Expert Systems with Applications 42 (2015) 3234–3241

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

# Recurrent neural network and a hybrid model for prediction of stock returns

Akhter Mohiuddin Rather [a,*], Arun Agarwal [a], V.N. Sastry [b]

[a] School of Computer and Information Sciences, University of Hyderabad, Hyderabad 500046, A.P., India
[b] Institute for Development and Research in Banking Technology, Castle Hills, Road No. 1, Masab Tank, Hyderabad 500057, A.P., India

ARTICLE INFO

Article history:
Available online 11 December 2014

Keywords:
Recurrent neural network
Autoregressive neural network
Genetic algorithms
Time series
Stock returns

ABSTRACT

In this paper, we propose a robust and novel hybrid model for prediction of stock returns. The proposed model is constituted of two linear models: autoregressive moving average model, exponential smoothing model and a non-linear model: recurrent neural network. Training data for recurrent neural network is generated by a new regression model. Recurrent neural network produces satisfactory predictions as compared to linear models. With the goal to further improve the accuracy of predictions, the proposed hybrid prediction model merges predictions obtained from these three prediction based models. An optimization model is introduced which generates optimal weights for proposed model; the model is solved using genetic algorithms. The results confirm about the accuracy of the prediction performance of recurrent neural network. As expected, an outstanding prediction performance has been obtained from proposed hybrid prediction model as it outperforms recurrent neural network. The proposed model is certainly expected to be a promising approach in the field of prediction based models where data is non-linear, whose patterns are difficult to be captured by traditional models.



## 1. Introduction

Prediction of stock returns has drawn attention from industry as well as from academicians. Drawing inferences and making precise predictions of stocks are challenging for most researchers because stock data is noisy and non-stationary (Abu-Mostafa & Atiya, 1996). However artificial neural networks (ANNs) are one of the most accurate prediction techniques used in stock prediction (Haykin, 1994). Prediction of stock returns are broadly classified into linear models and non-linear models. Linear models include autoregressive integrated moving average models, exponential smoothing models, and generalized autoregressive conditional heteroskedasticity as well as stochastic volatility model (Durham, 2007). Non-linear models include those models which are based on artificial intelligence such as ANNs (Chen, 1994), support vector machines (Burges, 1998; Huang, Nakamori, & Wang, 2005), genetic algorithms and particle swarm optimization (Majhi & Panda, 2008). These non-linear models overcome the limitation of linear models as they are able to capture non-linear pattern of data, thus improving their prediction performance (Armano, Marchesi, & Murru, 2005; Kim & Han, 2000). Chen, Fan, Chen, and Wei (2013) used ANN for the purpose of parameter optimization by constructing a model for stock price prediction. Liao and Wang (2010) investigated statistical properties of the fluctuations of Chinese stock index using ANNs. Shen, Guo, Wu, and Wu (2011) used radial basis function neural network to predict stock indices of Shanghai stock exchange. An algorithm for predicting stock prices was developed by means of wavelet de-noising-based ANN (Wang, Wang, Zhang, & Guo, 2011). As some ANNs are unable to yield accurate predictions, Kim and Ahn (2012) attempted to find global optimization approach of ANN. Recently Saadaoui and Rabbouch (2014) proposed wavelet-based ANN so as to make two dimensional information to make predictions. Genetic algorithm (GA) on the other hand is a type of searching algorithm proposed by Holland (1992) and has been well enhanced by other researchers (Gen & Cheng, 2000; Koza, 1994). A global optimization approach of ANN was performed by Dai, Wu, and Lu (2012), they used GA to optimize multiple architectural factors and feature transformations of ANN. Recently a study compared the importance of GA with those of logistic regression and support vector machine (Gordini, 2014).

Many hybrid forecasting models have also appeared over time which may include hybridization of both linear as well as non-linear models (Kwon & Moon, 2007; Wang, Wang, Zhang, & Guo, 2012; Zhang, 2003). Hsieh, Hsiao, and Yeh (2011) integrated wavelet transforms and recurrent neural network based on artificial bee colony algorithm. A novel hybrid model of ANNs

* Corresponding author. Tel.: +91 9989271952; fax: +91 040 23010120.
E-mail addresses: akhter.aki@gmail.com (A.M. Rather), aruncs@uohyd.ernet.in (A. Agarwal), vnsastry@idrbt.ac.in (V.N. Sastry).

http://dx.doi.org/10.1016/j.eswa.2014.12.003
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

and autoregressive integrated moving average was proposed by Khashei and Bijari (2010). Chen, Chen, Fan, and Huang (2013) used adaptive network-based fuzzy inference system to construct a hybrid prediction model for predicting stock prices and their direction. Chen (2013) combined adaptive-network-based fuzzy inference system model so as to construct a model for predicting business failures. By merging self-organizing map and support vector regression, Chen (2012) proposed a multi-phased and dynamic evaluation model of the corporate financial structure. A hybrid model was proposed by merging backpropagation neural network, feature selection and genetic programming so as to tackle future price forecasting problems (Hsu, 2013). Adhikari and Agrawal (2013) proposed a hybrid model by merging random walk model and ANN. In another work, a hybrid prediction model was proposed which is combination of autoregressive integrated moving average model and ANN (Wang, Zou, Su, Li, & Chaudhry, 2013). Rout, Majhi, Majhi, and Panda (2014) created a hybrid model by integrating autoregressive moving average model and differential evolution based training of ANN. Chaabane (2014) proposed a hybrid model for electricity market; the model uses autoregressive fractionally integrated moving average model as well as ANN. A multi-objective evolutionary algorithm was proposed for recurrent neural network using time series data (Smith & Jin, 2014). In a recent work Yi, Jin, John, and Shouyang (2014) proposed a hybrid model for financial volatility by integrating autoregressive integrated moving average model and ANN. Recently a hybrid prediction model was proposed by Rather (2014) in which multiplicative method and summation method are used along with recurrent neural network to predict stock returns.

The remainder of this paper is organized as follows: In Section (2), some existing mathematical models used in this work are discussed. Proposed hybrid prediction model and its mathematical formulation is discussed in detail Sections 3 and 4 respectively. Section 5 presents algorithm for proposed model. Experiments and results are presented in Sections 6 and 7 respectively. Finally Section 8 presents conclusions.

