International Journal of Industrial Electronics, Control and Optimization © 2019 IECO
Vol. 2, No. 2, pp. 81-90, April (2019)

Application of Covariance Matrix Adaptation-Evolution Strategy to Optimal Portfolio

S. Amir Ghoreishi[1], Hamid Khaloozadeh[2]

[*,1]Faculty of Electrical and Computer Engineering, Science and Research Branch Islamic Azad University, Tehran,Iran.
[2]Industrial Control Center of Excellence (ICCE), Department of Systems and Control, Faculty of Electrical Engineering, K. N. Toosi University of Technology, Tehran, Iran.

ABSTRACT

Capital portfolio management is considered an important issue in the field of economics and its main subject is about the scientific management of combination choice of assets that meet the specific investment objectives. Maximizing returns and minimizing asset risk are the most important goals in the management of the portfolio of capital. This paper proposes two novel risk measures based on the MLP neural networks and prediction intervals (PI). The MLP based risk is constant and assumes that the uncertainty is uniform in the dataset. The second one is a time varying risk measure that doesn't assume uniformity condition. After introducing two novel risk measures, a new cost function is presented to consider the expected returns and the involving risk at the same time. Finally, the covariance matrix adaptation evolution strategy (CMA-ES) algorithm is used to obtain the optimal portfolio. The validity of the proposed selection process (including risk measures, cost function, and the optimization method) is tested using the dataset of the 18 shares of the Tehran Stock Exchange, and the results are compared with the obtained portfolio using the conditional value at risk (CVaR) criterion as a well-known benchmark.

Article Info

Keywords:
Covariance Matrix Adaptation-Evolution Strategy
Cost Function, Optimal Portfolio, Prediction Price.

Article History:
Received 2018-09-03
Accepted 2018-11-12

I. Introduction

Stock price prediction involved factors such as political events, economic conditions, oil and gold prices, traders' expectations and other environmental factors that can influence stock prices, so it is a challenging problem. Moreover, the stock price series commonly have a noisy, dynamic, non-linear, complex, non-parametric and chaotic nature. So, stock market anticipating is considered as a challenging task in predicting financial time series. Financial time series shows rather complex patterns and such series are often non-stationary, whereby a variable does not have a clear tendency to move towards a linear trend or a constant value.

Static prediction models, containing time series, require a series of basic assumptions for variables (some preconditions are required for conventional statistical prediction models such as specific probability statistical distribution) and produces prediction models based on mathematical equations that are not easily understood by investors [1].

Neural networks are studied as one of the most powerful tools in financial prediction [2-4]. They have been inspired by nature, and are potentially able to model any complex nonlinear function with desired precision (They are general function estimated) [5]. They do not require any presupposition for the characteristics of the system data [6], and are widely used for modeling, classification, forecasting and control [7-9]. In the field of economics, some methods besides the neural networks, have been developed for predicting which in some aspects are competitive to neural

† Faculty of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.
* Corresponding Author: h_khaloozadeh@kntu.ac.ir
Tel: +98 21 84062422, Fax: +98 21 88462066, K.N. Toosi University of Technology.

networks. Agent-Based Modeling (ABM) methods are involved in this category.

The price per share is a function of the supply and demand for that share. Supply, indicating the level of scarcity and demand, reflects the demands and needs of traders. The ABM method is used to model and predict stock prices, to identify different categories of traders, and to analyze and predict the supply and demand of these different categories. In this way, the trader's behavior is modeled using any strategy that the agent is called. After modeling the behavior of all factors in the market, it is possible to estimate the supply and demand for the future, and from this amount of supply and demand, the price can be predicted.

In Ref. [10], the technical and fundamentalist agents are introduced and analyzed. Also, these two agents are used to provide a method to predict stock prices. Ref. [11] by presenting newer models for the mentioned agents, tries to approach the ABM method of [10] to human thinking as much as possible. Ref. [12] has also investigated the fundamentalist agent model in a particular way, reducing its hypotheses, to present a new method for combining the beliefs of the agents with each other. It has shown that the profits obtained with this method are greater than the traditional fundamentalist agents. Ref. [13] specifically addresses the technical agent and, presenting a new nonlinear combining belief model, improves the prediction result than the normal technical agent. The ABM method has been firmly established in its research in the field of economics, and even books have been published in this area [14].

Despite such superiorities, neural networks have two disadvantages. The first is a significant reduction in the prediction accuracy when there is a considerable uncertainty in the system data. Therefore, the reliability of predictions also decreases. Since these uncertainties are not predictable with high precision, the accuracy of the output of the neural networks, which is one-dimensional, is faltered. This weakness of the neural networks is due to the production of the mean value of the target function as output [15]. This problem has been investigated in the load forecasting of the distribution network [16-17], predicting the life expectancy of components [18], the predictions needed to provide financial services [19], the predictions needed in water distribution systems [20], transport systems [21-23] and baggage handling systems [24], have similar problems with conventional neural networks.

The second difficulty of the neural networks is that they generate a single point forecasting output and do not cater any criterion for measurement of its accuracy. It is serious to investigate the uncertainties in single-value and point predictions in order to amend the prediction accuracy and subsequent decision-making process.

To overcome these problems, PI are proposed for the neural networks. A PI is composed of upper and lower limits, which include the value of the target values, with a certain probability of (α-1) percent, called the confidence level. The main motivation of the PIs production is the quantification of uncertainty induced by the point predictions. In many papers, the production of PIs and their combination with the output of neural networks has been addressed [25-35]. Another issue that arises in this area is how to choose a stock portfolio and optimal allocation of capital. Maximizing returns and minimizing asset risk are the most important goals in the management of the portfolio of capital. According to the theory of mean - variance (MV), Markowitz, an investor seeks to maximize returns for a predetermined level of asset risk or minimize risk for a certain level of returns [36-37].

$$Maximize\left(Expected\ return - Risk\right) \quad (1)$$

Any kind of investment is confronted with uncertainties that put the future return of the portfolio at risk. The risk of an asset is due to the return or value of the portfolio which may be less than expected. The mean-risk models were introduced for the first time in the early 1950s to provide practical solutions to the issue of choosing a portfolio of capital. Using these models, the average risk uses only two parameters of average return and risk to describe the distribution [38].

