# PERFORMANCE COMPARISON OF ARTIFICIAL INTELLIGENCE METHODS FOR PREDICTING CASH FLOW

*Liu Hongjiu*[*,†], *Robert Rieg*[*], *Hu Yanrong*[†]

**Abstract:** Cash flow forecasting is indispensable for managers, investors and banks. However, which method is more robust has been argued under the condition of small size samples. With sliding window technique we create the Response Surface, Back Propagation Neural Network, Radial Basis Functions Neural Network and Support Vector Machine models respectively, which are examined by comparing performances of training and simulation. Performances of training models are measured by mean of squared errors while that of simulation is done by average relative errors of the results. By comparison, Support Vector Machine is most robust to forecast cash flow, followed by Radial Basis Function Neural Network, the third Back Propagation Neural Network and the last Response Surface Model. The optimal result of each model depends on the window size of the transmitter.

Key words: *Performance, prediction, cash flow*

*Received: April 18, 2012*
*Revised and accepted: December 4, 2012*

## 1. Introduction

Cash flow is the movement of cash into or out of a business, project, or financial product. Cash flow is the most critical factor affecting profitability [1, 2]. For managers, investors and banks, it is important to know about cash flow which indicates the operation performance of a company [3-5]. Therefore, they do not only need to understand the self-financing capability of a business but also to know the future profitability. Cash Flow forecasts help them to build a model of the way in which cash moves within a project or organization [6]. They help a manager to predict whether the sales or income will cover the costs of operation. They also allow an investor to analyze whether a project will be sufficiently profitable to justify the effort put into it. Cash flow forecasts can also be useful for bankers to judge whether they should loan to a borrower or not [7, 8].

[*]Liu Hongjiu, Robert Rieg
Hochshule Aalen, Aalen, Germany, E-mail: lionlhj@163.com
[†]Liu Hongjiu, Hu Yanrong
Changshu Institute of Technology, Changshu, P. R. China

Although forecasting cash flow is vital, it is argued what method is better [9]. Because factors influencing cash flow are complex and dynamic, it is exceedingly difficult to build a mapped relation among the factors and cash flow. Accordingly, most models employ time series approaches to forecast future values using history data [10]. Various methods and approaches that use traditional statistical methods or artificial intelligence (AI) techniques have been developed to deal with time series problems. A survey conducted by Sapankevych and Sankar (2009) found that time series analysis methods including autoregressive filters, artificial neural networks (ANN) and support vector machines (SVM) have been applied in various fields [11]. They also found that the most important current application of time series analysis is in financial forecasting.

Several studies analyzed forecasting ability of statistical methods with financial time series. Kenneth et al. (1996) developed a multivariate, time-series prediction model. Their predictive results indicated that their models clearly outperformed firm-specific and common-structure ARIMA model as well as a multivariate, cross-sectional regression model popularized in the literature [12]. Mooi (2007) used multivariate regression models and panel data on a sample of 173 firms listed on Bursa Malaysia to forecast future cash flow from operations. His study showed predictive ability of all the regression models was improved as more years of historical data of predictors were incorporated [13]. Blyth and Kaka (2006) attempted to produce an individual $S$-curve for an individual project instead of producing an $S$-curve that is based on historical projects combined, they created a multiple linear regression model to forecast cash flow. Their conclusions are that the models produced more accurate results than the existing value and cost models [14]. Haahtela (2010) estimated cash flow of a company with the response surface method [15].

Although these studies show some improvements, other scholars argue that by applying AI methods we could achieve even better results. Many scholars have tried to develop AI models and/or systems that tackle various practical problems. Two critical considerations are applied when employing an AI approach. Firstly, an appropriate approach must be developed based on objectives. Secondly, an applicable historical data pool with relevant parameters must be built. With AI approaches, even though the relationship between inputs and output is not identified, predicted result may still be assessed with an acceptably high degree of accuracy. Chua et al. (1997) used neural networks to assess project budget performance [16]. Boussabaine et al. (1999) used a neural network approach with initial cash flow for periods to address sequential period cost flows [17]. Lowe et al. (1993) used expert systems to help clients manage cash flows [18]. Lokmic and Smith (2000) introduced back-propagation neural networks as an alternative to cash flow forecasting. They also compare accuracy results of the neural network method with regression and a heuristic model [19]. Park et al. (2005) proposed a cash flow forecasting model for construction projects that considered both variable cost weights and time lag [20]. Wang and Yin (2004) analyzed the reason to select Back Propagation (BP) Neural Network model to forecast. They used BP neural network model to forecast free cash flow in order to break some inherent limitations of traditional statistical time series [21].

Cheng and Roy (2011) fused fuzzy logic, weighted support vector machines and a fast messy genetic algorithm, their simulation performed on historical cash flow data demonstrated that the support vector machine is an effective tool for predicting cash flow [22].

Above-mentioned scholars did many researches on how to predict cash flow. Every scholar emphasized that their method was better. However, under the condition of small sample size, few studies are done to test different methods. In order to solve the problem, the primary objective of this research is to compare the prediction performance of the models of Response Surface, BP neural network, Radial Basis Functions (RBF) neural network and Vector Machine (SVM). The rest of this article is organized as follows. Section 2 describes sample selection and the sliding window technique. Section 3 presents the methodology and simulation of RSM. Section 4 presents the methodology and simulation of BP. Section 5 presents the methodology and simulation of RBF. Section 5 presents the methodology and simulation of SVM. And section 7 contains a summary and conclusions of the study.