## 2. Prediction based models

This section discusses some prediction based models used in this work.

### 2.1. Autoregressive moving reference neural network

Suppose we have time series data of a stock with $T$ past returns as shown:

$$R = (r_{t-(T-1)}, \ldots, r_{t-1}, r_t) \tag{1}$$

A new type of autoregressive model for time series prediction was proposed by Freitas, de Souza, and de Almeida (2009). This regression method was implemented on ANN, thus called as autoregressive moving reference neural network, $AR-MRNN(p,k)$, where $p$ is the order of regression and $k$ is the delay from the reference location.

Using $AR-MRNN(p,k)$ following prediction system is implemented:

$$\widehat{r_{t+1}-z} = \Re(r_{t-(p-1)} - z, \ldots, r_{t-1} - z, r_t - z) \tag{2}$$

where $\widehat{r_{t+1}-z}$ is prediction for $r_{t+1} - z$ obtained at time $t$, $\Re$ is autoregressive predictor and $z$ is moving reference expressed as:

$$z = r_{t-(p-1)-k} \tag{3}$$

ANN uses Eq. (2) for input–output in supervised form. The output obtained from ANN are not the final predictions, rather final predictions are calculated by adding $z$ value to the output (from ANN) as shown:

$$\hat{r}_{t+1} = \widehat{r_{t+1}-z} + z \tag{4}$$

### 2.2. Autoregressive moving reference and recurrent neural network

Fig. 1 shows recurrent neural network (RNN) with one hidden layer used in this work. The network is supervised AR-MRNN having inputs $(r_t - z, r_{t-1} - z, \ldots, r_{t-(p-1)} - z)$ and expected output as: $\widehat{r_{t+1}-z}$. Long term memory shown in input layer holds the data and passes it to hidden layer after each pattern comes from input layer. Input layer possesses linear activation function having neurons equal to the regression order, i.e. $p$. Hidden layer has sixteen neurons $(h_{1,1}, \ldots, h_{1,16})$ and one neuron in output layer, both layers possessing logistic activation function. The network is able to learn as well as predict complex non-linear patterns of the data. This network is also known as Jordan Elman neural network; for training, the network uses backpropagation algorithm (Rumelhart & McClelland, 1986; Seker, Ayaz, & Turkcan, 2003).

### 2.3. Exponential smoothing

ES is a linear model used for prediction based process (Brown, 2004). ES model calculates geometric sum of past historical series which finally leads to one-step-ahead prediction as shown:

$$\hat{r}_{t+1} = \hat{r}_t + \alpha(r_t - \hat{r}_t) \tag{5}$$

where $\hat{r}_{t+1}$ is prediction for future value; $\hat{r}_t; \alpha$ is a smoothing factor in the range of $(0,1)$; $r_t - \hat{r}_t$ is prediction error.

### 2.4. Autoregressive moving average model

Introduced by Box and Jenkins (1970), ARMA models are very well known linear models in times series analysis. Their work is actually inspired from the early work of Yule (1926) and Wold (1938). ARMA model is constituted of two parts, autoregressive part, $AR(p)$ and moving average part, $MA(q)$, both parts are combined to form $ARMA(p,q)$ model; where $p$ and $q$ are order of AR model and MA model respectively. $AR(p)$ model is generally expressed as:

$$r_t = \omega_1 r_{t-1} + \omega_2 r_{t-2} + \cdots + \omega_p r_{t-p} + \varepsilon_t \tag{6}$$

Similarly $MA(q)$ model is generally expressed as:

$$r_t = \varepsilon_t - \varphi_1 \varepsilon_{t-1} - \varphi_2 \varepsilon_{t-2} - \cdots - \varphi_q \varepsilon_{t-q} \tag{7}$$

$ARMA(p,q)$ model is thus expressed by combining above two terms:

$$r_t = \omega_1 r_{t-1} + \omega_2 r_{t-2} + \cdots + \omega_p r_{t-p} + \varepsilon_t - \varphi_1 \varepsilon_{t-1} - \varphi_2 \varepsilon_{t-2} - \cdots - \varphi_q \varepsilon_{t-q} \tag{8}$$

where $\omega$ and $\varphi$ are coefficients and $\varepsilon_t$ are random errors.

## 3. Proposed hybrid prediction model

After studying various prediction based models proposed by various researchers, their advantages as well as their limitations, we propose a novel and robust hybrid prediction model (HPM). HPM is formed by merging predictions obtained from linear models and a non-linear model. Emphasis has been put on the method that predictions obtained from HPM are closer to target returns. HPM has been tested on stock data which is non-linear in nature. The strength of HPM lies in the fact that it is able to predict sudden jumps or spikes in data (non-linear patterns), thus improving prediction performance which finally results into least prediction error.