As noted above, stock price forecasting and the presentation of a suitable risk model are two important issues that must be addressed before the optimal portfolio selection. This paper investigates the risk measures. In this regard, two new risk metrics are proposed using MLP neural networks with differential output and interval prediction to model the uncertainty of the expected return on the allocation of capital. The first risk metric is based on the normalized average of square error of the predicted neural network for stock price prediction. In this criterion, it is assumed that the uncertainties uniformly affect the data and are not variable with time. In the second criterion, the width of PIs that are trained with the stock data are used. In this viewpoint, there is no assumption of uniformity for the uncertainty of the data. A novel cost function was presented to simultaneously consider the expected returns and risks. This solves using CMAES Algorithm by applying a step-by-step algorithm to facilitate the solving process.

The remainder of this article is as follows: In Section 2, the necessary preliminaries for making PI and the training process different from [39] are explained. In Section 3, the prediction stages with the aid of a different neural network to model the problem of capital allocation and its solving method will be explained. In Section 4, the algorithm presented in Section 3 is implemented for 18 selected shares of the Tehran Stock Exchange (TSE), and the efficiency of the two risk measures introduced, are measured in contrast with the CVaR, of which the results indicate that the risk measures are appropriate.

As mentioned before, the main advantage of the ABM method is the proximity of its logic to human thinking. On the other, neural networks have a great ability to model complex nonlinear functions as the stock price trending function. Therefore, comparing the proposed method for the stock price prediction to the ABM method can be a good benchmark.

II. NECESSARY PREPARATIONS FOR PI

The method of [39] has been applied to train the neural network for achieving PI. This method is called the Lower Upper Bound Estimation (LUBE). The advantage of this method is avoidance of using the output derivative of the neural network with respect to its parameters that can cause biases in the prediction process [34].

The aim is obtaining PIs with the minimum possible width and the acceptable coverage probability in the LUBE method. A neural network is utilized for the estimate of upper and lower bound in the production of PIs [24]. One of the most important criteria in the production of PI is the level of coverage$(1-\alpha)$, where$(\alpha<1)$. This measure indicates the percentages of the system output included by PI. The level of confidence is also known by the coverage percentage of PI or PICP. It can be represented by [24, 21, and 16]:

$$PICP=\frac{1}{n}\sum_{i=1}^{n}c_i \qquad (2)$$

Where n is the number of data samples and c_i will be equal to 1 if i'th data sample is inside the PI band, and otherwise is zero.

$$L(x_i)\le y_i\le U(x_i) \qquad (3)$$

Besides that $L(x_i)$ and $U(x_i)$ are lower and upper bounds of PI for i'th data sample respectively. A Schematic of a PI-based neural network is presented In Fig. 1. As mentioned before, only the PICP reaching one is not enough in the evaluation of the performance of the PI. Besides that the PI bandwidth should be as small as possible. The prediction interval normalized average width (PINAW) is considered for this intent:

$$PINAW=\frac{1}{nR}\sum_{i=1}^{n}(U_i-L_i). \qquad (4)$$

Where R is the maximum value in difference between the upper and lower bounds of PIs. The parameter R is used to normalize PINAW magnitude between zero and one. This process is done to make PINAW comparable respect to the PICP measure. The trend of PINAW and PICP has the same behavior.

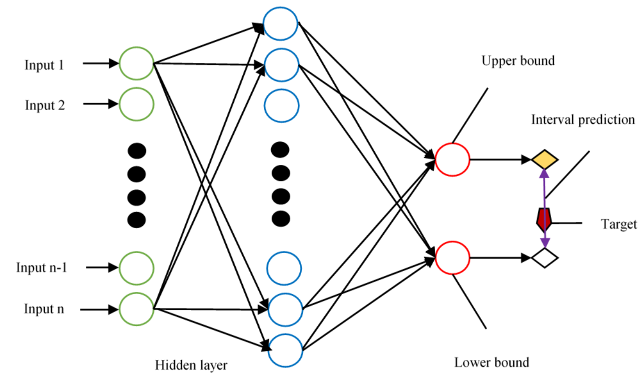

Fig. 1. A schematic of a PI-based neural network.

So, the higher the value of PINAW grows PICP will approach to one. PINAW can be replaced with Prediction Interval Normalized Root-mean-square Width (PINRW), because the neural networks are trained using square measures usually. This criteria is defined as:

$$PINRW=\frac{1}{R}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(U_i-L_i)^2} \qquad (5)$$

PINRW intensifies the bigger interval bandwidth influence. So, it is expected that the produced PIs using PINRW be narrower than the PINAW produced ones. As mentioned before, narrower PIs with a high coverage percentage are desired. Besides that, decreasing the width of the PIs will result in decreasing the coverage percentage. So, we should balance them in a reasonable manner. This result states that, the PI production process can be considered as a multi-objective optimization problem:

$$\begin{aligned}&\left(\max_{\omega} PICP\right)\&\left(\min_{\omega} PINRW\right)\\&s.t.\left(0<PICP\le 1\right)\&\left(PINRW>0\right)\end{aligned} \qquad (6)$$

The optimal weights of the neural network (W*) in (6) should be determined such that the PICP and PINRW are as high as possible and as small as possible respectively. The solution of (6) is tedious and many of the Pareto's answers aren't acceptable due to low PICP values that reduces the reliability of PI [39]. The minimum required PICP can be modelled as a constraint and the optimization problem will change to a single objective optimization problem:

$$\begin{aligned}&\left(\min_{\omega} PINRW\right)\\&s.t.\left(\mu\le PICP\le 1\right)\&\left(PINRW>0\right)\end{aligned} \qquad (7)$$

The optimal weights of the neural network (W*) in (7) are determined to minimize the PINRW. μ is the minimum acceptable level of PICP. It is worth noting that the single-objective optimization problem is easier than a multi-objective problem to solve. Finally, the mentioned optimization problem will be solved by the PSO algorithm.

III. LUBE METHOD

The traditional methods for producing neural network based PIs include following steps [40]:

- Neural networks are trained to minimize the estimation error.
- Assuming a specific data distribution, the mean and variance values are calculated then, the Hessian and Jacobian matrices of the neural network coefficients are also created. This process results in PIs creation.

