International Journal of Business and Development Studies **Vol. 10, No. 1,** (2018) pp 95-113

Optimal Portfolio Allocation based on two Novel Risk Measures and Genetic Algorithm

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

Abstract:

The problem of optimal portfolio selection has attracted a great attention in the finance and optimization field. The future stock price should be predicted in an acceptable precision, and a suitable model and criterion for risk and the expected return of the stock portfolio should be proposed in order to solve the optimization problem. In this paper, two new criterions for the risk of stock price prediction has been presented, of which the first one is based on the interval predictions which vary with time and proportional to the uncertainty of stock price data, while the second one is a constant risk term that is proportional to the prediction error variances of the neural networks. A novel cost function has been presented to simultaneously consider the expected returns and risks.

Genetic algorithm has been used to solve this optimization problem. Finally, 18 shares of the Tehran Stock Exchange have been considered to evaluate the performance of the proposed risk criterions. Two proposed risk criteria, by the conditional value at risk (CVaR) associated with the same stock. The problem of stock portfolio optimization has been solved for all three situations, and the PI-based risk criteria yielded a better return.

Keywords: Stock Portfolio Optimization; Interval Prediction Neural Networks; Conditional Value at Risk; Risk measure.

1. Introduction

Stock price prediction is a challenging problem, due to many involved factors, including political events, economic conditions, oil and gold prices, traders' expectations and other environmental factors, which can influence stock prices. In addition, the stock price series generally have a noisy, dynamic, non-linear, complex, non-parametric and chaotic nature.

Therefore, stock market forecasting is considered as a challenging task in predicting financial time series. Financial time series exhibits rather complex patterns (trends, sudden changes, and volatility clustering) and such series are often non-stationary, whereby a variable does not have a

[1] (Ph.D. Candidate) Faculty of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.

[2] (Professor) Industrial Control Center of Excellence (ICCE), Department of Systems and Control, Faculty of Electrical Engineering, K. N. Toosi University of Technology, Tehran, Iran. Corresponding Author Email: h_khaloozadeh@kntu.ac.ir

clear tendency to move towards a constant value or a linear trend. Static prediction models, including 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 considered as one of the most powerful tools in financial forecasting [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 approximators) [5]. They do not require any presupposition for the characteristics of the system data [6], and are widely used for modeling, classification, prediction and control [7-9].

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 [10]. This problem has been investigated in the load forecasting of the distribution network [11-12], predicting the life expectancy of components [13], the predictions needed to provide financial services [14], the predictions needed in water distribution systems [15], transport systems [16-18] and baggage handling systems [19], have similar problems with conventional neural networks.

The second difficulty of the neural networks is that they produce a single point prediction output and do not provide any criterion for measurement of its accuracy. It is crucial to consider the uncertainties in single-value and point predictions in order to improve 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 (-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 [20-30]. 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 [31-32].

$$Maximize(Expected\ return - Risk) \qquad (1)$$

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 of capital 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 [33].

As noted above, stock price prediction and the presentation of a suitable risk model are two important issues that must be addressed before the optimal portfolio selection. This paper has investigated 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 is solve using Genetic Algorithm by applying 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 [34] are explained. In Section 3, the prediction stages with the aid of a differential 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, 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.

2. The necessary preparations for PI

In order to achieve PI, the method presented in [34] has been utilized for training the neural network. In this method, which is called the Lower Upper Bound Estimation (LUBE), all efforts are made to avoid using the output derivative of the neural network with respect to its parameters, which, according to [29], can cause inaccuracy in prediction.

In the LUBE method, unlike previous methods, efforts are not made to reduce the sum of squares of error. But obtaining PIs with the minimum possible width and the acceptable level of coverage probability is the main target. A neural network is used to estimate the upper and lower bound for the production of PIs [19]. One of the criteria that plays the most important role in the production of an appropriate PI is the degree of confidence$(1 - \alpha)$, where$(\alpha < 1)$. By definition, this is the quantity indicating the percentages of the system output included by PI. Naturally, in the ideal state, $= 0$. The degree of confidence is also known by the name of the coverage percentage of PI or PICP. Expressing it mathematically we have [19, 15, and 11]:

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

Where n is the total number of data and c_i is equal to one if i thdata is inside the PI band, and it is zero if it is outside of this band.