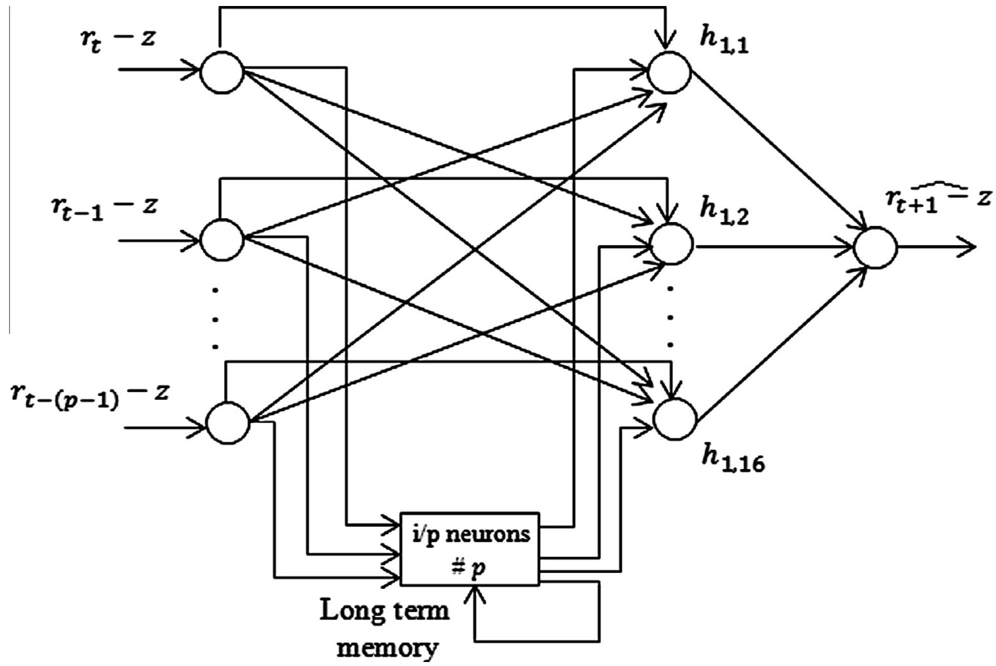


**Fig. 1.** RNN with ARMR.

A new type of regression model called autoregressive moving reference (ARMR) model was proposed by Freitas et al. (2009) and the same model is adopted in this work. For a given time series data, input–output (I–O) pairs are generated using ARMR. In order to obtain predictions, these I–O pairs are given to RNN in a supervised manner. This ANN can be called as AR-MRNN. Using AR-MRNN four different types of ANNs have already been implemented and compared with each other (Rather, 2011). In this work, beside using RNN, predictions are also obtained from ES model and ARMA model. Finally HPM is formed by merging the predictions of three prediction based models i.e. RNN, ES and ARMA. Optimal weights for HPM have been obtained using GA. As compared to linear models, RNN yields much better predictions in terms of lesser prediction error and higher correlation between target returns and predicted returns. However, as desired HPM outperforms RNN thus resulting into more accurate predictions.

## 4. Formulation of proposed hybrid prediction model

Suppose we have a choice of $s$ number of prediction based models. Let $r_t(t = 1, \ldots, T)$ be the actual value at time period $t$ and $\hat{r}_{lt}(l = 1, \ldots, s)$ be predicted value at time $t$ by model $l$. Thus $\varepsilon_{lt} = r_t - \hat{r}_{lt}$ can be corresponding prediction error. Let the weight vector is $W = [w_1, \ldots, w_s]$. The final predicted value of HPM is obtained as shown in the following equation.

$$\hat{r}_t = \sum_{i=1}^{s} w_i \hat{r}_{it} \quad (t = 1, \ldots, T) \tag{9}$$

The sum of weights must be equal to 1 as shown:

$$\sum_{i=1}^{s} w_i = 1 \tag{10}$$

The prediction error of HPM is calculated as shown:

$$\varepsilon_t = r_t - \hat{r}_t \tag{11}$$

Since HPM is obtained by combining three different prediction based models, therefore $s = 3$. HPM is thus obtained as shown below:

$$\hat{r}_{hybrid(t)} = w_1 \hat{r}_{AR-MRNN(t)} + w_2 \hat{r}_{ARMA(t)} + w_3 \hat{r}_{ES(t)} \quad (t = 1, \ldots, T) \tag{12}$$

where $\hat{r}_{hybrid(t)}$ is prediction of HPM at time period $t$; $\hat{r}_{AR-MRNN(t)}$ is prediction of $AR - MRNN$ at time period $t$; $\hat{r}_{ARMA(t)}$ is prediction obtained from ARMA model at time period $t$; $\hat{r}_{ES(t)}$ is prediction obtained from ES model at time period $t$; $\sum_{i=1}^{3} w_i$ and $0 \leqslant w_1 \leqslant 1$.

The smoothing factor ($\alpha$) of ES model is obtained from the following optimization model:

$$Min \quad \frac{\sum_{t=1}^{T} (r_t - \hat{r}_{ES(t)})^2}{T} \tag{13}$$

where $\hat{r}_{ES(t)} = \hat{r}_t + \alpha(r_t - \hat{r}_t)$

$$s.t \quad 0 \leqslant \alpha \leqslant 1 \tag{14}$$

where $r_t$ is actual value; $\hat{r}_{ES(t)}$ is predicted value; $\alpha$ is smoothing factor; $T$ is number of past historical series. The smoothing factor $\alpha$ is associated with the term $\hat{r}_{ES(t)}$. Thus Eq. (13) is an objective function which minimizes mean square error (MSE) of the predictions obtained from ES model.

### 4.1. Weights of a hybrid prediction model

Optimal weights of HPM are obtained from an optimization model which minimizes MSE between target returns and predicted returns as shown:

$$Min \quad \frac{\sum_{t=1}^{T} (r_t - \hat{r}_{hybrid(t)})^2}{T} \tag{15}$$

where $\hat{r}_{hybrid(t)} = w_1 \hat{r}_{AR-MRNN(t)} + w_2 \hat{r}_{ARMA(t)} + w_3 \hat{r}_{ES(t)}$

$$s.t. \quad w_1 + w_2 + w_3 = 1 \tag{16}$$

$$0 \leqslant w_i \leqslant 1 \quad (i = 1, \ldots, 3) \tag{17}$$

Eq. (15) is an objective function which minimizes MSE between target returns ($r_t$) and predicted returns ($\hat{r}_{hybrid(t)}$), of HPM; Eq. (16) ensures that sum of weights associated with three corresponding prediction based models equals to 1; Eq. (17) ensures that weights range between 0 and 1.

*4.1.1. Genetic algorithms representation*

The optimization model given in Eq. (15) to Eq. (17) is solved using GA so as to obtain optimal weights. The population size is chosen as 20 for every stock. The probability of crossover is kept equal to 0.80 and the probability of mutation equal to 0.010. On average it took over 100 generations (for each stock) for GA to arrive at the solution.