These building PIs methods suffer from multiple issues. For instance, the Delta method assumes data and residuals are Gaussian distributed. Also, both the Delta and Bayesian methods need the Jacobin and Hessian matrices calculation. This may lead to the problems of accuracy of matrixes and reliability decay of PIs. Besides that, the calculation of these matrices is time consuming and greatly increases the implementation difficulty of these methods. The LUBE method has been proposed in [41] for the first time. A two-output neural network is trained to build PIs in one step, without any assumption on data distribution in this method (these two outputs are related to the lower and upper bounds of PI). The LUBE algorithm can be expressed as:

1) Neural networks are trained to minimize the estimation error. The available data are divided into 3 groups: train, evaluation and test. The train data is used to set the parameters of the neural network. The evaluation dataset is used to specify the optimal structure of the neural network (the number of hidden layers, the number of neurons in each layer). Finally, the test data is used to evaluate the performance of the trained neural network.

2) Since the stock market data is not stationary, they will be differences. According to the proposed method [42], this operation is done only for the output of the neural network.

$$x_{t+n} = \Delta x_{t+n} + x_t \quad (8)$$

In (8), Δx_{t+n} is the stationary dataset gained by differencing.

3) Next, the optimal structure of the neural network will be assigned. The training is repeated to a certain limit (in this paper, the training is repeated 3 times) for each candidate of the neural network structure. After that, the PICP and PINRW of each trained neural network are calculated using the evaluation data. The optimal structure of the neural network is selected within the trained networks finally. In this paper, despite of [39], the neural networks with larger cost functions J are chosen. The cost function J of each candidate of the neural network is the mean of the 3 neural networks cost functions.

$$J = \frac{PICP}{PINRW} \quad (9)$$

The PI will be chosen with a higher PICP and a lower PINRW using this criterion. It is worth noting that this criterion is calculated only for the cases of PICP> μ.

4) The main stage of PSO algorithm is updating the particle velocity and position values. This process is done as follows [43]:

$$v_n(t+1) = Wv_n(t) + c_1 rand(.)\left(p_{best,n} - x_n(t)\right) + c_2 rand(.)\left(g_{best,n} - x_n(t)\right) \quad (10)$$

$$x_n(t+1) = x_n(t) + v_n(t+1) \quad (11)$$

Where v_n and rand (.) are the velocity of the particle in the n-th dimension, and a random number between 0 and 1 respectively. Also W, C_1 and C_2 are is a weighting factor and coefficients that determine the relative importance between P_{best} and g_{best} [44] respectively. Besides that the maximum allowable particle velocity and position are determined by V_{max} and X_{max}.

5) If the above algorithm is performed a certain number of times, or the produced PI is satisfying, go to the next step, and otherwise return to step 4.

6) After the PI training, g_{best} will be used as the neural network weights. Finally, the PICP and PINRW values are reported.

IV. THE PROPOSED METHOD

In this section, the process including the prediction of stock prices to the optimal allocation of the portfolio is explained.

A. *Stock Price Prediction*

The MLP neural networks are applied to predict 'n' future days of stock prices (n = 5 is taken in this paper). It is well known that the performance the neural networks are satisfied in the case of stationary dataset. This neural network model can be expressed as:

$$\hat{x}_{t+n} = f\left(x_t, x_{t-1}, x_{t-2}, \ldots, x_{t-p}\right) + e_{t+n} \quad (12)$$

Where x_{t-i} is the stock price data of i days before and e_{t+n} is the estimation error of x_{t+n}. As mentioned earlier [45-46], stock prices data do not have stationary behavior. So the neural networks will not succeed in the prediction operation, unless the data transform to be stationary. Differentiating is the one of the approaches making data samples stationary. The data with n index difference are subtracted from each other due to the prediction of n forward samples:

$$\Delta\hat{x}_{t+n} = f\left(x_t, x_{t-1}, x_{t-2}, \ldots, x_{t-p}\right) + e_{t+n} \quad , \quad \hat{x}_{t+n} = \Delta\hat{x}_{t+n} + x_t \quad (13)$$

x_{t+n} and x_t samples have a high correlation for small and medium n values due to non-stationary nature of the dataset. So Δx_{t+n} and consequently $\Delta\hat{x}_{t+n}$ will be small numbers [43]. Thus the problem of non-stationary property of the dataset in prediction will be greatly resolved.

$$\Delta\hat{x}_{t+5} \approx f\left(x_t, x_{t-1}, x_{t-2}, \ldots, x_{t-p}\right) \quad , \quad \hat{x}_{t+5} = \Delta\hat{x}_{t+5} + x_t \quad (14)$$

After training neural networks using well known algorithms as gradient descent and assign the optimal structure, the neural networks' output and the price of today are used to estimate the price of n days later. After that, the expected returns can be resulted as follows:

$$r_{t+n} = \frac{\hat{x}_{t+n} - x_t}{x_t} = \frac{\Delta \hat{x}_{t+n}}{x_t} \tag{15}$$

B. Expected Return Risk Estimation

Since the uncertainty is involved in the dataset, it is necessary to incorporate the expected predictions with these uncertainties. The two viewpoints are considered in this for this problem:

1) Taking the variance of the estimation error of the previous mentioned differential neural network as a constant risk:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \hat{x}_i\right)^2 \tag{16}$$

2) Taking PI based neural network as the time varying risk of each share $U_i - L_i$. Since the PIs are produced for the uncertainty involvement in system data, it is expected that the PI based risk would be more realistic and will increase returns with respect to the previous case.

C. Cost Function and Optimization of Portfolio

The expected returns have been introduced and the related risks are assigned. Now, the cost function will be introduced to solve the portfolio optimization problem. The expected returns and related risk will be simultaneously considered in a cost function as below to make the process of the portfolio allocation easier:

$$J = \sum_{i=1}^{m} \begin{pmatrix} w_i(t+n)\, r_i(t+n) - \\ \left[w_i(t+n) - w_i(t)\right]. \\ tr\left(w_i(t+n) - w_i(t)\right) \end{pmatrix} \exp\left(-\sigma_i(t+n)\right) \tag{17}$$

$$tr(x) = sgn(x)\left(\frac{bp+sp}{2}\right) + \frac{bp-sp}{2}, \qquad \sum_{i=1}^{m} w_i(t+n) = \sum_{i=1}^{m} w_i(t) = 1 \tag{18}$$

Where $w_i(t)$ is the weight of the i-th capital in the portfolio at t time instant. m is the number of shares in the stock portfolio. $r_i(t+n)$ and $\sigma_i(t+n)$ are normalized expected return and the related risk of the i-th share respectively. A penalty is also incorporated for the transaction costs. bp and sp are buy and sale penalties respectively. The function $tr(x)$ has been proposed for the easier involvement of these two different penalties.