# 2. Sample Selection and Sliding Window

## 2.1 Sample selection

Because prior researches depended on different sample data, it is difficult to judge the advantages or disadvantages of different methods. Every researcher emphasized that their method has better accuracy and robustness. In order to compare robustness of different methods impartially, it is necessary to select same sample data to test the performance of different models. The data of this paper are from Faw Car Limited Company (Ticker symbol: 000800), Chinese security market. During the period of its operation, there are no mergers & acquisitions and restructuring to happen because they can bring an abnormal change of cash flow of a company, which will influence accuracy of prediction. Because Chinese security market was open in 1989, the size of sample data is small. The sample has 34 time series data of cash flow from December 31$^{\text{th}}$, 2002 to March 15$^{\text{th}}$, 2011 (quarterly). Cash flow (CF) of a company denoted by $CF_t(t = 0, 1, \ldots, n)$ is a time series where $t$ represents elapsed time and $n$ represents the length of the time series.

In simulation, all data are normalized into a (0, 1) range, which helps avoid attributes with greater numeric ranges dominating those with smaller numeric ranges, and also helps avoid numerical difficulties (Hsu et al., 2003) [15]. The function used to normalize data is shown in Eq. (1).

$$CF'_t = \frac{(CF_t - CF_{t\,\min})}{(CF_{t\,\max} - CF_{t\,\min})}, \tag{1}$$

where, $CF_{t\,\max}$ and $CF_{t\,\min}$ are the maximum and minimum of a time series respectively.

## 2.2 Sliding window

This research is based on the sliding temporal window technique. The transmitter and receiver have a window size $wt$ and $wr$ respectively [23, 24]. We select transmitter's window size $wt$ varying from one to eight ($wt = 1, 2, \ldots, 8$) in order to find by what size good prediction performance can be acquired. The receiver's window size $wr$ is equal to one fixedly. Thus, when we slide the windows of the transmitter and receiver over a time series of the cash flow simultaneously, we can get input vector matrix $P(i,\ j)$ ($i = 1, 2, \ldots, wt$; $j = 1, 2, \ldots, n - wt$) and output vector matrix $T(j)$ ($j$=1, 2, $n$-$wt$) of a model (the windows limited in the time series).

$$P(i,j) = \begin{pmatrix} CF_1 & CF_2 & \ldots & CF_{n-wt} \\ CF_2 & CF_3 & \ldots & CF_{n-w+1} \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ CF_{wt} & CF_{wt+1} & \ldots & CF_{n-1} \end{pmatrix}, \tag{2}$$

$$T(j) = \begin{pmatrix} CF_{wt+1} & CF_{wt+2} & \ldots & CF_n \end{pmatrix}. \tag{3}$$

$P(i,\ j)$ and $T(j)$ are divided into training data and test data respectively. Every group of training data or test data consists of $P(i,\ j)$ and $T(j)$. For example, input 1 consists of $P(1, j)$ and $T(1)$ ($j = 1, \ldots, wt$). There are three inputs and outputs vectors in the group of test data which are used to check the forecasting performance of the trained model. And training data are applied to train a model.

If we assume that the cash flow of the time $t$ is decided by that of the time $t-1,\ t-2, \ldots, t-wt$. [25], we can find a function $f : \Re^{wt} \rightarrow \Re$ such as to obtain an estimate of $CF$ at time $t$ from the $wt$ time steps back from time $t$, so that:

$$CF(t) = f(CF(t-1), CF(t-2), \ldots, CF(t-wt)), \tag{4}$$

$$T(j) = f(P(i,j)), (i = 1, 2, \ldots, wt; j = 1, 2, \ldots, n - wt). \tag{5}$$

# 3. Response Surface Model and Simulation

In statistics, response surface methodology (RSM) explores the relationships between several explanatory variables and one or more response variables. The method was introduced by G. E. P. Box and K. B. Wilson in 1951. The main idea of RSM is to use a sequence of designed experiments to obtain an optimal response. Box and Wilson suggest using a second-degree polynomial model to do this. They acknowledge that this model is only an approximation, but used it because such a model is easy to estimate and apply, even when little is known about the process [26, 27]. If $y$ represents output and $x_i$ ($i$=1, 2, ..., $n$) represents input, the equation of response surface model is following:

$$y = b_0 + \sum_{i=1}^{n} b_i x_i + \sum_{i=1,j=1}^{n} b_{ij} x_i x_j + \sum_{i=1}^{n} b_{ii} x_i^2, (i \neq j), \tag{6}$$

where,

$b_0$ — Constant terms,

$b_i$ — Linear terms,

$b_{ij}$ — Interaction terms,
$b_{ii}$ — Squared terms [28, 29].

The model is trained and tested in *Matlab 2009b*. We use Function *rstool* of *Matlab* Statistic Toolbox to calculate the $b_0$, $b_i$, $b_{ij}$, $b_{ii}$. For Function *rstool*, there are four models to choose: *Linear*, *Pure Quadratic*, *Interactions*, *Full Quadratic.*

*Linear* — Constant and linear terms (the default),
*Pure Quadratic* — Constant, linear, and squared terms,
*Interactions* — Constant, linear, and interaction terms,
*Full Quadratic* — Constant, linear, interaction, and squared terms.

Performances of different models depend on their root mean square errors (*rmse*). The smaller its *rmse* is, the better its performance is. The results of *rmse* are listed in Tab. I for different models and *wt*s. According to Tab. I, *Interactions* model fits the training data best when *wt* is 6. It prediction plot is seen in Fig. 1.

| ***wt*** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **Linear** | 0.1641 | 0.1638 | 0.1632 | 0.1689 | 0.1465 | 0.1533 | 0.1611 | 0.1676 |
| **Pure Quadratic** | 0.1669 | 0.1691 | 0.1661 | 0.1694 | 0.1465 | 0.1505 | 0.1516 | 0.1303 |
| **Interactions** | 0.1641 | 0.167 | 0.1726 | 0.1754 | 0.1015 | **0.1012** | - | - |
| **Full Quadratic** | 0.1669 | 0.1717 | 0.1739 | 0.1856 | 0.1369 | - | - | - |