## 5. Algorithm for proposed hybrid prediction model

Fig. 2 shows the general diagram of proposed HPM. It is assumed here that time series data is given. The process of hybrid prediction is happening in the dotted box. As shown in the figure there are $E$ number of different linear prediction models $(L_1, L_2, \ldots, L_E)$ generating $E$ corresponding predictions $(\hat{r}_{L_1}, \hat{r}_{L_2}, \ldots, \hat{r}_{L_E})$. Similarly there are $F$ number of different non-linear models $(NL_1, NL_2, \ldots, NL_F)$ which generate $F$ corresponding predictions $(\hat{r}_{NL_1}, \hat{r}_{NL_2}, \ldots, \hat{r}_{NL_F})$. In order to integrate these different prediction based models, optimal weights have to be generated. As shown, each model is associated with its corresponding weights. Linear models have weights as: $w_{L_1}, w_{L_2}, \ldots, w_{L_E}$ and non-linear models have weights as: $w_{NL_1}, w_{NL_2}, \ldots, w_{NL_F}$. After integrating these different models, HPM is produced which yields final prediction. The predictive performance of linear models, RNN as well as of proposed HPM are measured using two error metrics i.e. mean square error and mean absolute error (MAE). Beside these two error metrics, correlation coefficient $(\rho)$ between target returns and predicted returns have also been calculated.

The algorithm for proposed HPM is given below.

---

**Algorithm 1.** Hybrid prediction

---

**Input**: Target data $r_t (t = 1, \ldots, T)$
**Output**: Final predicted value $\hat{r}_t (t = 1, \ldots, T)$ of proposed HPM
**for** $i = 1$ to $E$ **do**
  Find predictions $\hat{r}_{t,L_i}$ from linear prediction model, $L_i$
**for** $j = 1$ to $F$ **do**
  Find predictions $\hat{r}_{t,NL_j}$ from non-linear prediction model, $NL_j$
**Compute predictions** $\hat{r}_t$:
  Compute weights $W_{L_{it}}, i = 1$ to $E, t = 1$ to $T$
  Compute weights $W_{NL_{jt}}, j = 1$ to $F, t = 1$ to $T$
   Using optimization criteria

$$\text{Min} \quad \frac{\sum_{t=1}^{T}(r_t - \hat{r}_t)^2}{T}, \text{ where}$$
$$\hat{r}_t = \sum_{i=1}^{E} W_{L_{it}} \hat{r}_{t,L_i} + \sum_{j=1}^{F} W_{NL_{jt}} \hat{r}_{t,NL_j}$$
s.t.
$$\sum_{i=1}^{E} W_{L_{it}} + \sum_{j=1}^{F} W_{NL_{jt}} = 1$$
$$0 \leqslant W_{L_{it}} \leqslant 1, 1 \leqslant i \leqslant E$$
$$0 \leqslant W_{NL_{jt}} \leqslant 1, 1 \leqslant j \leqslant F$$

**Calculate final predictions from HPM**
$$\hat{r}_t = \sum_{i=1}^{E} W_{L_{it}} \hat{r}_{t,L_i} + \sum_{j=1}^{F} W_{NL_{jt}} \hat{r}_{t,NL_j}$$

---

## 6. Experiments

In order to verify the performance of proposed HPM, set of experiments have been carried out on real stock data. The data of six stocks, shown in Table 1 was obtained from National stock exchange of India (NSE) (http://www.nseindia.com/). Each stock was considered between 02-01-2007 and 22-03-2010 and weekly returns of 164 weeks have been calculated since 08-01-2007 to 22-03-2010. Mean and standard deviation of each stock is calculated as shown in Table 1.

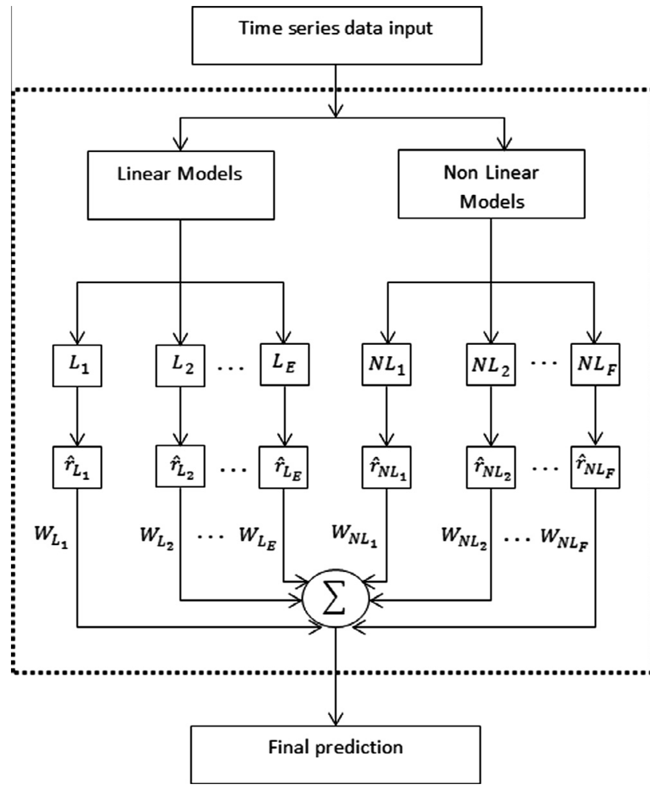


**Fig. 2.** Process of HPM.

**Table 1**
List of six stocks.

| Stock $i$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Stock | TCS | BHEL | Wipro | Axis Bank | Maruti | Tata Steel |
| $\mu_i$ | 0.0004 | 0.0039 | 0.0196 | 0.0111 | 0.0044 | 0.0134 |
| $\sigma_i$ | 0.0666 | 0.0762 | 0.2876 | 0.0861 | 0.0617 | 0.1670 |

### 6.1. Prediction of stock returns using linear models

Using linear models, predictions are obtained from ES model and $AR(6)$ model.