The portfolio selection optimization problem will be solved by an evolutionary search algorithm as CMA-ES at the next stage. A strategy called step-by-step is used due to time consuming solve process optimization problem. According to this strategy, we only buy (sell) a share that has a higher price growth (decline) than a predefined threshold (this is the desired value that is taken 5% here). So, the stock will not take part in the optimization process if its price is almost constant. This strategy is presented in (19):

$$\begin{cases} if \quad \dfrac{\hat{x}_{t+n} - x_t}{x_t} \gg 0.05 \rightarrow x_t \ \textit{selected for optimization} \\ other \rightarrow \ x_t \ \textit{not selected for optimization} \end{cases} \tag{19}$$

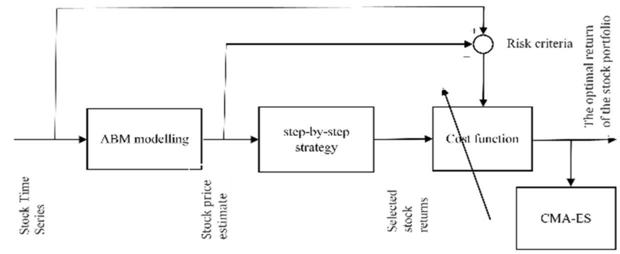

Fig. 2. Diagram of the proposed method for assigning the optimal portfolio.

D. CMA-ES:

The covariance matrix adaptation evolution strategy (CMA-ES) enumerates amongst the most efficacious evolutionary methods for continuous parameter optimization. This scheme was proposed by [43] and further advanced in [44, 45]. The CMA-ES is rather intricate and includes a number of free parameters which have to be set with no or little theoretical guidance. It is important that the CMA-ES does not work well for large population sizes. The application of the CMA-ES algorithm is useful. Consider the objective function $f : IR^n \rightarrow IR$ and there is a sequence of $x(t)$, $t = 1, 2, ... \in IR^n$ which minimize $f(x(t))$.

$$f : x \subseteq IR^n \rightarrow R \qquad x \rightarrow f(x) \tag{20}$$

f as objective function is non-linear, non-separable, non-convex and non-smooth. The aim is to find the new point x that minimizes the cost function. This novel search point is sampled normally distributed as:

$$x_i \sim m + \sigma N_i(0, C) \qquad i = 1, 2, ..., \lambda \tag{21}$$

Where $m \in IR^n$, $\sigma \in IR_+$ and $C \in IR^{n \times n}$. The m is the mean vector characterizes the favorite solution. Step size σ controls the step length. Also, C is covariance matrix which shows the shape of the distribution ellipsoid. This matrix of the distribution is updated such that the likelihood of previously successful search steps is increased and followed by:

$$\{x \in IR^n \mid x^T C^{-1} x = 1\} \tag{22}$$

The candidate solutions x_i are evaluated on the objective function to be minimized:

$$x_{i:\lambda} \rightarrow f(x_i) \qquad i = 1, 2, ..., \lambda \tag{23}$$

The results are sorted via fitness and then weighted mean is calculated as written below:

$$x_{i:\lambda} \rightarrow sort(x_{i:\lambda})$$
$$m_{k+1} = m_k + \sum_{i=1}^{\mu} w_i (x_{i:\lambda} - m_k) \qquad k = 1,2,...,K \tag{24}$$

Where k is number of iteration. $\mu \le \lambda/2$ and $w_i = 1,2,...,\mu$ is positive weights as:

$$w_1 \ge w_2 \ge ... \ge w_\mu > 0$$
$$\sum_{i=1}^{\mu} w_i = 1 \tag{25}$$

The step-size σ_k is updated using cumulative step-size adaptation (CSA), sometimes also indicated as path length control. The evolution path p_σ is updated first.

$$p_\sigma = (1-c_\sigma)p_\sigma + \sqrt{1-(1-c_\sigma)^2}\sqrt{\mu_w}\, C_k^{-1/2 \frac{m_{k+1}-m_k}{\sigma_k}}$$
$$\sigma_{k+1} = \sigma_k \times \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|p_\sigma\|}{E\|N(0,I)\|} - 1\right)\right) \tag{26}$$
$$\mu_w = \lambda/4$$

Where $\frac{1}{c_\sigma} \approx n/3$, $C_k^{-1/2}$ is the unique symmetric square root of the inverse of C_k , and d_σ is the damping parameter usually close to one. For $c_\sigma = 0$ the step-size remains unchanged. The step-size σ_k is increased if and only if $\|p_\sigma\|$ is larger than the anticipated and decreased if it is smaller value given by:

$$E\|N(0,I)\| = \sqrt{2}\Gamma((n+1)/2/\Gamma(n/2) \approx \sqrt{n}(1-1/(4n)+1/(21\times n^2) \tag{27}$$

Finally, the covariance matrix is updated as:

$$p_c = (1-c_c)p_c + IF \times \sqrt{1-(1-c_c)^2}\sqrt{\mu_w}\frac{m_{k+1}-m_k}{\sigma_k}$$
$$C_{k+1} = (1-c_1-c_\mu+-c_s)C_k + c_1 p_c p_c^T +$$
$$c_\mu \sum_{i=1}^{\mu} wi \frac{x_{i:\lambda}-m_k}{\sigma_k}\left(\frac{x_{i:\lambda}-m_k}{\sigma_k}\right)^T \tag{28}$$
$$IF = \begin{cases} 1 & if \rightarrow \|p_\sigma\| \in [0, \alpha\sqrt{n}] \\ \|p_\sigma\| \le \alpha\sqrt{n} & Others \end{cases}$$

Where T indicates the transpose and $\frac{1}{c_c} \approx n/4$, $\alpha \approx 1.5$, $c_1 \approx 2/n^2$ and $c_\mu \approx \mu_w/n^2$. Also cs given by $cs = (1-IF)c_1 c_c(2-c_c)$.