**Tab. I** *Rmse of different models.*

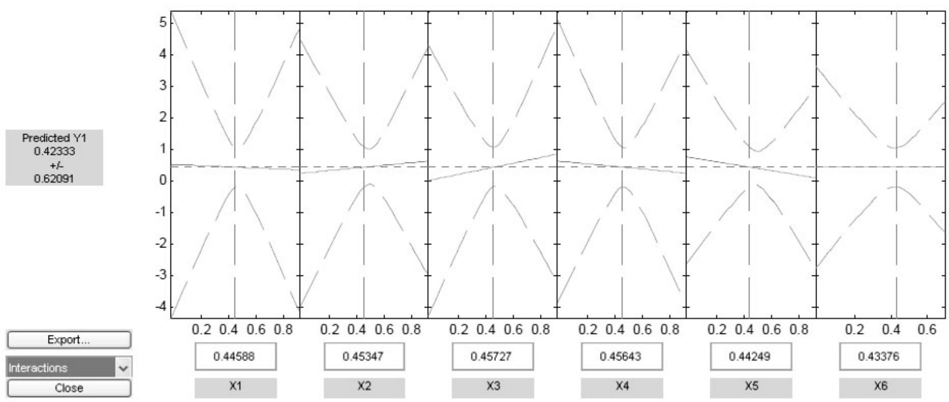


**Fig. 1** *Prediction plot of Interactions Model.*

We can export the results of *rstool* into the *workshop* of *Matlab* and get fitted coefficients $b_0$, $b_i$, $b_{ij}$ and $b_{ii}$ of four Models. Inputting the coefficients and test into Eq. (6), the prediction value and its errors can be easily calculated as Tab. II.

| *wt* | *Linear* | *Pure Quadratic* | *Interactions* | *Full Quadratic* |
|---|---|---|---|---|
| 1 | 0.35029 | 0.34569 | 0.35029 | 0.34569 |
| 2 | 0.313239 | 0.331248 | 0.315309 | 0.318497 |
| 3 | 0.313459 | 0.348399 | 0.300741 | 0.369618 |
| 4 | 0.311432 | 0.337864 | 0.298176 | 0.442151 |
| 5 | 0.300523 | 0.320734 | 0.361373 | 0.496463 |
| 6 | 0.290342 | 0.328346 | 0.454112 | - |
| 7 | **0.280113** | 0.413551 | - | - |
| 8 | 0.295761 | 0.465548 | - | - |

**Tab. II** *Average relative errors of test data with different wts in RSM.*

Compared with other models, it is obvious that the MAPE of *linear* are smallest when *wt* equals 7 with MAPE of 28.01 percent. So, performance of simulation is not good (see Fig. 2).

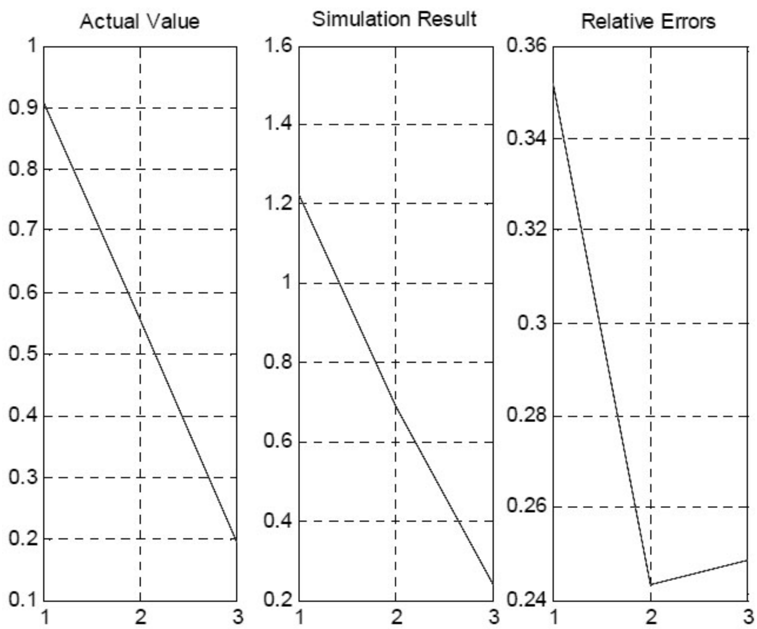


**Fig. 2** *Performance of simulation of wt=6 in RSM.*

## 4. BP Neural Network Model and Simulation

BP neural network (BPNN) is a common method of teaching artificial neural networks how to perform a given task, which is a supervised learning method, and is a generalization of the delta rule [30]. It requires a teacher that knows, or can calculate, the desired output for any input in the training set [31]. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop) [32]. Although BP can be applied to networks with any number of layers, just as for networks with binary units, it has been shown that only one layer of hidden units succeeds to approximate any function with finitely many discontinuities to arbitrary precision, provided the activation functions of the hidden units are non-linear (the universal approximation theorem) [33]. The activation of a hidden unit is a function $f_k$ of the weighted inputs plus a bias $\theta_k(t)$, as given in Eq. (7).

$$y_k(t+1) = f_k\left(\sum_j w_{jk}(t)y_j(t) + \theta_k(t)\right), \tag{7}$$

where $y_k(t+1)$ is the $k$ output of the $t$ hidden layer, $y_j(t)$ is the $j$ input of the $t$ hidden layer, $w_{jk}$ is the weight between the $j$th input $y_j(t)$ and the $k$th output $y_k(t+1)$, and $\theta_k(t)$ is the bias [34]. The model is designed and operated in the Neural Network Toolbox of *Matlab 2009b*. We use function *newff* to create a network. The network structure is shown in Fig. 3.