### 6.2. Prediction of stock returns using non-linear model

ARMR model has been used to calculate I–O pairs at different regression orders. These I–O pairs were given to RNN and training of RNN happens in supervised manner. The output from RNN was taken and prediction error calculated. I–O pairs have been formed at regression orders $p = 4, p = 5$ and $p = 6$ and delay from the point of reference i.e. $k = 1$ has been kept constant in all cases. Thus we have the following $AR - MRNNs$:

$$AR - MRNN(4, 1), AR - MRNN(5, 1) \text{ and } AR - MRNN(6, 1).$$

Data was divided into two segments of equal size, i.e. 50:50. 50% data was used for training RNN and rest 50% for testing. 82 returns (08-01-2007 to 28-07-2008) from each stock was used as training data for RNN and remaining 82 returns (04-08-2008 to 29-03-2010) for testing. Proposed HPM integrates predictions (linear and non-linear) for test period only. I–O to RNN was given in the form of sliding windows. In each window, $AR - MRNN(p, 1)$ regression was performed. Table 2 shows I–O pairs in each sliding window and total number of I–O pairs corresponding to each regression order. For each stock, 83 predictions were obtained and future period $(r_{t+1})$ was given by initial window. For the sake

**Table 2**
Input–output pairs.

| $p$ | I–O pairs in each window | Total I–O pairs |
|---|---|---|
| 4 | 78 | 6474 |
| 5 | 77 | 6391 |
| 6 | 76 | 6308 |

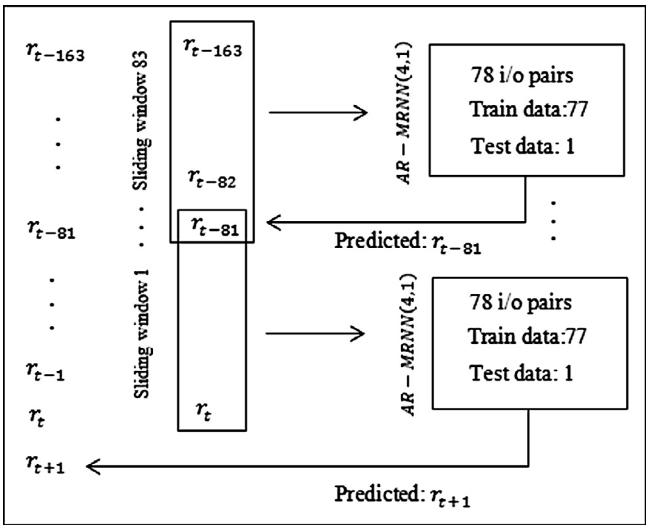


**Fig. 3.** Sliding windows formation.

**Table 3**
Performance metrics of linear models.

| Stock $i$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| *AR*(6) | | | | | | |
| MSE | 0.0044 | 0.0058 | 0.0802 | 0.0071 | 0.0037 | 0.0276 |
| MAE | 0.0443 | 0.0527 | 0.0715 | 0.0638 | 0.0474 | 0.0830 |
| $\rho$ | 0.1708 | 0.1304 | 0.2331 | 0.2158 | 0.2037 | 0.1752 |
| *ES model* | | | | | | |
| MSE | 0.0047 | 0.0059 | 0.0830 | 0.0077 | 0.0039 | 0.0279 |
| MAE | 0.0460 | 0.0536 | 0.0819 | 0.0655 | 0.0478 | 0.0815 |
| $\rho$ | −0.0645 | −0.0554 | −0.1486 | 0.0094 | 0.0384 | −0.0505 |

on understanding, Fig. 3 shows formation of sliding windows for $AR-MRNN(4,1)$. As shown, sliding window 83 gives prediction for $r_{t-82}$, thus $r_{t-81}$ is predicted. Finally prediction for future period ($r_{t+1}$) is given by initial window.

#### 6.2.1. Learning rate and momentum

Learning rate for RNN has been kept as 0.5 and momentum as 0.3. Another important factor is average error or threshold for training data, which has been preset as 0.0002. A network converges only once the error reaches below the chosen threshold. RNN is trained separately for every stock, which implies that for six stocks, RNN may require different number of epochs and computation time to reach below pre-set error threshold.

## 7. Results

Results of linear models are shown initially followed by the results of RNN and finally results of proposed HPM are shown.

**Table 4**
Performance metrics of RNN.

| Stock $i$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| *Training data* | | | | | | |
| MSE | 0.0010 | 0.0010 | 0.0140 | 0.0000 | 0.0000 | 0.0030 |
| MAE | 0.0140 | 0.0220 | 0.0700 | 0.0080 | 0.0140 | 0.0350 |
| $\rho$ | 0.9790 | 0.9860 | 0.9480 | 0.9900 | 0.9870 | 0.9640 |
| *Test data* | | | | | | |
| MSE | 0.0005 | 0.0007 | 0.0113 | 0.0001 | 0.0003 | 0.0026 |
| MAE | 0.0120 | 0.0236 | 0.0731 | 0.0068 | 0.0128 | 0.0354 |
| $\rho$ | 0.9649 | 0.9457 | 0.9658 | 0.9933 | 0.9755 | 0.9780 |

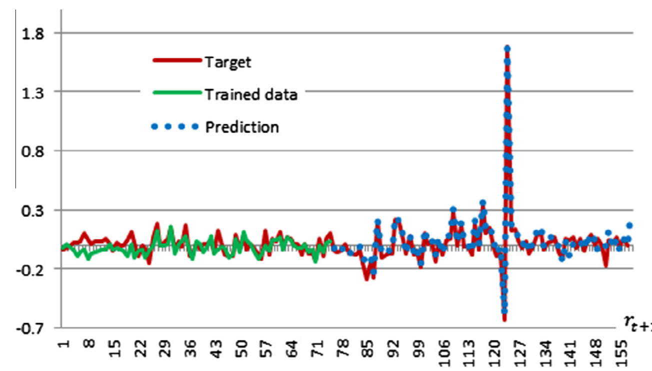


**Fig. 4.** Prediction output of RNN for stock 6.

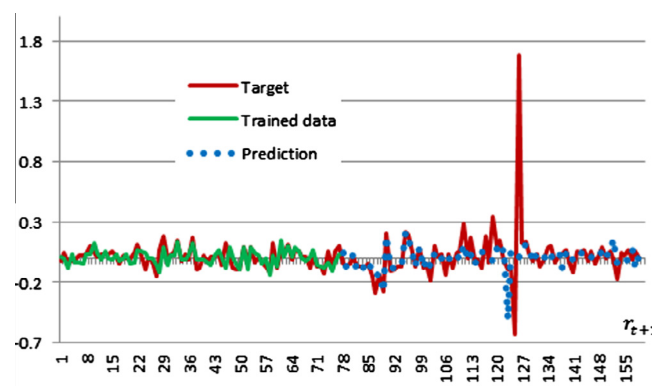


**Fig. 5.** Prediction output of MLP for stock 6.

### 7.1. Results of linear models

The performance metrics of both linear models is shown in Table 3. As seen, MSE and MAE of both models is high. The correlation between target and predicted returns is also very low which implies that predictions obtained from linear models are not satisfactory.