V. Simulation

The proposed algorithm for forecasting stock prices and modeling the capital allocation problem with the two risk measures is simulated for the data of 18 shares of the TSE during the period from 12/2/2012 to 13/4/2016. The stock list selected is provided in Table 1.

TABLE I
SELECTED STOCKS TO BE PRESENT IN THE PORTFOLIO WITH NUMBER INDICES

Stocks name with number indices		
GOGEL (1)	VOTSOUM (2)	MADARAN (3)
GHAPINO (4)	KASEFA (5)	KAHIFEH (6)
WASINA (7)	KAMA (8)	JAM (9)
KHARIJK (10)	GHADASHT (11)	PARSAN (12)
ZAHDER (13)	KHOSAZ (14)	SAKHASH (15)
GHASALM (16)	KHORRAM (17)	WAGHADIR (18)

The first step is the training of the differential MLP neural network to predict the price of the 18 shares. Since we want to change the allocation of capital every 5 days, the forecast horizon is considered to be 5 days. In Table 2, the mean variance of the normalized estimation error (mse) of the prediction of price for each of the 18 stocks for the next 5 days has been presented.

TABLE II
MEAN VARIANCE OF NORMALIZED ESTIMATION ERROR (MSE) OF THE PRICE PREDICTION FOR NEXT 5 DAYS

Stock No.	1	2	3
mse	0.1396	0.0003	0.0098
Stock No.	4	5	6
mse	0.0060	0.0716	0.0870
Stock No.	7	8	9
mse	0.0018	0.1252	0.1308
Stock No.	10	11	12
mse	0.0211	0.0215	0.0254
Stock No.	13	14	15
mse	0.0031	0.0006	0.0548
Stock No.	16	17	18
mse	0.0838	0.0011	0.0035

The predictions made by the differential neural network for only two shares of Ghapino (Pars Minoo) and Ghasalm (Salemin) as an example of the stock portfolio available, are shown in Fig. 3. The neural networks trained to predict share prices in the next 5 days have been used and the above variances are used as a risk measure.

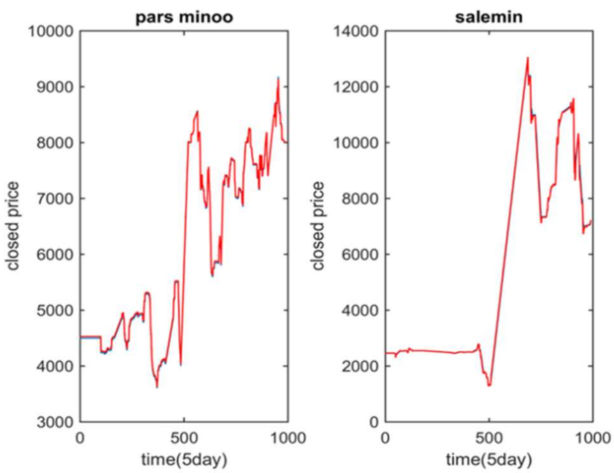

Fig. 3. Real stock price and stock price data predicted by the MLP differential neural network.

The second proposed criterion for the risk is the PI's width. In this regard, for each of the 18 shares, separately, another PI-based neural network with two hidden layers, by the training LUBE algorithm and for the 6 different neuron modes and for each case, the neural network training process is performed 3 times. In Table 3, the average $J = \frac{PICP}{PINRW}$ of trained neural networks has been written for 18 shares.

6 Neuron modes are as follows:

Mode (1): 3 neurons in the first layer and 1 neuron in the second layer.

Mode (2): 3 neurons in the first layer and 2 neurons in the second layer.

Mode (3): 5 neurons in the first layer and 1 neuron in the second layer.

Mode (4): 5 neurons in the first layer and 2 neurons in the second layer.

Mode (5): 7 neurons in the first layer and 1 neuron in the second layer.

Mode (6): 7 neurons in the first layer and 2 neurons in the second layer.

The PIs of the two Ghapino (Pars Minoo) and Ghasalm (Salemin) shares are listed as examples in Fig. 4.

TABLE III
RESULTS OF PI TRAINING

J / Stock No.	1	2	3	4	5	6
1	0.9207	0.704	1.541	5.6033	0.4239	1.376
2	1.0008	1.1564	0.7779	1.2004	1.612	1.815
3	3.2069	0.4524	2.3283	0.8628	1.8364	0.6337
4	2.2979	1.4974	1.9382	3.1303	2.075	7.3732
5	1.655	2.2632	0.8101	0.7829	1.1524	0.7613
6	1.3088	1.1248	1.9015	0.7435	1.4703	0.467
7	0.6015	1.2823	0.8006	1.7229	1.2824	1.9237
8	0.7245	0.8942	3.7359	1.0633	1.1845	0.8071
9	4.1241	0.7206	1.3364	1.4327	1.2723	2.0765
10	2.0808	1.9174	2.0396	1.3829	0.9781	0.6392
11	0.8539	0.4722	2.204	1.7669	1.5481	1.2342
12	1.2428	0.6476	1.2009	1.5271	0.8024	1.4014
13	1.4872	1.6919	0.7834	0.6851	1.5115	2.1117
14	1.3457	0.1568	1.8406	2.8036	2.6902	2.0154
15	1.0762	1.8889	0.4868	1.9573	1.7026	0.4958
16	1.4507	1.2671	3.2854	1.1466	2.1733	3.9235
17	1.4507	1.2671	3.2854	1.9683	1.697	1.4969
18	2.7421	0.635	1.4234	1.1165	0.4705	1.2961

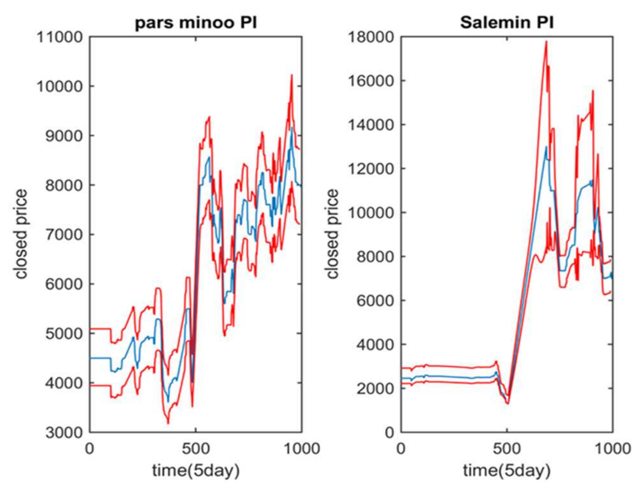

Fig. 4. PIs Generated for the two selected shares.