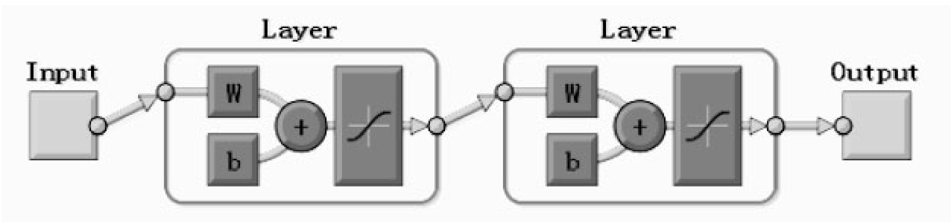


**Fig. 3** *The structure of BP neural network model.*

Before training and testing the model, we need to choose the parameters of the BPNN model. Firstly, number of neutrons of input, hidden and output layers should be decided. Number of neurons of input layer equals the number $n$ of input vectors. That of output layer equals one because there is one data in the output. Number $n1$ of hidden layer depends on the number of neurons of input layer, which can be calculated as follows Eq. (8) [35]:

$$n1 = 2n + 1. \tag{8}$$

Secondly, transfer functions will be chosen. We choose *tansig* and *logsig* as those of hidden layer and output respectively because input and output data are between $-1$ and 1.

Thirdly, learn and training functions are *learngdm* and *trainlm*. Leargdm is the gradient descent with momentum weight/bias. *Trainlm* is a network training function that updates weight and bias states according to *Levenberg-Marquardt* optimization. It is often the fastest back propagation algorithm in the toolbox, and is highly recommended as a first choice supervised algorithm, although it does require more memory than other algorithms.

Fourthly, the network's performance is controlled by function *mse* (mean of squared errors). The goal of performance is 0.0001, which is accurate enough to train a model.

After all parameters are set, we start to train the model with 2000 epochs. Because training performances of the models with different *wt*s are the same, prediction performances of the model are measured by relative errors between test data and simulation results. At the same time, considering that there are some limitations for BPNN, for example that the convergence is very slow and not guaranteed, the result may generally converge to any local minimum on the error surface [36], every group of test data is simulated twice (see Fig. 4 and Tab. III).

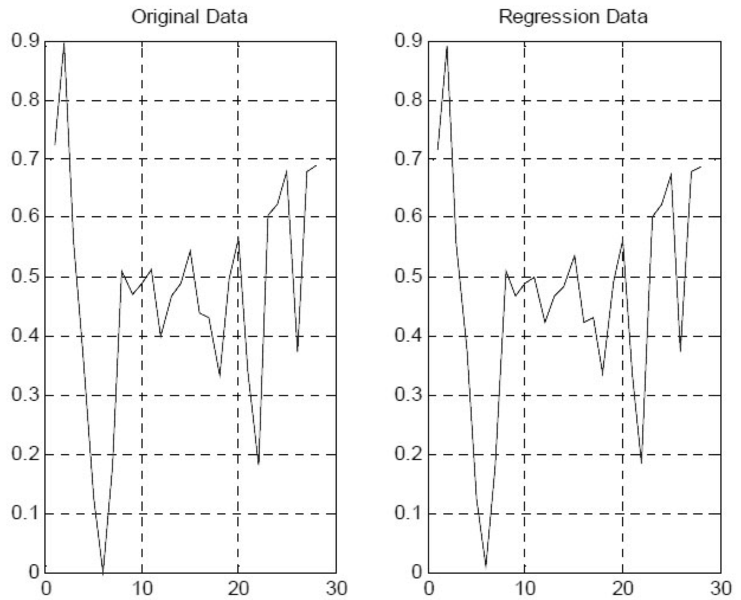


**Fig. 4** *Performance of training of wt=3 in BP.*

| No. of test data | *wt*=1 | *wt*=2 | *wt*=3 | *wt*=4 | *wt*=5 | *wt*=6 | *wt*=7 | *wt*=8 |
|---|---|---|---|---|---|---|---|---|
| **1 (%)** | 6.96 | 47.33 | 3.36 | 4.85 | 72.66 | 80.05 | 8.91 | 12.69 |
| **2 (%)** | 71.70 | 7.18 | 1.95 | 20.13 | 56.18 | 87.83 | 49.22 | 44.76 |
| **3 (%)** | 4.87 | 109.25 | 43.93 | 204.06 | 50.11 | 74.16 | 245.74 | 333.38 |
| **MAPE (%)** | 27.84 | 54.59 | **16.41** | 76.35 | 59.65 | 80.68 | 101.29 | 130.28 |
| **Training Mse** | 0.0075 | 9E-05 | 9E-05 | 4.07E-06 | 2.62E-05 | 3.75E-05 | **1.23E-05** | 3.75E-05 |

**Tab. III** *Relative errors of test data with different wts in BP.*

Evidently, when the window size of transmitter is 3, the average relative error (16.41 percent) is smallest from Tab. IV. Performances of training and simulation