### 7.2. Results of non-linear model

The results of RNN ($AR-MRNN(6,1)$) are shown here. As compared to rest of the two regression orders, accurate predictions were obtained when this particular regression order was used. The performance of RNN with respect to all six stocks can be better judged by observing Table 4, which shows the values of performance metrics for both training as well as for test data. There is less error for each stock and higher correlation coefficient between target and predicted returns.

**Table 5**
A-MSE and A-MAE of RNN.

| $p$ | A-MSE | A-MAE |
|---|---|---|
| *Training data* | | |
| 4 | 0.0035 | 0.0293 |
| 5 | 0.0035 | 0.0328 |
| 6 | 0.0032 | 0.0272 |
| Min | 0.0032 | 0.0272 |
| *Test data* | | |
| 4 | 0.0069 | 0.0582 |
| 5 | 0.0037 | 0.0354 |
| 6 | 0.0026 | 0.0273 |
| Min | 0.0026 | 0.0273 |

**Table 6**
Performance metrics of HPM.

| Stock $i$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| MSE | 0.0001 | 0.0002 | 0.0040 | 0.0001 | 0.0002 | 0.0010 |
| MAE | 0.0077 | 0.0082 | 0.0290 | 0.0052 | 0.0083 | 0.0173 |
| $\rho$ | 0.9913 | 0.9785 | 0.9947 | 0.9977 | 0.9848 | 0.9910 |

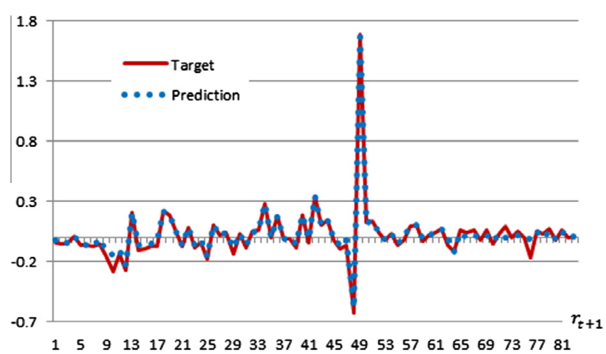


**Fig. 6.** Output of HPM for stock 6.

Fig. 4 displays prediction performance of RNN. Target returns are shown by solid red line, trained data is shown by solid green line and predictions are shown by dotted blue line. The graph shows RNN is able to capture the fluctuations or non-linear patterns, which implies that predictions are satisfactory. This particular stock possesses more noise than other stocks, however RNN is able to predict its patterns very well.

We repeated same set of experiments on same data using simple ANN i.e. multilayer perceptron (MLP) in the ratio of 4:16:4:1 (4 neurons in input layer, 16 neurons in first hidden layer, 4 neurons in second hidden layer and an output neuron). The learning rate is chosen as 0.009 and momentum as 0.95. All neurons in input layer possess linear activation function while as neurons in hidden layer and in output layer possess sigmoid activation function. The same configuration of ANN was chosen by Freitas et al. (2009). Fig. 5 displays the prediction performance of MLP for stock 6. As seen MLP is unable to predict the sudden jump or spike in data, which implies predictions are unsatisfactory. The same behavior of MLP was observed for rest of the stocks.

#### 7.2.1. Average prediction error

Since I–O pairs were formed using three different regression orders, hence it becomes cumbersome to judge the performance of RNN by means of MSE and MAE for every stock. An alternative is to find the average of error metrics used. Table 5 shows average of MSE (A-MSE) and average of MAE (A-MAE) for both training data as well as for test dataset. The average of an error metric is obtained by calculating the error for every stock and finally finding its average. As shown in Table 5, A-MSE and A-MAE is minimum for those input–output pairs which were formed using regression order $p = 6$. This is true for both training data and as well as for test data.

### 7.3. Results of proposed hybrid prediction model

Using proposed HPM, predictions of six stocks have been obtained for the same time period as obtained using RNN. Table 6 shows performance metrics of HPM for six stocks. As per the results given in Table 5, HPM has outperformed RNN with least prediction error.

Fig. 6 shows the prediction performance of proposed HPM for stock 6. As expected, HPM is able to capture the sudden jump in the data which most of the models fail to capture. Thus, prediction error observed in this case is lowest so far. The correlation between target and predicted returns is also highest for all stocks than observed before. A-MSE and A-MAE of six stocks is calculated as 0.0009 and 0.0127 respectively, which is lowest than observed previously.

### 7.4. Further tests of hybrid prediction model

In order to verify the robustness of HPM, a large set of experiments are again performed with more number of stocks. The daily closing price of 25 stocks (including international stocks) starting from 14-05-2013 till 30-12-2013 has been obtained from Bombay stock exchange (BSE) (http://www.bseindia.com/). Daily returns have been calculated between15-05-2013 and 30-12-2013. As shown in Table 7, list of stocks have been selected sectorwise, having five stocks in each sector. Stock 1 to stock 5 belong to information technology sector; stock 6 to stock 10 belong to banking sector; stock 11 to stock 15 belong to automobile sector; stock 16 to stock 20 belong to pharmaceutical sector and finally stock 21 to stock 25 belong to electrical appliances sector.

As already done in previous set of experiments, the same procedure was repeated again. $AR - MRNN(6, 1)$ was used on RNN as it was proved to be advantageous in previous set of experiments. The predictions from proposed HPM were obtained between 06-09-2013 and 31-12-2013. Table 8 shows the performance metrics of RNN (test data) as well as that of proposed HPM model.