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

Note that $L(x_i)$ and $U(x_i)$ are, respectively lower and upper bounds of PI for i thdata. In Fig. 1, the Pi-based neural network is presented. As noted before, only the PICP reaching one is not a good measure for evaluating the performance of PI. Another important criterion is the PI band being as small as possible. The prediction interval normalized average width (PINAW) can be used for this purpose:

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

Where R is the maximum difference between the upper and lower bounds of PIs. The reason for using it is to normalize PINAW between zero and one. This makes it possible to compare it with the PICP measure. PINAW and PICP have a direct relationship with each other.

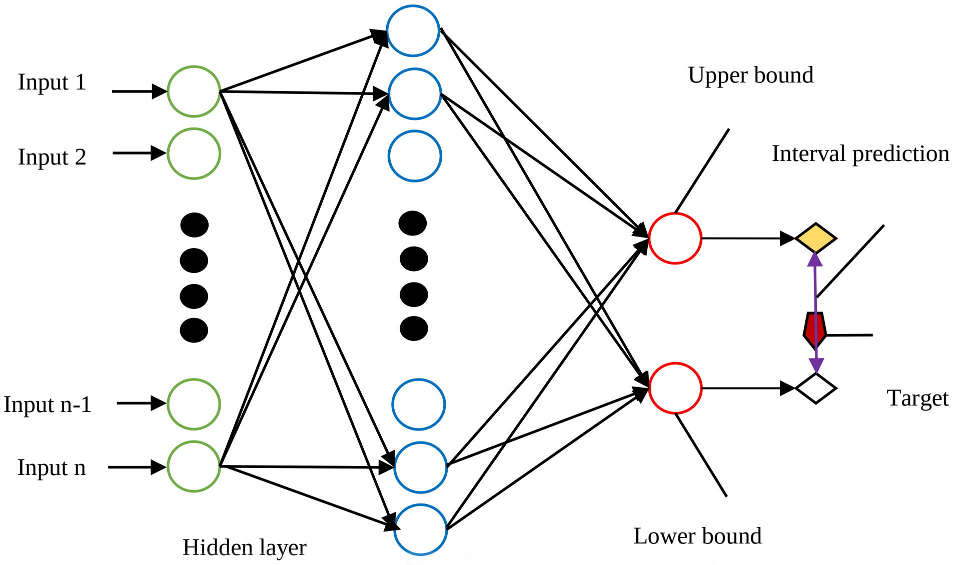

Fig. 1. The PI based on neural network

Therefore, the more the value of PINAW the more will be the value of PICP. But since the neural networks are trained using square measures, instead of using the PINAW criterion for producing PI, the Prediction Interval Normalized Root-mean-square Width (PINRW) is used, which is defined as follows:

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

The advantage of PINRW over PINAW is that it gives greater importance to longer-range intervals and as a result, PIs produced with PINRW are expected to be smaller than those produced by PINAW.

As stated, PIs with a higher coverage percentage and lower widths are preferable and on the other hand, increasing the coverage percentage and reducing the width of the PIs is in contradiction with each other and should be a balance between them. Therefore, PI can be modeled by a multi-objective optimization problem:

$$\left(\underset{\omega}{\text{Max}}\, PICP\right) \& \left(\underset{\omega}{\min}\, PINRW\right) \qquad s.t.\,(0 < PICP \leq 1) \,\&\, (PINRW > 0) \quad (6)$$

In (6) finding the optimal weights of the neural network (W^*) is in such a way that the maximum PICP and the PINRW are minimized. The solution of (6) is time consuming and even a significant portion of Pareto's answers will not be acceptable, because the overall reliability of PI will be lost if the PICP falls below a certain threshold as stated in [34]. The requirement that the PICP be increased beyond the limits is considered as a requirement and the optimization problem in modified in the following single objective form:

$$\left(\min_{\omega} PINRW\right) \qquad s.t.\,(\mu \leq PICP \leq 1)\ \&\ (PINRW > 0) \qquad (7)$$

In (7), finding the optimal weights of the neural network (W^*) is done to minimize the PINRW. is the threshold of minimum acceptable PICP. It should be noted that solving the problem of single-objective optimization is usually simpler than solving a multi-objective problem and requires less time. Afterward, the single-objective optimization problem will be solved by the PSO algorithm.