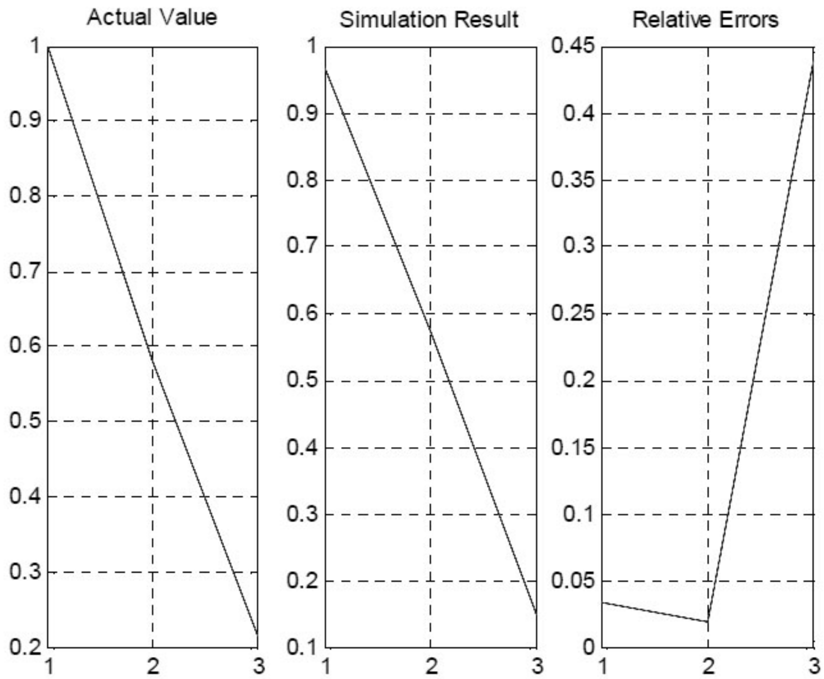


**Fig. 5** *Performance of simulation of wt=3 in BP.*

| **No. of test data** | *wt*=1 | *wt*=2 | *wt*=3 | *wt*=4 | *wt*=5 | *wt*=6 | *wt*=7 |
|---|---|---|---|---|---|---|---|
| **1 (%)** | 40.45 | 41.43 | 29.42 | 62.65 | 73.46 | 7.55 | 79.53 |
| **2 (%)** | 69.45 | 71.13 | 50.52 | 107.59 | 126.82 | 12.44 | 136.76 |
| **3 (%)** | 65.95 | 38.98 | 0.25 | 73.42 | 71.88 | 24.03 | 370.49 |
| **MAPE (%)** | 58.61 | 50.51 | 26.73 | 81.22 | 90.72 | **14.67** | 195.59 |
| **Mse** | 0.0251 | 0.0238 | 0.0219 | 0.0188 | 0.0092 | 0.0052 | **0.0043** |
| **SPREAD** | 200 | 200 | 200 | 200 | 300 | 300 | 300 |

**Tab. IV** *Relative errors of test data with different wts in RBF.*

of *wt*=3 are given by Fig. 4 and Fig. 5. Although there is strong ability of approximation for BP, many times the tests show that the average relative errors are not stable, sometimes being very big.

## 5. RBF Neural Network Model and Simulation

Radial Basis Functions (RBF) is embedded into a two-layer feed-forward neural network. Such a network is characterized by a set of inputs and a set of outputs.

Between the inputs and output there is a layer of processing units called hidden units. Each of them implements a radial basis function [37]. The way in which the network is used for data modeling is different when approximating time-series and in pattern classification. In time-series, the network inputs represent data samples at certain past time-laps, while the network has only one output representing a signal value. Various functions have been tested as activation functions for RBF networks. The most used activation function is the Gaussian function [38, 39]. Mixtures of Gaussians have been considered in various scientific fields. The Gaussian activation function for RBF networks is given by Eq. (9).

$$\varphi_j(X) = \exp[-(X-u_j)^T \sum\nolimits_j^{-1} (X-u_j)]. \tag{9}$$

For $j$=1, …, $L$, where $X$ is the input feature vector, $L$ is the number of hidden units, $\mu_j$ and $\sum_j$ are the mean and the covariance matrix of the $j$th Gaussian function [40].

The model is designed and operated in the *Neural Network Toolbox* of *Matlab 2009b.* Function *newrbe* is used to create a network. In the model parameter *SPREAD* (a radius) is vital. The larger the *SPREAD*, the smoother the function approximation will be. But, too large a spread can cause numerical problems [41]. In order to obtain an optimal value for *SPREAD,* we increase it varying from 1 to 1000. The result of simulation shows that performance of the model is good if *SPREAD* is between 200 and 300. Relative errors of test data are seen in Tab. V.

| **No. of test data** | *wt*=1 | *wt*=2 | *wt*=3 | *wt*=4 | *wt*=5 | *wt*=6 | *wt*=7 |
|---|---|---|---|---|---|---|---|
| **1** (%) | 30.19 | 16.05 | 34.69 | 23.75 | 17.89 | 20.37 | 6.11 |
| **2** (%) | 0.27 | 16.94 | 18.86 | 9.23 | 0.25 | 10.52 | 7.93 |
| **3** (%) | 19.13 | 18.90 | 18.75 | 15.08 | 17.71 | 18.27 | 23.74 |
| **MAPE (%)** | 16.53 | 17.30 | 24.10 | 16.02 | **11.95** | 16.39 | 12.59 |
| **Mse** | 0.0420 | 0.0241 | 0.0369 | 0.0421 | 0.0354 | 0.0052 | **0.0027** |
| **Weight function** | *wmyriad* | *whampel* | *whampel* | *wlogisti* | ***whampel*** | *whuber* | *whampel* |

**Tab. V** *Relative errors of test data with different wts.*

We can draw a conclusion that average relative error is smallest when *wt* equals six. The relative error of 14.67 percent may be accepted although it is not small enough. Performances of training and simulation are given by Fig. 6 and Fig. 7.

# 6. Support Vector Machine Model and Simulation

A support vector machine (SVM) is a concept in computer science for a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. The standard SVM takes a set of input