A-MSE and A-MAE of 25 stocks for RNN is calculated as 0.0028 and 0.0107 respectively. For HPM, A-MSE and A-MAE is observed as 0.0013 and 0.0047 respectively. Undoubtedly RNN yielded much better predictions than linear models. As expected, proposed HPM

**Table 7**
25 stocks listed in BSE.

| | | | | |
|---|---|---|---|---|
| 1. Infosys | 6. IDBI Bank | 11. Bajaj Auto | 16. GlaxoSmithKline | 21. Whirlpool |
| 2. Tech Mahindra | 7. Bank of India | 12. Maruti Suzuki | 17. Cipla | 22. Videocon |
| 3. HCl | 8. Federal Bank | 13. Tata Motors | 18. Sun Pharma | 23. Bajal Electricals |
| 4. Mind Tree | 9. J&K Bank | 14. TVS Motors | 19. Lupin | 24. Hitachi Home |
| 5. Mphasis | 10. ICICI Bank | 15. Eicher Motors | 20. Glenmark | 25. Blue Star |

**Table 8**
Performance metrics output of 25 stocks.

| Stock | RNN | | | HPM | | |
|---|---|---|---|---|---|---|
| | MSE | MAE | $\rho$ | MSE | MAE | $\rho$ |
| 1 | 0.0000 | 0.0042 | 0.8823 | 0.0000 | 0.0024 | 0.8824 |
| 2 | 0.0000 | 0.0027 | 0.9649 | 0.0000 | 0.0017 | 0.9762 |
| 3 | 0.0000 | 0.0043 | 0.9740 | 0.0000 | 0.0021 | 0.9900 |
| 4 | 0.0000 | 0.0035 | 0.9712 | 0.0000 | 0.0044 | 0.9766 |
| 5 | 0.0000 | 0.0041 | 0.9156 | 0.0000 | 0.0026 | 0.9743 |
| 6 | 0.0001 | 0.0030 | 0.9301 | 0.0000 | 0.0033 | 0.9579 |
| 7 | 0.0001 | 0.0056 | 0.9768 | 0.0000 | 0.0043 | 0.9882 |
| 8 | 0.0674 | 0.1580 | 0.8247 | 0.0318 | 0.0398 | 0.9899 |
| 9 | 0.0000 | 0.0027 | 0.9910 | 0.0000 | 0.0040 | 0.9814 |
| 10 | 0.0000 | 0.0031 | 0.9847 | 0.0000 | 0.0018 | 0.9937 |
| 11 | 0.0000 | 0.0017 | 0.9881 | 0.0000 | 0.0030 | 0.9800 |
| 12 | 0.0000 | 0.0036 | 0.9610 | 0.0000 | 0.0023 | 0.9834 |
| 13 | 0.0000 | 0.0023 | 0.9751 | 0.0000 | 0.0024 | 0.9809 |
| 14 | 0.0001 | 0.0092 | 0.9686 | 0.0001 | 0.0083 | 0.9469 |
| 15 | 0.0000 | 0.0019 | 0.9857 | 0.0000 | 0.0023 | 0.9885 |
| 16 | 0.0002 | 0.0090 | 0.8114 | 0.0001 | 0.0032 | 0.9514 |
| 17 | 0.0000 | 0.0025 | 0.9450 | 0.0000 | 0.0011 | 0.9835 |
| 18 | 0.0006 | 0.0189 | 0.1894 | 0.0002 | 0.0088 | 0.6747 |
| 19 | 0.0001 | 0.0055 | 0.8952 | 0.0000 | 0.0018 | 0.9674 |
| 20 | 0.0000 | 0.0023 | 0.9651 | 0.0000 | 0.0042 | 0.9703 |
| 21 | 0.0000 | 0.0021 | 0.9539 | 0.0000 | 0.0035 | 0.9778 |
| 22 | 0.0000 | 0.0038 | 0.9083 | 0.0000 | 0.0020 | 0.9586 |
| 23 | 0.0001 | 0.0038 | 0.9000 | 0.0000 | 0.0020 | 0.9726 |
| 24 | 0.0002 | 0.0069 | 0.9482 | 0.0000 | 0.0027 | 0.9909 |
| 25 | 0.0000 | 0.0032 | 0.9104 | 0.0000 | 0.0027 | 0.9327 |

model has again outperfomed RNN, in terms of least prediction error and higher correlation between target and predicted returns. However it is important to mention that the prediction performance of proposed HPM is mainly contributed by RNN. RNN performs better because of its chosen structure and also because of the new regression model.

## 8. Conclusions

We achieved higher degree of accuracy in our proposed model because of several reasons. The first reason is the unique regression model that calculates a new differenced time series data from original series and also calculates a future prediction in every sliding window. The second reason is chosen RNN that has been trained on differenced series. With the result, training of RNN enhances as it needs to search for smaller weights. The third reason is an optimization model introduced, which minimizes prediction error. All these methods together contribute in obtaining excellent predictions.

The limitation of this work is that the proposed model does not guarantee excellent predictions on all data sets, rather the model is data dependent. Another limitation of this work is that the optimal regression order for ARMR model is unknown; regression order has to be chosen by trial and error.

The future work involves to include newly developed soft computing models such as extreme learning machines and particle swarm optimization in hybrid system. The future work also involves to minimize computation time of RNN. The proposed model can be further enhanced by testing it in different areas such as engineering sciences, exchange rate risk etc. This is certainly an important avenue for future research.

## Acknowledgements

The work reported here has been supported by School of Computer & Information Sciences, University of Hyderabad and Institute for Development & Research in Banking Technology, Hyderabad. The authors acknowledge anonymous reviewers for their valuable suggestions and comments.