2.1. LUBE method

The traditional methods for producing neural network based PIs are in two stages [35]:

1- Neural networks similar to point prediction methods are trained to model learning data.

2- Assuming a specific distribution for the data, their mean values and variance are calculated then, according to the method used, the Hessian and Jacobian matrices of the neural network coefficients are also created. Then PIs are made.

These methods suffer from multiple problems in building PIs. For example, the Delta method takes specific assumptions for data distribution and residues. Also, in the process of building PIs, both the Delta and Bayesian methods require the calculation of the Jacobin and Hessian matrices. This may lead to the problems of specificity of matrices and reduced reliability of PIs. Moreover, the calculation of these matrices is tedious and greatly increases the implementation complexity of these methods. The new LUBE method has been introduced in [36] for the first time. This method trains a two-output neural network to build PIs in one step, without any presupposition on data distribution such that, these two outputs are related to the lower and upper bounds of PI. The LUBE algorithm briefly includes the following steps:

(i) All available data are divided into 3 parts: train, evaluation and test. The train data is used to determine the parameters of the neural network and the evaluation data is used to determine the optimal structure of the neural network, including the number of hidden layers and the number of neurons in each layer. Test data is also used to measure the performance of the neural network.

(ii) Since the stock market data is not static, they become stationary with differencing. According to the proposed algorithm [37], this act is done only for the output of the neural network.

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

In (8), Δx_{t+n} is the static data set obtained by differencing.

(iii) At this stage, the optimal structure of the neural network will be determined. Therefore, for each candidate of the neural network structure, the training process is repeated to a certain number limit (in this paper, the training process is repeated 3 times) and using the evaluation data, the PICP and PINRW of each trained network are calculated. Then, the optimal structure of the neural network is selected with a certain criterion. In this paper, contrary to [34], a neural network is chosen with a larger cost function J. Where J of each candidate of the neural network is the mean of the 3 neural networks cost functions.

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

The reason for choosing this criterion is that the PI will be chosen with a higher PICP and a lower PINRW. Of course, this criterion is calculated only for the neural network where PICP> .

(iv) Updating the particle velocity and position values are the main requirements of PSO algorithm. The update of the velocity and particle position are as follows [38]:

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

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

Where v_n is the velocity of the particle in the n-th dimension, rand(.) is a random number between 0 and 1, W is a weighting factor, and C_1 and C_2 are also coefficients that determine the relative importance between P_{best} and g_{best} [39]. In addition to update laws, V_{max} and X_{max} determine the maximum allowed particle velocity and position.

(v) Selection, crossover, and mutation are the main operators of the genetic algorithm. The mutation operator of the genetic algorithm, which causes variation in the population, has been employed in the mentioned PSO algorithm. This will greatly increase the search power of the PSO algorithm and prevents it from being trapped in the local extremum (The mutation operator will act randomly on the position and velocity of a number of particles).

(vi) After updating the neural network weights, PICP and PINRW are calculated for the train and evaluation data.

(vii) P_{best} and g_{best} are updated such that, P_{best} is the best value of each particle and g_{best} is the best value for the entire population. It should be noted that during this update, the optimization constraints will also be considered.

(viii) If the above algorithm is implemented to a certain repetition, or the problem of optimization is satisfactorily solved, proceed to the next step, otherwise return to step 4.

(ix) Upon completing the training, g_{best} will be used for neural network weights and PI production for test data. Then, PICP and PINRW are calculated and these values are reported.

3. The proposed method

In this section, the stages of the process from the prediction of stock prices to optimal allocation of capital to each share will be explained.

