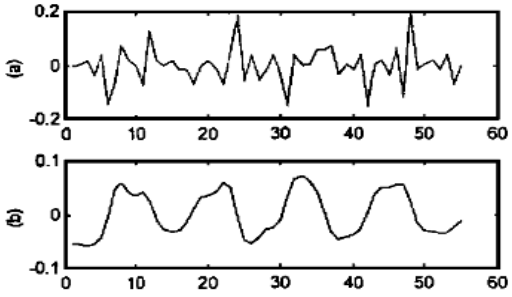

Figure 8. The results of secondary decomposition (left: high frequency: right: low frequency)

After that, the binary wavelet transform series W_j of each layer was fitted and predicted. Each W_j was plotted to observe if it is periodic and easy to simulate. Those with high frequency and small amplitude were neglected. Those suitable for simulation were modelled to find the fitted and predicted results. The series were thus processed layer by layer. The above processes were repeated depending on the size of the random terms, and terminated after the synthesis of the results of different layers. Finally, the fitted value and predicted value of the original time series were synthesized according to the iteration formula. Figure 9 compares the predicted curve and the original series curve.

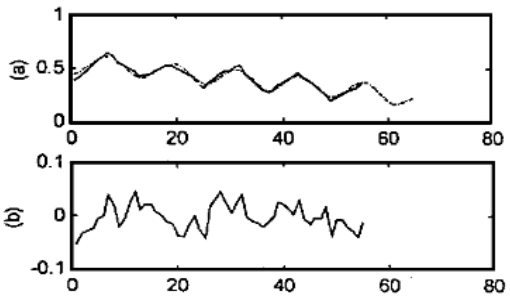

Figure 9. Comparison of the predicted curve and the original series curve

It can be seen that the predicted curve fitted well with the original series curve. The non-stationary time series is very difficult to predict because of the complex intertwining of various factors. Thanks to wavelet analysis, the time series was decomposed into different layers according to the scale, making the problem easy to analyze and predict. In particular, the trend and periodic terms were separated for independent analysis and prediction. In this way, the important information can be decomposed in a more refined manner than general problems.

4. Conclusions

Based on the existing research results, this paper applies the WNN in the evaluation of CIC goodwill value through computer programming. The designed WNN program combines the merits of both neural network and wavelet analysis. The evaluation of CIC goodwill value was elaborated in details from the aspects of wavelet transform and multi-resolution analysis, the learning algorithm and training process, as well as the non-stationary time series analysis and prediction. The research results show that the WNN-based program outperformed the traditional analytical methods in the accuracy of CIC goodwill value evaluation.

Acknowledgement

This paper is made possible thanks to the generous support from the General Project of National Natural Science Foundation of China (Grant No.: 71532004; 71672046).

References

Carvalho C., Rodrigues A., Ferreira C. (2016). The recognition of goodwill and other intangible assets in business combinations–the portuguese case. *Australian Accounting Review*, Vol. 26, No. 1, pp. 4-20. http://doi.org/10.1111/auar.12073

Casta J., Paugam L., Stolowy H. (2010). An explanation of the nature of internally generated goodwill based on aggregation of interacting assets. *Economics Papers from University Paris Dauphine*, Vol. 12, pp. 46-53. http://doi.org/10.2139/ssrn.1716907

Chauhan N., Ravi V., Chandra D. K. (2009). Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks. *Expert Systems with Applications*, Vol. 36, No. 4, pp. 7659-7665. http://doi.org/10.1016/j.eswa.2008.09.019

Christopher X., Avleen B., Juan L. F. (2016). An online prediction framework for non-stationary time series. *Statistics*, No. 11, pp. 55-57+65.