In recent years, the CVaR as the mean of the tail of the probability density (risk greater than VaR) has attracted a lot of attention. The CVaR as a risk criterion has shown better characteristics than VaR. This risk criterion measures the expected risk when it is greater than the specified percentile (VaR), indicating that, if the conditions are unfavorable, how much risk would it have? In other words, CVaR indicates that if changes in the stock portfolio value are likely to be $(1-\alpha)$ in the tail section of the probability density curve, then how much is the risk during a n-day period [42]. For comparing the efficiency of the proposed two risk measures, the criterion of CVaR is used in the following equation [42]:

$$CVaR_{\beta}(x) = (1-\beta)^{-1}\int_{f(x,y)\geq VaR_{\beta}(x)}^{a} f(x,y)\,p(y)\,dy \qquad (29)$$

Afterward, the problem of 5-day period capital allocation is solved for three modes using the CMAES algorithm. At this stage, as outlined in Section 3, the step-by-step strategy is applied. Thus, the low speed of the CMAES algorithm is not too troublesome, and we can still take advantage of the global search capability. At each stage, only stocks with more than 5% price change are entering the optimization process. Of course, this measurement of price change is obtained using the subtraction of price predicted by the neural network for each share for the next 5 days and the current price. Therefore, the optimization problem has been solved and the normalized returns are calculated at each step. This return is as follows:

$$R_{total}(k)=(1+r_{r_1})(1+r_{r_2})\ldots(1+r_{r_k})-1 \quad (30)$$

Where r_{r_j} is the real return of the j step compared to the previous step and is derived from the sum of the individual stock returns in the portfolio. It should be attended that the following equation is used to calculate the return on each stock:

$$r_{r_{ij}}=\left[w_{ij}r_{ij}-\left[w_{ij}-w_{i(j-1)}\right]tr\left(w_{ij}-w_{i(j-1)}\right)\right] \quad (31)$$

In Fig. 5, these returns have been plotted.

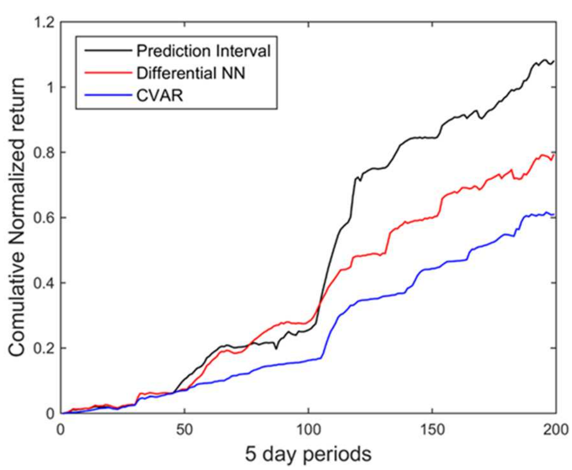

Fig. 5. Normalized returns with 5-day periods for portfolio modification with three criteria of error variance, PI width and CvaR.

It should be noted that all three methods have positive and upward returns over time. However, in all stages, the state of the PI-based risk has higher return. This comes from the fact that in a PI-based risk state, a fixed value is not assigned to the risk. In the first case, noise and uncertainty are assumed to be uniformly distributed at all times, while this is not the case in practice and the uncertainties in stock price data are affected by varying factors over time. In either case, the result of CVaR was better. The optimal weight of the stock on the last optimization step for the three methods is shown in Table 4. This difference between the weights in Table 4 is from the difference in the viewpoint in modeling of risk.

TABLE IV
THE WEIGHT OF DIFFERENT STOCKS IN THE PORTFOLIO FOR THREE DIFFERENT RISK STATES

Stock weight in CVaR risk	**Stock weight in risk mode with PI width**	**Stock weight in risk mode with estimated error variance**	**Stock No.**
0	0.0169	0.0224	**1**
0.0406	0	0	**2**
0.036	0.1534	0.1333	**3**
0.0379	0.02	0.0266	**4**
0.0319	0.0096	0.0128	**5**
0.4249	0.0018	0.0685	**6**
0.0359	0.0258	0.0158	**7**
0.038	0.154	0.0044	**8**
0.0363	0	0	**9**
0.0364	0	0	**10**
0.0373	0.1272	0.3613	**11**
0.0378	0.0013	0.0011	**12**
0.0394	0.0301	0.04	**13**
0.0298	0.0491	0.0653	**14**
0.0384	0	0	**15**
0.0296	0	0	**16**
0.0382	0.0026	0.0059	**17**
0.0323	0.5467	0.2424	**18**

Optimization of the portfolio process is repeated using PI risk for ABM and differential neural network, to compare the performance of the proposed prediction method. The result of the returns have been plotted in Fig. 6. It is obvious that the proposed NN based prediction method is more reliable than the ABM Method.

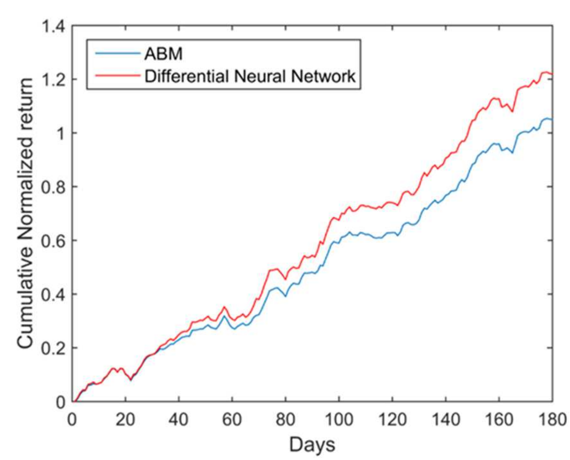

Fig 6. .Normalized returns with 5-day periods for portfolio modification with prediction Methods of neural network and ABM.