3.1. Stock price prediction

At this stage, the MLP neural networks are used to predict n future days of stock prices (for the purpose of this study, $n = 5$). Since TANSING neurons in the neural network are used, input data must be normalized to the interval [-1 and 1]. Neural networks work well in the case of static data prediction. Then, the neural network model can be written in the following form:

$$\hat{x}_{t+n} = f(x_t, x_{t-1}, x_{t-2}, \dots, x_{t-p}) + e_{t+n} \tag{12}$$

Where x_{t-i} is the stock price data of i days before and e_{t+n} is the estimated error of x_{t+n}. But, as stated earlier in different references, [40-41], stock prices data do not have static nature and so neural networks will not succeed in estimating the future values of these data, unless somehow the data will be static. Differencing is the main approach in creating static data. Because we want to predict the data of n forward samples, the data with n intervals are subtracted from each other. So, we have:

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

Since the data are not static, x_{t+n} and x_t have a relatively high correlation for small and medium n, then, Δx_{t+n} and consequently $\Delta\hat{x}_{t+n}$ will be small [38]. So, the problems in the estimation of non-static data by the neural network will be greatly resolved. Therefore, the inputs of the neural network are normalized stock price data of today and the past days. In predicting stock prices for the next 5 days, the price data sample of each day is subtracted from the data of 5 days later and consider is considered as the output of the neural network after normalizing between [-1 1].

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

Neural network training can be done using common algorithms such as gradient descend, Levenberg Marquardt, and so on. After training neural networks and determining their optimal structure for each share, the neural networks output and today's price data are used to estimate the price of n days later. These estimates will be used to obtain expected returns as follows:

$$r_{t+n} = \frac{\hat{x}_{t+n} - x_t}{x_t} = \frac{\Delta\hat{x}_{t+n}}{x_t}. \qquad (15)$$

It is deduced from (14) that it is not necessary to calculate $\hat{x}_{t+n}$, and $\Delta\hat{x}_{t+n}$ can be used directly, since the purpose of the future stock price estimation is to obtain the expected returns and use it in the process of optimizing the allocation of capital.

3.2. Expected return risk estimation

Since the uncertain factors are involved, there cannot be 100% trust in the price estimation performed by the neural network. Therefore, there is need to incorporate this uncertainty into the expected period. This paper uses two perspectives for this issue:

(1) Using the variance of the estimation error of the neural network training phase as a constant risk:

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

(2) Use of interval prediction based neural network. After training the relevant neural network, PI width of the test data is used as the time varying risk of each share $U_i - L_i$. Given that PIs have been developed for the uncertainty impacts on system data, it is expected that the defined risk would be more realistic and will increase real returns in the next stage, i.e., the allocation of capital.

3.3. Cost function and optimization of capital allocation

Now, the expected returns and the associated risk have been introduced, the cost function will be introduced for solving the problem of optimizing the capital allocation. The expected returns and associated risk will be simultaneously incorporated into a cost function in the form below to simplify the solving process of the optimization problem:

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

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

Where $w_i(t)$ is the weight of the i-th share in the stock portfolio at time t, and $w_i(t+n)$ is the weight of the i-th share at time t + n. m is the total number of stocks in the portfolio, $r_i(t+n)$ is also normalized expected return and $\sigma_i(t+n)$ is also the risk of predicting the return on the i-th share. A penalty is also included for the purchase and sale of portfolio stock shares. Depending on how much share is sold and the amount of share purchased, our capital will be lost. bp and sp are, respectively share buy and sale penalties. Since these two values are different, the function $tr(x)$ has been proposed for the easier impact of these two different penalties.

This function is based on the difference in stock weight over two consecutive periods. If the weight of the i-th share decreases in the next period, this difference will be negative and $tr=sp$. Also, if the weight of i-th share in the next period increases, this difference will be positive and $tr=bp$. Thus, both buy and sale penalties for each share will be easily applied.