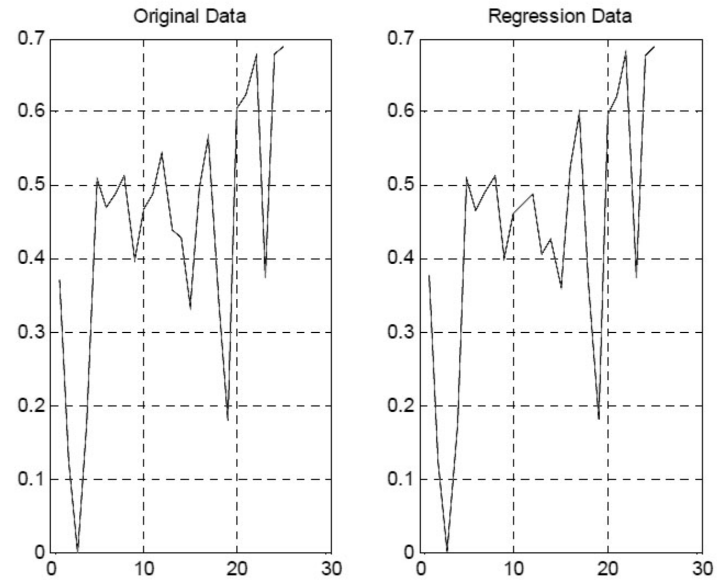


**Fig. 6** *Performance of training of wt=6 in RBF.*

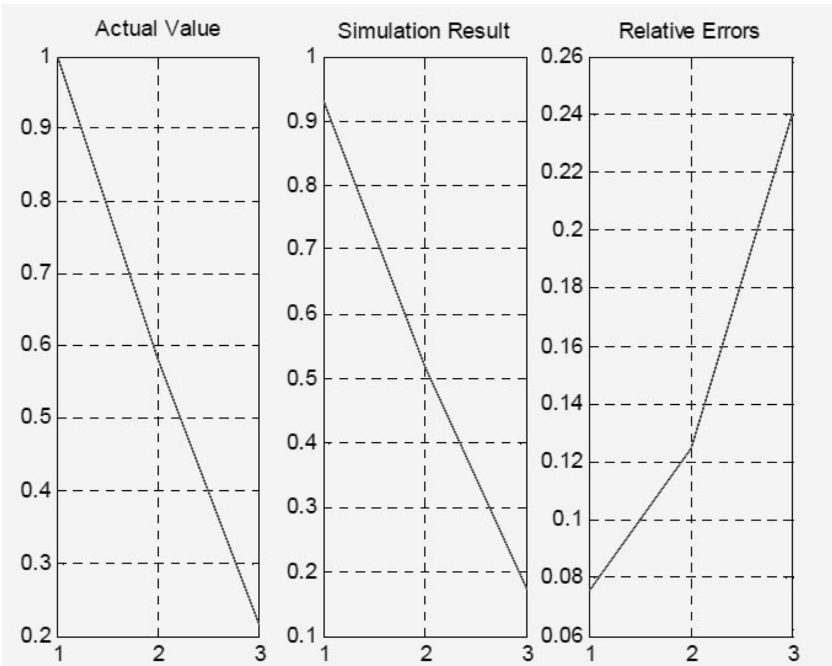


**Fig. 7** *Performance of simulation of wt=6 in RBF.*

data and predicts, for each given input, which of the two possible classes the input is a member of, which makes the SVM a non-probabilistic binary linear classifier [42]. Given a set of training examples, each marked as belonging to one of the two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other [43]. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on [44, 45].

In this paper, we introduce Least Squares Support Vector Machines (LS-SVM) proposed by J. Vandewalle and J.A.K. Suykens for regression, which is closely related to regularization networks and Gaussian processes but additionally emphasize and exploit primal-dual interpretations [46].

In order to obtain an LS-SVM model (with the RBF kernel), we need two extra tuning parameters: $\gamma$ (gam) is the regularization parameter, determining the trade-off between the training error minimization and smoothness of the estimated function. $\sigma^2$ (sig2) is the kernel function parameter [47]. The model is created and operated in the *LS-SVMlab* Toolbox. We choose training function *robustlssvm* and weight functions including *whuber*, *whampel*, *wlogistic* and *wmyriad* as training parameters. The optimal results of training with different weight functions and simulation are displayed in Tab. VI, Fig. 8, Fig. 9.

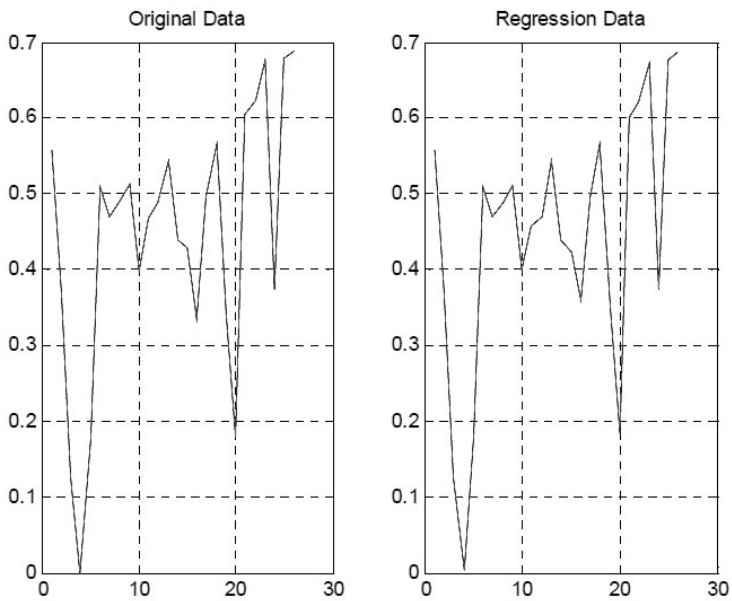


**Fig. 8** *Performance of training of wt=5 in SVM.*

For SVM model, the average relative error is smallest when *wt* equals five, the relative error of 11.95 percent is only a little smaller than that of RBF model.