VI. CONCLUSIONS

In this paper, a neural network with differential output has been proposed to predict stock prices, which can easily withstand the non-stationary trend nature of the stock price data, and its neurons do not go to saturation. In the next step, two novel risk measures are introduced for the uncertainty involved in the stock price data. The first criterion is based on the variance of the estimation error of the neural network of the first part and is presented as a time invariant risk. The second criterion is based on the prediction interval (PI) neural network as a time varying risk to make the prediction of the stock prices more realistic. In the next step, selecting the optimal stock portfolio is considered. In this regard, first, a scalar cost function is proposed for the simultaneous involvement of the stock prices predictions of the initial neural network, risk criteria and the transaction costs. This scalar cost function transforms the selection of optimal portfolio into a single-objective optimization problem which is solved using the CMA-ES algorithm. The step-by-step strategy was used to make the optimal allocation of capital more realistic and approach to human thinking which also improved the speed of optimization problem solving. The real returns of two risk scenarios indicated that the use of normal Neural Networks based risk measures can be more effective in comparison with CVaR risk. Also, the proposed differential neural network based prediction method resulted in a higher return value and better performance than the ABM method.

REFERENCES

[1] J. A. Ou, S. H. Penman, “Financial statement analysis and the prediction of stock returns,” Journal *of accounting and economics*, vol. 11, pp. 295-329, 1989.

[2] A. Patra, S. Das, S. N. Mishra, and M. R. Senapati, “An adaptive local linear optimized radial basis functional neural network model for financial time series prediction,” *Neural Computing and Applications,* vol. 28, no. 1, pp. 101-110, 2017.

[3] L. Zhang, F. Wang, B.Xu, W. Chi, Q. Wang, & T. Sun, “Prediction of stock prices based on LM-BP neural network and the estimation of overfitting point by RDCI,” *Neural Computing and Applications,* pp.1-20, 2017.

[4] R. Adhikari, R. K. Agrawal, “A combination of artificial neural network and random walk models for financial time series forecasting,” *Neural Computing and Applications,* vol.24, no. 6, pp. 1441-1449, 2014.

[5] G. Armano, A. Murru, and F. Roli, “Stock Market Prediction by a Mixture of Genetic-Neural Experts,” *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 16, no. 5, pp. 501-526, 2002.

[6] C. M. Bishop, “Neural networks for pattern recognition,” *Oxford university press*, 1995.

[7] J.Wu, “Hybrid Optimization Algorithm to Combine Neural Network for Rainfall-Runoff Modeling,” *International Journal of Computational Intelligence and Applications,* vol. 15, no. 3, pp. 1650015-1 to 19, 2016.

[8] D. C. Park, D. M. Woo, C. S. Kim, and S. Y. Min, “Clustering of 3D Line Segments Using Centroid Neural Network for Building Detection,” *Journal of Circuits Systems and Computers,* vol. 23, no. 5, 1450071, 2014.

[9] Y. P. Huang, S. Y.Wen, W. C. Xiang, and Y. S. Jin, “PID Parameters Self-tuning Based on Genetic Algorithm and Neural Network,” *Artificial Intelligence*, vol. 10**,** 9789813206823_0003, 2016.

[10] Anufriev, M. and Hommes, C., 2012. Evolution of market heuristics. *The Knowledge Engineering Review*, *27*(02), pp.255-271.

[11] M. Anufriev; T. Bao; J. Tuinstra, “Microfoundations for Switching Behavior in Heterogeneous Agent Models: An Experiment”, Journal of Economic Behavior & Organization Volume 129, September 2016, Pages 74-99.

[12] C. Hommes, D. Veld, "Booms, busts and behavioural heterogeneity in stock prices",Journal of Economic Dynamics and Control Volume 80, July 2017, Pages 101-124.

[13] Agliari, A.; Hommes, C.H.; Pecora, N., "Path dependent coordination of expectations in asset pricing experiments: a behavioral explanation", Journal of Economic Behavior & Organization Volume 121, January 2016, Pages 15-28.

[14] S. Chen, "Agent-Based Computational Economics: How the idea originated and where it is going", Routledge, 2015.

[15] W. H. Liu, “Forecasting the semiconductor industry cycles by bootstrap prediction intervals,” *Applied Economics,* vol. 39, no. 13, pp.1731-1742, 2007.

[16] A. Khosravi, S. Nahavandi, and D. Creighton, “Construction of optimal prediction intervals for load forecasting problems,” *IEEE Transactions on Power Systems,* vol. 25, issue. 3, pp. 1496 - 1503 2010.

[17] J. H. Zhao, Z. Y. Dong, Z. Xu, and K. P. Wong, “A statistical approach for interval forecasting of the electricity price,” *IEEE Transactions on Power Systems*, vol. 23, pp. 267-276, 2008.

[18] S. G. Pierce, K. Worden, and A. Bezazi, "Uncertainty analysis of a neural network used for fatigue lifetime, prediction,” *Mechanical Systems and Signal Processing*, vol. 22, pp. 1395-1411, 2008.

[19] D. F. Benoit, D. Van den Poel, “Benefits of quantile regression for the analysis of customer lifetime value in a contractual setting: An application in financial services,” *Expert Systems with Applications*, vol. 36, pp.10475-10484, 2009.

[20] De Bock, Koen W. "Quantile Regression for Database Marketing: Methods and Applications." *In Advanced Database Marketing,* pp. 119-138. Routledge, 2016.

[21] D. L. Shrestha, D. P. Solomatine, “Machine learning approaches for estimation of prediction interval for the model output,” *Neural Networks : the Official Journal of the International Neural Network Society*, vol. 19, no. 2, pp. 225-35, 2006.

[22] J. P.IJ. van Hinsbergen, J.W.C. van Lint, and H. J. van Zuylen, “Bayesian committee of neural networks to predict travel times with confidence intervals,” *Transportation Research Part C. Emerging Technologies*, vol. 17, no. 5, pp. 498-509, 2009.

[23] A. Khosravi, S. Nahavandi, D. Creighton, E. Mazloumi, and L. J. W. C. Van, " Prediction intervals to account for uncertainties in travel time prediction,” *IEEE Transactions*

on *Intelligent Transportation Systems*, vol. 12, no. 2, pp. 537-547, 2011.