Afterward, the mentioned optimization problem will be solved by a global search algorithm such as Genetics algorithm. It is not a wise decision to incorporate all the stock in the portfolio in each stage of optimization. Because the optimization problem solving process will be too time consuming. A strategy called step-by-step is used to overcome this problem. This strategy states that we only buy a stock that has a higher price growth than a predetermined threshold and only sell shares that lower price drop than a specified threshold percentage (this is the desired limit, which is assumed to be 5% here). Therefore, if the stock price in the portfolio did not change much (whether ascending or descending), it will not take part in the optimization. This strategy is also closer to human thinking and it also causes all the stock in the portfolio not to enter the algorithm at each optimization step and so the time duration of the optimization problem will be greatly reduced. This strategy is expressed in (19):

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

Diagram of the proposed method for determining the stock optimal portfolio is shown in Fig. 2.

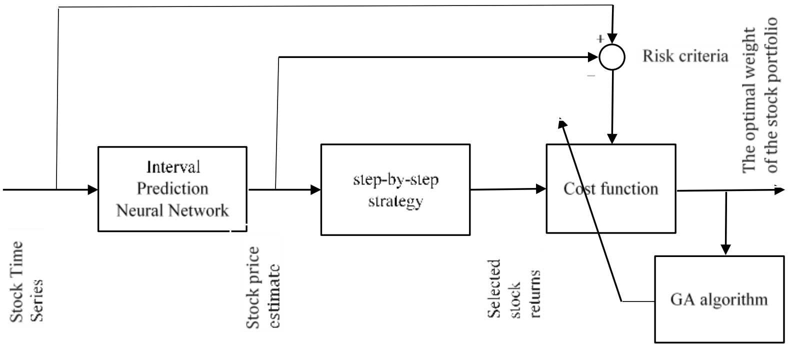

Fig. 2. Diagram of the proposed method for determining the stock optimal portfolio

4. Simulation

In this section, the proposed algorithm for predicting stock prices and modeling the capital allocation problem with the two risk measures presented in Section 3, is simulated for the data of 18 shares of the Tehran Stock Exchange during the period from 12/2/2012 to 13/4/2016. The stock list selected is provided in Table 1.

Table 1. Selected stocks to be present in the portfolio 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)	SAHASHAH (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 2. Mean variance of normalized estimation error (mse) of the price prediction for next 5 days

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

The predictions made by the differential neural network for only two shares of KHOOSAZ and WAGHADIR as an example of the stock portfolio available, are shown in Fig. 3. The normalized estimation error of these two shares is also shown in Fig. 4. 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.

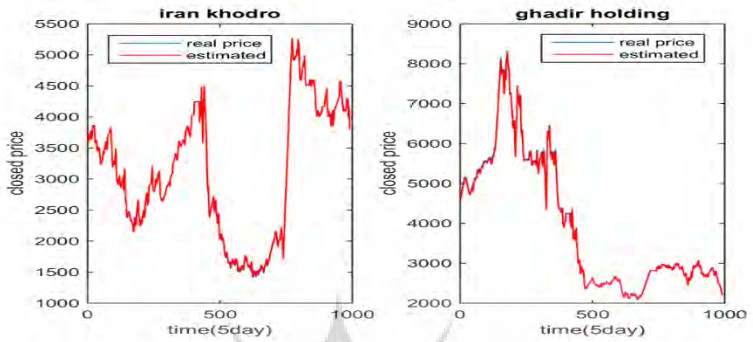

Fig. 3. Real stock price and stock price data predicted by the MLP differential neural network