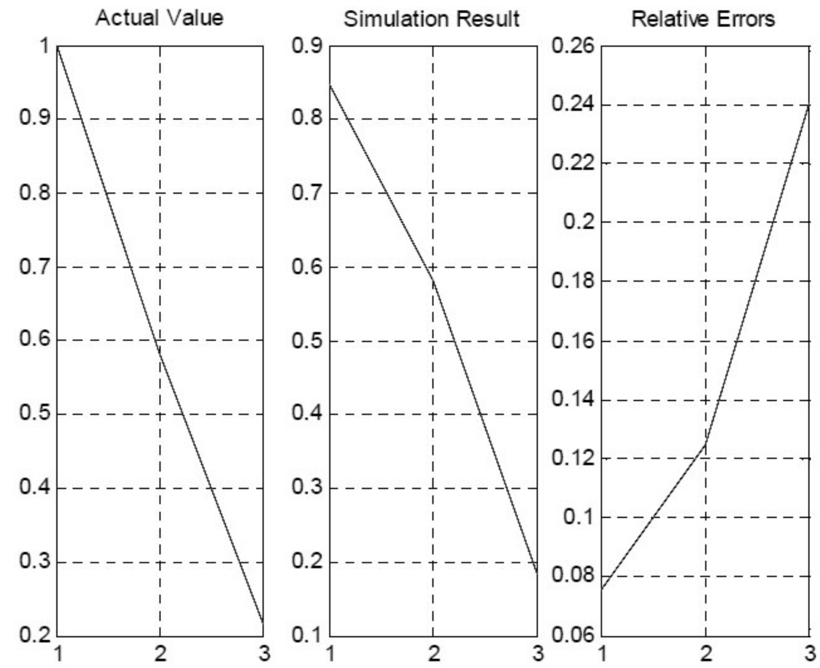


**Fig. 9** *Performance of simulation of wt=5 in SVM.*

# 7. Conclusions

This research drew a comparison of robustness of forecasting cash flow among the Response Surface, Back Propagation Neural Network, Radial Basis Function and Support Vector Machine. The research is based on small size samples (34 data, less than 300). In order to predict the future cash flow, we also use the sliding temporal window technique which divides sample data into two parts: input and output. And we assume that the value of the receiver window is decided by that of the transmitter window, which means that future cash flow depends on history cash flows.

Analyzing the performance of training and simulation of each model, we can draw a few conclusions. Firstly, among all models, SVM has the best predicting performance ($wt$=5), followed by RBF, BP and RSM. Secondly, BP has the best training performance ($wt$=7), followed by SVM, RBF and RSM. As a whole, the predicting robustness of SVM and RBF is stronger than that of the others under the condition of small size samples.

However, above findings do not imply SVM is the most robust tool for forecasting cash flow because this study has compared only four popular methods. In fact, there are still newer methods such as genetic algorithm, ant colony optimization and particle swarm optimization etc. which are valuable to be discussed in the future. In addition, because the quantity of samples could not be big enough, it is suggested to collect the data of more companies to examine the performance of different models of predicting cash flow in the future work.

## Acknowledgement

I am grateful for the work environment provided by Hochschule Aallen. I would like to thank my cooperators Prof. Dr. Robert Rieg and Dr. Hu Yanrong who gave me many good advices and ideas. The work was supported by the Jiangsu Philosophical and Social Science Program for Colleges and Universities (2010SJB790001).

# References

[1] Hwee N. G., Tiong R. L. K.: Model on cash flow forecasting and risk analysis for contracting firms. International Journal of Project Management, **20**, 5, 2002, pp. 351-363.

[2] Orpurt S. F., Zang Y.: Do Direct Cash Flow Disclosures Help Predict Future Operating Cash Flows and Earnings? Accounting Review, **84**, 3, 2009, pp. 893-935.

[3] DeFond M. L., Hung M. Y.: Investor protection and analysts' cash flow forecasts around the world. Review of Accounting Studies, **12**, 2-3, 2007, pp. 377-419.

[4] Estep P. W.: Cash flows, asset values, and investment returns - Tying return forecasting to uses of cash. Journal of Portfolio Management, **29**, 3, 2003, pp. 17-26.

[5] Lehavy R.: Discussion of "Are earnings forecasts more accurate when accompanied by cash flow forecasts?". Review of Accounting Studies, **14**, 2-3, 2009, pp. 392-400.

[6] Gormley F. M., Meade N.: The utility of cash flow forecasts in the management of corporate cash balances. European Journal of Operational Research, **182**, 2, 2007, pp. 923-935.

[7] Givoly D., Hayn C., Lehavy R.: The Quality of Analysts' Cash Flow Forecasts. Accounting Review, **84**, 6, 2009, pp. 1877-1911.

[8] Riedl E. J.: Discussion of Accounting Conservatism and the Temporal Trends in Current Earnings' Ability to Predict Future Cash Flows versus Future Earnings: Evidence on the Trade-off between Relevance and Reliability. Contemporary Accounting Research, **27**, 2, 2010, pp. 461-467.

[9] Pirchegger B.: Hedge accounting incentives for cash flow hedges of forecasted transactions. European Accounting Review, **15**, 1, 2006, pp. 115-135.

[10] Cryer D. J., Chan K.-S.: Time Series Analysis with Applications in R, seconded. New York: Springer, 2008.

[11] Sapankevych N., Sankar R.: Time Series Prediction Using Support Vector Machines: A Survey. Computational Intelligence Magazine, IEEE **4**, 2, May 2009, pp. 24-38.

[12] Kenneth S., Lorek G. and Lee W.: A multivariate time-series prediction model for cash-flow data. Accounting Review, **71** (1), 1996, pp. 81-101.

[13] Mooi T. L.: Predicting future cash flows: does cash flow have incremental information over accrual earnings? Malaysian Accounting Review, **6**, 2, 2007, pp. 63-80.

[14] Blyth K., Kaka A.: A novel multiple linear regression model for forecasting S-curves. Engineering, Construction and Architectural Management, **13**, 1, 2006, pp. 82-95.

[15] Haahtela T. J.: Regression sensitivity analysis for cash flow simulation based real option valuation. Procedia Social and Behavioral Sciences, **2**, 6, 2010, pp. 7670-7671.

[16] Chua D. K. H., Loh P. K., Kog Y. C., Jaselskis E. J.: Neural networks for construction project success. Expert Systems with Applications, **13**, 4, 1997, pp. 317-328.

[17] Boussabaine A. H., Thomas R., Elhag T.: Modeling cost-flow forecasting for water pipeline projects using neural network. Engineering. Construction and Architectural Management, **6**, 3, 1999, pp. 213-224.

[18] Lowe J., Neveen M., Lowe H.: Cash Flow Management: an Expert System for the Construction Client. Journal of Applied Expert Systems, **1**, 2, 1993, pp. 134-152.