[24] A. Khosravi, S. Nahavandi, D. Creighton, E. Mazloumi, and L. J. W. C. Van, "A genetic algorithm-based method for improving quality of travel time prediction intervals," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1364-1376, 2011.

[25] A. Khosravi, S. Nahavandi, and D. Creighton, "A prediction interval-based approach to determine optimal structures of neural network metamodels," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2377-2387, 2010.

[26] G. Chryssolouris, M. Lee, and A. Ramsey, "Confidence Interval Prediction for Neural Network Models," *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 229-232, 1996.

[27] J. T. G. Hwang, A. A. Ding, "Prediction Intervals for Artificial Neural Networks," *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 748-757, 1997.

[28] G. A. F. Seber and C. J. Wild, "Nonlinear regression," John Wiley, New York, 1989.

[29] Richard D. de Veaux, J. Schumi, J. Schweinsberg and L. H. Ungar "Prediction Intervals for Neural Networks via Nonlinear Regression," *Technometrics*, vol. 40, no. 4, pp. 273-282, 1998.

[30] S. L. Ho, M. Xie, L. C. Tang, K. Xu, and T. N. Goh, "Neural network modeling with confidence bounds: a case study on the solder paste deposition process," *IEEE Transactions on Electronics Packaging Manufacturing*, vol. 24, no. 4, pp. 323-332, 2001.

[31] T. Lu, M. Viljanen, "Prediction of indoor temperature and relative humidity using neural network models: model comparison," *Neural Computing and Applications*, vol. 18, no. 4, pp. 345-357, 2009.

[32] A. Khosravi, S. Nahavandi, and D. Creighton, "Improving Prediction Interval Quality: A Genetic Algorithm-Based Method Applied to Neural Networks," *Lecture Notes in Computer Science*, pp. 141-149, 2009.

[33] D. J. C. MacKay, "The Evidence Framework Applied to Classification Networks," *Neural Computation*, vol. 4, no. 5, pp. 720-736, 1992.

[34] T. Heskes, "Practical Confidence and Prediction Intervals," *Advances in Neural Information Processing Systems*, vol. 9, pp. 176-182, 1997.

[35] Oleng', N. O., Gribok, A., & Reifman, J. (January 01, 2007). Error bounds for data-driven models of dynamical systems. Computers in Biology and Medicine, 37, 5, 670-9.

[36] N. O. Oleng', A. Gribok, and J. Reifman, "Error bounds for data-driven models of dynamical systems," Computers *in Biology and Medicine*, vol. 37, no. 5, pp. 670-9, 2007.

[37] B. Aouni, "Multi-Attribute Portfolio Selection: New Perspectives," *Infor: Information Systems and Operational Research*, vol. 47, issue. 1, pp. 1-4, 2009.

[38] G. Mainik, G. Mitov, and L. Rüschendorf, "Portfolio optimization for heavy-tailed assets: Extreme Risk Index vs. Markowitz," *Journal of Empirical Finance*, vol. 32, pp. 115-134, 2015.

[39] H. Markowitz, "Portfolio selection," *The journal of finance*, vol. 7, pp. 77-91, 1952.

[40] R. T. Rockafellar, S. Uryasev, "Optimization of conditional value-at-risk," *Journal of Risk*, vol. 2, no. 3, pp. 21-42, 2001.

[41] H. Quan, D. Srinivasan, and A. Khosravi, "Construction of neural network-based prediction intervals using particle swarm optimization," *IEEE. The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-7, 2012.

[42] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE Transactions on Neural Networks*, vol. 22, issue. 3, pp. 337-46, 2011.

[43] T. W. S. Chow, C. T. Leung, "Neural Network Based Short-Term Load Forecasting Using Weather Compensation," *IEEE Transactions on Power Systems Pwrs*, vol. 11, issue. 4, pp. 1736-1742, 1996.

[44] A. Pavelka, A. Procházka, "Algorithms for initialization of neural network weights," *In Proceedings of the 12th Annual Conference*, pp. 453-459, 2004.

[45] R. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory," *IEEE. MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science* , pp. 39-43, 1995.

[46] N. E. Huang, M. L. Wu, W. Qu, S. R. Long, and S. S. P. Shen, " Applications of Hilbert-Huang transform to non-stationary financial time series analysis," *Applied Stochastic Models in Business and Industry*, 19, 3, 245-268, 2003.

[47] R. T. Rockafellar, J. O. Royset "Superquantile/CVaR Risk Measures: Second-Order Theory", Calhoun: The NPS Institutional Archive DSpace Repository, 2015.

[48] H. Quan, D. Srinivasan, and A. Khosravi, " Short-term load and wind power forecasting using neural network-based prediction intervals," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 303-315, 2014.

[49] C. Igel, N. Hansen, and S. Roth, "Covariance Matrix Adaptation for Multi-objective Optimization," *Evolutionary Computation* vol. 15, issue. 1, pp. 1-28, 2007.

S. Amir Ghoreishi was born in Tehran, Iran. He received his B.S. degree in Electrical Engineering from Birjand University, Birjand, Iran, in 2008, the M.Sc. degree in Electrical Engineering from Islamic Azad University, South Tehran Branch, Teharn, Iran, in 2010, and the Ph.D. degree in control engineering from Islamic Azad University, Science and Research Branch, Teharn, Iran, in 2018. In 2010, he joined Islamic Azad University, Varamin Branch as an Instructor in the School of Electrical Engineering. His current research interests include optimal control, modelling, soft computing, intelligent systems, and fuzzy asset.

Hamid Khaloozadeh received the B.Sc. degree in control engineering from Sharif University of Technology, Tehran, Iran, in 1990, the M.Sc. degree in control engineering from K. N. Toosi University of Technology, Tehran, in 1993, and the Ph.D. degree in control engineering from Tarbiat Modarres University, Tehran, in 1998.

He is currently a Professor with the Department of Systems and Control, Faculty of Electrical Engineering at the K.N. Toosi University of Technology. He is director of the Industrial Control Center of Excellence (ICCE), at K.N. Toosi University of Technology. His interest area is Stochastic Estimation and Control, System Identification, Optimal Control, Adaptive Control, and Time Series Analysis.