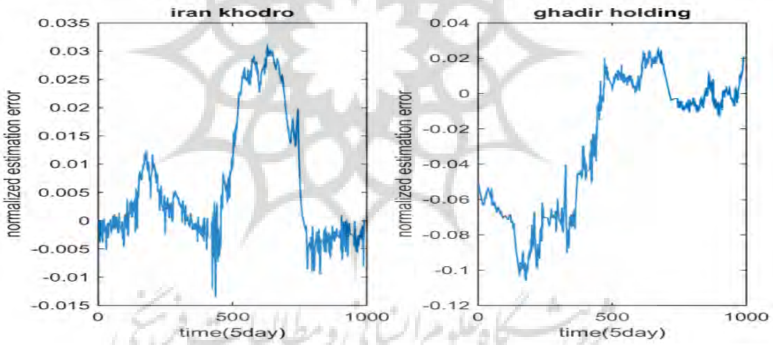

Fig. 4. Normalized estimation error in two sample shares

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:
(1) 3 neurons in the first layer and 1 neuron in the second layer
(2) 3 neurons in the first layer and 2 neurons in the second layer
(3) 5 neurons in the first layer and 1 neuron in the second layer
(4) 5 neurons in the first layer and 2 neurons in the second layer
(5) 7 neurons in the first layer and 1 neuron in the second layer
(6) 7 neurons in the first layer and 2 neurons in the second layer
The PIs of the two KHOOSAZ and WAGHADIR shares are listed as examples in Fig. 5.

Table 3. Results of PI training

The value of J Stock No.	۱	۲	۳	۴	۵	۶
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

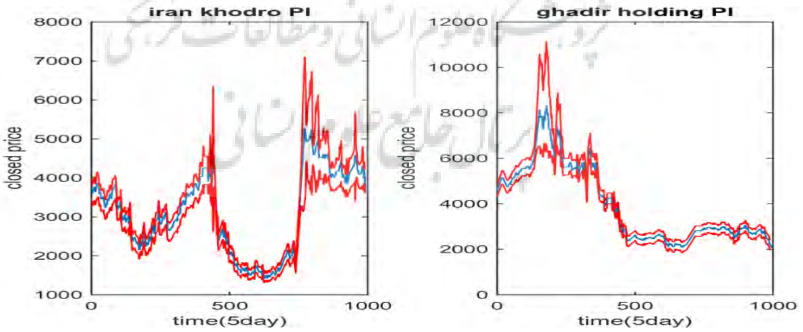

Fig. 5. PIs generated for the two selected shares

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]:

$$CVaR_{\beta}(x) = (1-\beta)^{-1} \int_{f(x,y) \geq VaR_{\beta}(x)}^{a} f(x,y)p(y)dy. \qquad (20)$$

Afterward, the problem of 5-day period capital allocation is solved for three modes using the genetic algorithm. At this stage, as outlined in Section 3, the step-by-step strategy is applied. Thus, the low speed of the genetic 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 in each step. This return is as follows:

$$R_{total}(k) = \left(1 + r_{r_1}\right)\left(1 + r_{r_2}\right) \dots \left(1 + r_{r_k}\right) - 1. \qquad (21)$$

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 noted that the following equation was used to calculate the return on each share:

$$r_{r_{ij}} = \left[w_{ij}r_{ij} - \left[w_{ij} - w_{i(j-1)}\right].tr\left(w_{ij} - w_{i(j-1)}\right)\right]. \qquad (22)$$

In Fig. 6, these returns have been plotted.

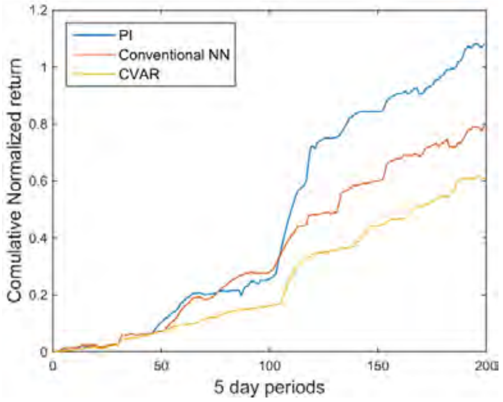

Fig. 6. Normalized returns with 5-day periods for portfolio modification with three criteria of error variance, PI width and CVaR

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 at 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 4. 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

Conclusion

In this paper, MLP neural networks were used to predict stock prices. These predictions were considered as the expected returns of different stocks. The risk of these predictions should also be taken into account to make the allocation process more realistic. The proposed approach utilized the uncertainty of neural network in the prediction of output. Once the error variance of neural network training phase estimation was used as a constant risk value, afterward the interval prediction neural network using PI width for the test data was applied as a time-varying risk. Then, a cost function was proposed for simultaneous consideration of the proposed risk and expected returns, and the problem of optimal allocation of capital to different stocks was modeled as a single-objective optimization problem.