[19] Lokmic L., Smith K. A.: Cash flow forecasting using supervised and unsupervised neural networks. Proceedings of the IEEE-INNS-ENNS International Joint Conference, **6**, 2000, pp. 343-347.

[20] Park H. K., Han S. H., Russell J. S.: Cash Flow Forecasting Model for General Contractors Using Moving Weights of Cost Categories. Journal of Management in Engineering, **21**, 4, 2005, pp. 164-172.

[21] Wang H. C., Yin M. Q.: Time series forecasts of free cash flow with BP neural network model, 2004.

[22] Cheng M.-Y., Roy A. F. V.: Evolutionary fuzzy decision model for cash flow prediction using time-dependent support vector machines. International Journal of Project Management, **29**, 1, 2011, pp. 56-65.

[23] Buzikashvili N.: Comparing Web Logs: Sensitivity Analysis and Two Types of Cross-Analysis. Proceedings of AIRS'2006, 2006, pp. 508-513.

[24] Khan M. S., Coenen F., Reid D., Patel R., Archer L.: A sliding windows based dual support framework for discovering emerging trends from temporal data. Knowledge Based Systems - KBS, **23**, 4, 2010, pp. 316-322.

[25] Chu C.-S. J.: Time series segmentation: A sliding window approach. Information Sciences, **85**, 1-3, 1995, pp. 147-173.

[26] Box G. E. P., Wilson K. B.: On the Experimental Attainment of Optimum Conditions. Journal of the Royal Statistical Society Series B, **8**, 1, 1951, pp. 1-45.

[27] Crispim E. G., Piai J. F., Muniz A. F. R. a. E. C.: Addition of methacryloil groups to poly(vinyl alcohol) in DMSO catalyzed by TEMED: Optimization through response surface methodology Polymer Testing, **25**, 3, 2006, pp. 377-383.

[28] Atkinson A., Donev A., Tobias R.: Optimum Experimental Designs, with SAS (Oxford Statistical Science Series): Oxford University Press, 2007.

[29] Kalila S. J., Maugerib F., Rodrigues M. I.: Response surface analysis and simulation as a tool for bioprocess design and optimization. Process Biochemistry, **35**, 6, 2000, pp. 539-550.

[30] Bryson J. A. E., Ho Y.-C.: Applied Optimal Control: Optimization, Estimation, and Control: Taylor & Francis, 1975.

[31] Russell S., Norvig P.: Artificial Intelligence: A Modern Approach. 2 ed.: Prentice Hall, 2002.

[32] Jiang W., Zhang L. and Wang P.: Nonlinear time series forecasting of time-delay neural network embedded with Bayesian regularization. Applied Mathematics and Computation, **205**, 1, 2008, pp. 123-132.

[33] Wang J.-Z., Wang J.-J., Zhang Z.-G.: Forecasting stock indices with back propagation neural network. Expert Systems with Applications, 2011.

[34] Liu H., Hu Y., Ma W.: Cash flow forecasting based on GA-BP model in valuation. Proceedings of 2010 IEEE International Conference on Advanced Management Science, **2**, 2010, pp. 299-302.

[35] Mao Y., Huang X.: Human recognition based on head-shoulder moment feature. IEEE International Conference on Service Operations and Logistics, and Informatics 2008, pp. 622-625.

[36] Hu C., Zhao F.: Improved Methods of BP Neural Network Algorithm and its Limitation. International Forum on Information Technology and Applications (IFITA), 2010, pp. 11-14.

[37] Rojas I. et al.: Time series analysis using normalized PG-RBF network with regression weights. Neurocomputing, **42**, 1-4, 2002, pp. 267-285.

[38] Gan M., Peng H., Peng X., Chen X., Inoussa G.: A locally linear RBF network-based state-dependent AR model for nonlinear time series modeling. Information Sciences, **180**, 22, 2010, pp. 4370-4383.

[39] Rank E., Kubin G.: Application of Bayesian trained RBF networks to nonlinear time-series modeling. Signal Processing – From signal processing theory to implementation, **83**, 27, 2003, pp. 1393-1410.

[40] Rivas V. M., Merelo J. J., Castillo P. A., Arenas M. G., Castellano J. G.: Evolving RBF neural networks for time-series forecasting with EvRBF. Information Sciences, **165**, 3-4, 2004, pp. 207-220.

[41] Kokshenev I., Braga A. P.: An efficient multi-objective learning algorithm for RBF neural network. Neurocomputing, **73**, 16-18, 2010, pp. 2799-2808.

[42] Rubio G., Pomaresa H., Rojasa I., Herrera L. J.: A heuristic method for parameter selection in LS-SVM: Application to time series prediction. International Journal of Forecasting, **27**, 3, 2011, pp. 725-739.

[43] Sanz S. S. et al.: Performance Comparison of Multilayer Perceptrons and Support Vector Machines in a Short-Term Wind Speed Prediction Problem. Neural Network World, **19**, 1, 2009, pp. 37-51.

[44] Orsenigo C., Vercellis C.: Combining discrete SVM and fixed cardinality warping distances for multivariate time series classification. Pattern Recognition, **43**, 11, 2010, pp. 3787-3794.

[45] Thissen U., Van Brakel R., Weijer A. P. D., Melssen W. J., Buydens L. M. C.: Buydens Using support vector machines for time series prediction. Chemometrics and Intelligent Laboratory Systems, **69**, 1-2, 2003, pp. 35-49.

[46] Sun J., Zheng C., Zhou Y., Bai Y., Lu J.: Nonlinear noise reduction of chaotic time series based on multidimensional recurrent LS-SVM. Neurocomputing, **71**, 16-18, 2008, pp. 3675-3679.

[47] Kim K.-J.: Financial time series forecasting using support vector machines. Neurocomputing, **55**, 1-2, 2003, pp. 307-319.