## References

Abu-Mostafa, Y., & Atiya, A. (1996). Introduction to financial forecasting. *Applied Intelligence, 6*, 205–213.
Adhikari, R., & Agrawal, R. (2013). A combination of artificial neural network and random walk models for financial time series forecasting. *Neural Computing and Applications, 22*, 1–9.
Armano, G., Marchesi, M., & Murru, A. (2005). A hybrid genetic-neural architecture for stock indexes forecasting. *Information Sciences, 170*, 3–33.
Box, G., & Jenkins, G. (1970). *Time series analysis, forecasting and control*. San Francisco: Holden-Day.
Brown, R. (2004). *Smoothing, forecasting and prediction of discrete time series*. Mineola, N.Y.: Courier Dover Publications.
Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery, 2*, 121–167.
Chaabane, N. (2014). A hybrid ARFIMA and neural network model for electricity price prediction. *Electric Power and Energy Systems, 55*, 187–194.
Chen, C. (1994). Neural networks for financial market prediction. In *IEEE world congress on computational intelligence* (pp. 1199–1202). Orlando, FL.
Chen, M. (2012). Visualization and dynamic evaluation model of corporate financial structure with self-organizing map and support vector regression. *Applied Soft Computing, 12*, 2274–2288.
Chen, M. (2013). A hybrid ANFIS model for business failure prediction utilizing particle swarm optimization and subtractive clustering. *Information Sciences, 220*, 180–195.
Chen, M., Chen, D. R., Fan, M. H., & Huang, T. Y. (2013). International transmission of stock market movements: an adaptive neuro-fuzzy inference system for analysis of TAIEX forecasting. *Neural Computing and Applications, 23*, 369–378.
Chen, M., Fan, M., Chen, Y., & Wei, H. (2013). Design of experiments on neural network's parameters optimization for time series forecasting in stock markets. *Neural Network World, 23*, 369–393.
Dai, W., Wu, J., & Lu, C. (2012). Combining nonlinear independent component analysis and neural network for the prediction of asian stock market indexes. *Expert Systems with Applications, 39*, 4444–4452.
Durham, G. (2007). Sv mixture models with application to S & P 500 index returns. *Journal of Financial Economics, 85*, 822–856.
Freitas, F., de Souza, A., & de Almeida, A. (2009). Prediction-based portfolio optimization using neural networks. *Neurocomputing, 72*, 2155–2170.
Gen, M., & Cheng, R. (2000). *Genetic algorithms and engineering optimization*. New York: John Wiley and Sons.
Gordini, N. (2014). A genetic algorithm approach for smes bankruptcy prediction: Empirical evidence from Italy. *Expert Systems with Applications, 41*, 6433–6445.
Haykin, S. (1994). *Neural networks: A comprehensive foundation*. Saddle River: Prentice-Hall.
Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence*. Cambridge, MA, USA: MIT Press.
Hsieh, T., Hsiao, H., & Yeh, W. (2011). Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. *Applied Soft Computing, 11*, 2510–2525.
Hsu, V. (2013). A hybrid procedure with feature selection for resolving stock/futures price forecasting problems. *Neural Computing and Applications, 22*, 651–671.
Huang, W., Nakamori, Y., & Wang, S. (2005). Forecasting stock market movement direction with support vector machine. *Computers and Operations Research, 32*, 2513–2522.
Khashei, M., & Bijari, M. (2010). An artificial neural network $(p,d,q)$ model for timeseries forecasting. *Expert Systems with Applications, 37*, 479–489.
Kim, K., & Ahn, H. (2012). Simultaneous optimization of artificial neural networks for financial forecasting. *Applied Intelligence, 36*, 887–898.
Kim, K., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications, 19*, 125–132.
Koza, J. R. (1994). *Genetic programming: On the programming of computers by means of natural selection*. Cambridge, MA, USA: MIT Press.
Kwon, Y., & Moon, B. (2007). A hybrid neurogenetic approach for stock forecasting. *IEEE Transactions on Neural Networks, 18*, 851–864.
Liao, Z., & Wang, J. (2010). Forecasting model of global stock index by stochastic time effective neural network. *Expert Systems with Applications, 37*, 834–841.
Majhi, R., & Panda, G. (2008). Prediction of S & P 500 and DJIA stock indices using particle swarm optimization technique. In *IEEE world congress on computational intelligence* (pp. 1276–1282). Hong Kong.
Rather, A. (2011). A prediction based approach for stock returns using autoregressive neural networks. In *IEEE world congress on information and communication technologies* (pp. 1271–1275). Mumbai.
Rather, A. (2014). A hybrid intelligent method of predicting stock returns. *Advances in Artificial Neural Systems, 2014*.
Rout, M., Majhi, B., Majhi, R., & Panda, G. (2014). Forecasting of currency exchange rates using an adaptive ARMA model with differential evolution based training. *Journal of King Saud University-Computer and Information Sciences, 26*, 7–18.
Rumelhart, D., & McClelland, J. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition*. Cambridge, MA, USA: MIT Press.

Saadaoui, F., & Rabbouch, H. (2014). A wavelet-based multiscale vector-ANN model to predict comovement of econophysical systems. *Expert Systems with Applications, 41*, 6017–6028.

Seker, S., Ayaz, E., & Turkcan, E. (2003). Elman's recurrent neural network applications to condition monitoring in nuclear power plant and rotating machinery. *Engineering Applications of Artificial Intelligence, 16*, 647–656.

Shen, W., Guo, X., Wu, C., & Wu, D. (2011). Forecasting stock indices using radial function neural networks optimized by artificial fish swarm algorithm. *Knowledge-Based Systems, 24*, 378–385.

Smith, C., & Jin, Y. (2014). Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction. *Neurocomputing, 143*, 302–311.

Wang, J.-Z., Wang, J.-J., Zhang, Z.-G., & Guo, S.-P. (2011). Forecasting stock indices with back propagation neural network. *Expert Systems with Applications, 39*, 14346–14355.

Wang, J., Wang, J., Zhang, Z., & Guo, S. (2012). Stock index forecasting based on a hybrid model. *Omega, 40*, 758–766.

Wang, L., Zou, H., Su, J., Li, L., & Chaudhry, S. (2013). An ARIMA-ANN hybrid model for time series forecasting. *Systems Research and Behavioral Science, 30*, 244–259.

Wold, H. (1938). *A study in the analysis of stationary time series*. Stockholm: Almqvist and Wiksells.

Yi, X., Jin, X., John, L., & Shouyang, W. (2014). A multiscale modeling approach incorporating ARIMA and ANNS for financial market volatility forecasting. *Journal of Systems Science and Complexity, 27*, 225–236.

Yule, G. (1926). Why do we sometimes get nonsense correlations between time series? A study in sampling and the nature of time series. *Journal of the Royal Statistical Society, 89*, 30–41.

Zhang, G. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing, 50*, 159–175.