This optimization problem was solved by the genetic 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 on PI to determine the risk, can be more effective in comparison with CVaR risk.

References

1. Ou, J. A., & Penman, S. H. Financial statement analysis and the prediction of stock returns. Journal of accounting and economics, 11(1989), 295-329.

2. Patra, A., Das, S., Mishra, S. N., & Senapati, M. R. An adaptive local linear optimized radial basis functional neural network model for financial time series prediction. *Neural Computing and Applications*, 28(2017), 101-110.

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

4. Adhikari, R., & Agrawal, R. K. A combination of artificial neural network and random walk models for financial time series forecasting. *Neural Computing and Applications*, 24, (2014), 1441-1449.

5. Armano, G., Murru, A., & Roli, F. Stock market prediction by a mixture of genetic-neural experts. *International Journal of Pattern Recognition and Artificial Intelligence*, *16(2002),* 501-526.

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

7. Wu, J. Hybrid Optimization Algorithm to Combine Neural Network for Rainfall-Runoff Modeling. *International Journal of Computational Intelligence and Applications*, *15(2016),* 1650015.

8. Park, D. C., Woo, D. M., Kim, C. S., & Min, S. Y. Clustering of 3D line segments using centroid neural network for building detection. *Journal of Circuits, Systems, and Computers*, *23(2014),* 1450071.

9. HUANG, Y. P., WEN, S. Y., XIANG, W. C., & JIN, Y. S. PID Parameters Self-tuning Based on Genetic Algorithm and Neural Network. Artificial Intelligence, 10(2016), 9789813206823_0003.

10. Liu, W. H... Forecasting the semiconductor industry cycles by bootstrap prediction intervals. *Applied Economics*, 39 (2007), 1731-1742.

11. Khosravi, A., Nahavandi, S., & Creighton, D. Construction of optimal prediction intervals for load forecasting problems. *IEEE Transactions on Power Systems*, 25(2010), 1496-1503.

12. Zhao, J. H., Dong, Z. Y., Xu, Z., & Wong, K. P. A statistical approach for interval forecasting of the electricity price. *IEEE Transactions on Power Systems*, 23(2008), 267-276.

13. Pierce, S. G., Worden, K., & Bezazi, A. Uncertainty analysis of a neural network used for fatigue lifetime, prediction.*Mechanical Systems and Signal Processing*, *22*(2008), 1395-1411.

14. Benoit, D. F., & Van den Poel, D. Benefits of quantile regression for the analysis of customer lifetime value in a contractual setting: An application in financial services. *Expert Systems with Applications*, 36(2009), 10475-10484.

15. Shrestha, D. L., & Solomatine, D. P... Machine learning approaches for estimation of prediction interval for the model output. *Neural Networks*, 19(2006), 225-235.