Davanipoor M., Zekri M., Sheikholeslam F. (2012). Fuzzy wavelet neural network with an accelerated hybrid learning algorithm. *IEEE Transactions on Fuzzy Systems*, Vol. 20, No. 3, pp. 463-470. http://doi.org/10.1109/tfuzz.2011.2175932

Dhibi N., Elkefi A., Bellil W., Amar C. B. (2016). Multi-layer compression algorithm for 3D deformed mesh based on multi library wwavelet neural network architecture. *Multimedia Tools & Applications*, No. 3, pp. 1-19. http://doi.org/10.1007/s11042-016-3996-8

Efendi R., Deris M. M., Ismail Z. (2016). Implementation of fuzzy time series in forecasting of the non-stationary data. *International Journal of Computational Intelligence and Applications*, Vol. 15, No. 2, pp. 105-108+121. http://doi.org/10.1142/S1469026816500097

Ergur H. S., Oysal Y. (2015). Estimation of cutting speed in abrasive water jet using an adaptive wavelet neural network. *Journal of Intelligent Manufacturing*, Vol. 26, No. 2, pp. 1-11. http://doi.org/10.1007/s10845-013-0798-y

Giovanni L., Francesco M. (2010). Goodwill write-off and financial market behaviour: an analysis of possible relationships. *Advances in Incorporating Advances in International Accounting*, Vol. 9, No. 5, pp. 333-339. http://doi.org/10.1016/j.adiac.2010.08.001

Linhares S., Fonte A., Martins M., Araújo F., Silveira L. (2015). Fuzzy wavelet neural network using a corr-entropy criterion for nonlinear system identification. *Mathematical Problems in Engineering*, Vol. 2015, No. 5, pp. 1-12. http://dx.doi.org/10.1155/2015/678965

Mikhailov L. (2003). Deriving priorities from fuzzy pairwise comparison judgements. *Fuzzy Sets and Systems*, No. 6, pp. 365-385. http://doi.org/10.1016/s0165-0114(02)00383-4

Mikhailov L. (2004). Group prioritization in the AHP by fuzzy preference programming method. *Computers and Operations Research*, pp. 293-301. http://doi.org/10.1016/s0305-0548(03)00012-1

Paugam L. (2011). Valuation and reporting of goodwill: Theoretical and empirical issues. *Dissertations & Theses–Gradworks*, Vol. 1, No. 5, pp. 58-58.

Pih L. (2014). When organizations in the cultural industries seek new business models: A case study of the French online press. *International Journal of Arts Management*, Vol. 16, No. 3, pp. 147-54. http://doi.org/10.3109/00365597409132120

Porpora D. V. (2016). The meaning of culture and the culture of empiricism in American sociology. *American Sociologist*, Vol. 47, pp. 1-12. http://doi.org/10.1007/s12108-016-9316-y

Schenten D., Kracker S. G., Franco S., Klein U., Murphy M. (2016). Location patterns and location factors in cultural and creative industries. *Quaestiones Geographicae*, Vol. 34, No. 2, pp. 7-27. http://doi.org/10.1515/quageo-2015-0011

Shoaib M., Shamseldin Y., Melville W., Khan M. (2016). Hybrid wavelet neural network approach. *Springer International Publishing*, No. 2, pp. 127-143. http://doi.org/10.1007/978-3-319-28495-8_7

Wen H., Stephen R. (2016). Moehrle accounting for goodwill: An academic literature review and analysis to inform the debate. *Research in Accounting Regulation*, Vol. 28, No. 1, pp. 11-21.

Wójcik-Jurkiewicz M. (2009). Enterprise value and goodwill. *Theoretical Journal of Accounting*, Vol. 53, No. 109, pp. 303-320.

Yilmaz S., Oysal Y. (2010). Fuzzy wavelet neural network models for prediction and identification of dynamical systems. *IEEE Transactions on Neural Networks*, Vol. 21, No. 10, pp. 1599-1609. http://doi.org/10.1109/TNN.2010.2066285

Zanoni A., Vernizzi S. (2014). Goodwill reduction: The competitive analysis of enterprise value. *International Business Research*, Vol. 11, pp. 86-92. http://doi.org/10.5539/ibr.v7n11p35