16. Van Hinsbergen, C. I., Van Lint, J. W. C., & Van Zuylen, H. J. Bayesian committee of neural networks to predict travel times with confidence intervals. *Transportation Research Part C: Emerging Technologies*, 17(2009), 498-509.
17. Khosravi, A., Mazloumi, E., Nahavandi, S., Creighton, D., & Van Lint, J. W. C. Prediction intervals to account for uncertainties in travel time prediction. *IEEE Transactions on Intelligent Transportation Systems*, 12 (2011), 537-547.
18. Khosravi, A., Mazloumi, E., Nahavandi, S., Creighton, D., & Van Lint, J. W. C. A genetic algorithm-based method for improving quality of travel time prediction intervals.*Transportation Research Part C: Emerging Technologies*, 19 (2011), 1364-1376.
19. Khosravi, A., Nahavandi, S., & Creighton, D... A prediction interval-based approach to determine optimal structures of neural network metamodels. *Expert systems with applications*, 37(2010), 2377-2387.
20. Chryssolouris, G., Lee, M., & Ramsey, A. Confidence interval prediction for neural network models. *IEEE Transactions on neural networks*, 7(1996), 229-232.
21. Hwang, J. G., & Ding, A. A. Prediction intervals for artificial neural networks. *Journal of the American Statistical Association*, 92 (1997), 748-757.
22. Wild, C. J., & Seber, G. A. F. Nonlinear regression. 1989.
23. De VlEAUX, R. D., Schumi, J., Schweinsberg, J., & Ungar, L. H. Prediction intervals for neural networks via nonlinear regression. *Technometrics*, 40(1998), 273-282.
24. Ho, S. L., Xie, M., Tang, L. C., Xu, K., & Goh, T. N. Neural network modeling with confidence bounds: a case study on the solder paste deposition process. *IEEE Transactions on Electronics Packaging Manufacturing*, 24(2001), 323-332.
25. Lu, T., & Viljanen, M. Prediction of indoor temperature and relative humidity using neural network models: model comparison. *Neural Computing and Applications*, 18(2009), 345.
26. AKhosravi, A., Nahavandi, S., & Creighton, D. Improving prediction interval quality: A genetic algorithm-based method applied to neural networks. In*International Conference on Neural Information Processing, 2009* (pp. 141-149). Springer, Berlin, Heidelberg.
27. MacKay, D. J. The evidence framework applied to classification networks. *Neural computation*, 4(1992), 720-736.
28. Heskes, T. Practical confidence and prediction intervals. In *Advances in neural information processing systems, (1997)*, (pp. 176-182).
29. Dybowski, R., & Roberts, S. J. Confidence intervals and prediction intervals for feed-forward neural networks.*Clinical Applications of Artificial Neural Networks*, 2001, 298-326.
30. Oleng, N. O., Gribok, A., & Reifman, J. Error bounds for data-driven models of dynamical systems. *Computers in biology and medicine*, 37(2007), 670-679.

31. Aouni, B. Multi-attribute portfolio selection: New perspectives. *INFOR: Information Systems and Operational Research*, 47(2009), 1-4.
32. Mainik, G., Mitov, G., & Rüschendorf, L. Portfolio optimization for heavy-tailed assets: Extreme Risk Index vs. Markowitz. *Journal of Empirical Finance*, *32(2009),* 115-134.
33. Markowitz, H. Portfolio selection. *The journal of finance*, *7(1952,* 77-91.
34. Rockafellar, R. T., & Uryasev, S. Optimization of conditional value-at-risk. *Journal of risk*, *2*(2000), 21-42.
35. Quan, H., Srinivasan, D., & Khosravi, A. Construction of neural network-based prediction intervals using particle swarm optimization. In *Neural Networks (IJCNN), The 2012 International Joint Conference on, 2000,* (2012). (pp. 1-7). IEEE.
36. Khosravi, A., Nahavandi, S., Creighton, D., & Atiya, A. F. Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE Transactions on Neural Networks,* 22(2011), 337-346.
37. Chow, T. W. S., & Leung, C. T. Neural network based short-term load forecasting using weather compensation. *IEEE Transactions on Power Systems*, 11(1996), 1736-1742.
38. Pavelka, A., & Procházka, A. Algorithms for initialization of neural network weights. In *In Proceedings of the 12th Annual Conference, MATLAB,* (2004), (pp. 453-459).
39. Eberhart, R., & Kennedy, JA new optimizer using particle swarm theory. In Micro Machine and Human Science, 1995. MHS 95Proceedings of the Sixth International Symposium on, (1995) (pp. 39-43). IEEE.
40. [Huang, N. E., Wu, M. L., Qu, W., Long, S. R., & Shen, S. S. Applications of Hilbert˚ Huang transform to non-stationary financial time series analysis. *Applied stochastic models in business and industry*, 19(2003), 245-268.
41. Clements, M. P., & Hendry, D. F. *Forecasting non-stationary economic time series*. Mit Press. 2001.
42. Quan, H., Srinivasan, D., & Khosravi, a Short-term load and wind power forecasting using neural network-based prediction intervals. *IEEE transactions on neural networks and learning systems*, 25(2014), 303-315.