01 Jan 2004

Forecasting Series-Based Stock Price Data using Direct Reinforcement Learning

H. Li

Cihan H. Dagli
Missouri University of Science and Technology, dagli@mst.edu

David Lee Enke
Missouri University of Science and Technology, enke@mst.edu

Recommended Citation

H. Li et al., "Forecasting Series-Based Stock Price Data using Direct Reinforcement Learning," *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2004.
The definitive version is available at https://doi.org/10.1109/IJCNN.2004.1380088

Forecasting Series-based Stock Price Data using Direct Reinforcement Learning

Hailin Li, Cihan H. Dagli, and David Enke
Department of Engineering Management
University of Missouri-Rolla
Rolla, MO USA 65409-0370
E-mail: {hl8p5, dagli, enke}@umr.edu

Abstract—**A significant amount of work has been done in the area of price series forecasting using soft computing techniques, most of which are base upon supervised learning. Unfortunately, there has been evidence that such models suffer from fundamental drawbacks. Given that the short-term performance of the financial forecasting architecture can be immediately measured, it is possible to integrate reinforcement learning into such applications. In this paper, we present the novel hybrid view for a financial series and critic adaptation stock price forecasting architecture using direct reinforcement. A new utility function called policies-matching ratio is also proposed. The need for the common tweaking work of supervised learning is reduced and the empirical results using real financial data illustrate the effectiveness of such a learning framework.**

I. Introduction

Forecasting series-based stock price data via soft computing techniques has in fact already taken shape in the past decade. Although many academics and practitioners have tended to regard such application with a high degree of skepticism, there has been accumulating evidence that the markets are not fully efficient and the Artificial Intelligent-based models can outperform the benchmark models (*e.g.* Random Walk model). We anticipate that more Internet financial service providers will incorporate AI techniques in their service to address the current industry trends, such as cheap real-time information, financial market and institutional deregulation, and global capitalization.

Till now, most of related research is based on traditional supervised learning techniques. Usually, artificial neural network (ANN) models are adopted as the core engine in the forecasting architecture. It is believed that non-linear relationships exist between financial variables and stock returns. Often hybrid architectures are proposed in order to obtain better predictions than simple ANNs models currently provide. Such developments include the synthesis of genetic algorithms and ANNs [1], Neuro-fuzzy architectures [2], combination of qualitative and quantitative data [3], and ARIMA-based ANNs [4]. Unfortunately, much of the published literature is still somewhat academic. The results are case sensitive and hard to apply. In essence, all efforts under the supervised learning framework will be subject to the fundamental limitation in terms of why historical patterns of the financial series should be expected to repeat. The strength of models in interpolation cannot promise the exploration ability that obviously is crucial for this application.

Real time financial performance depends upon the sequences of interdependent decisions and is thus path-dependent. In other words, a series of actions must be taken in sequence and the quality of these actions usually cannot be determined until the end of the sequence. This is a much harder problem than supervised learning algorithms often face. In this sense, many financial applications will fall into the reinforcement learning domain. Classical dynamic programming methods have already been applied to asset allocation [5], portfolio optimization [6], and derivatives pricing applications [7]. Recently, Moody *et al* [8] proposed a recurrent reinforcement-learning method to learn trading policies. In terms of forecasting financial series data, it is difficult to integrate reinforcement learning techniques directly. Few published research studies cover this issue. In [9] we proposed a new hybrid view for financial series. The architecture offers the basis for further analysis using reinforcement learning. The Q-Learning approach is also adopted to forecast future prices trends purely from historical stock price data. The empirical results reported in that paper show the effectiveness of this new thought. Opposite to the claims of the Random Walk Theory, historical financial series may provide indications to predict future trends.

However, the Neuro-Q forecasting architecture in [9] is likely to suffer from several limitations due to the nature of the value function reinforcement learning. For a discrete-time dynamic system that is suitable for most financial series prediction environment, the "Bellman's curse of dimensionality" is unavoidable. Moreover, as pointed out by Brown [10], the policies produced by Q-learning tend to be brittle because of the noisy financial data. The recurrent reinforcement learning (RRL) algorithm presented by [8] is a type of direct policy learning that eliminates the intermediate value function estimation procedure. Inspired by its strength in problem representation and computation efficiency, this paper proposes the novel Critic adaptation forecasting architecture to predict series-based stock prices. RRL is utilized as the learning algorithm for the critic network. We demonstrate how learning can be implemented under the

0-7803-8359-1/04/$20.00 ©2004 IEEE

proposed schema. Experiment results using real financial data are used to illustrate the effectiveness of such a learning framework.

The remainder of this paper is structured as follows. Section 2 describes the related adaptive critic design concepts and the novel hybrid view (model) for financial series prediction. Section 3 presents the implementation of the proposed critic adaptation stock price forecasting architecture. In section 4, empirical results are provided and discussions are included, followed by conclusions in Section 5.

II. Financial Series Learning using a Critic and Hybrid View

The concern of reinforcement learning regards decision-making in uncertain environments. The essential idea behind reinforcement learning is a simple penalty-reward strategy. Compared with supervised learning, reinforcement learning techniques are a form of match-based learning (other than error-based learning) due to the fact that the correct target output value for each input set or pattern is lacking and typically only delayed reward signals are available. Classical dynamic programming (DP) is the well-known approach used to handle the reinforcement learning problem under both deterministic and stochastic cases, but its exhaustive search strategy is computationally expensive for most real problems. Furthermore, the backward search direction of DP precludes its utilization in real-time (or on-line) policy generation. To address such issues, adaptive critic design frameworks and various heuristic-programming methods are gaining popularity in the recent literature [11, 12].

In general, critic methods integrate backpropagation (a way to obtain necessary derivatives) with reinforcement learning through a critic network. The critic network, rather than "actor" or control network, learns to approximate a strategic utility function (*i.e.* J function in DP's Bellman's equation) and uses the actor's output as part of its inputs, directly or indirectly. Thus, the needed learning signal is the critic's output J, or one-step utility U, which vary according to the adopted techniques, instead of desired system output. Specifically, learning for this research will be implemented based on the RRL algorithm under which there is no backpropagation path to the action network, but which uses the signal of the actor to estimate a utility function. For such a critic adaptation model, the cost-to-go function is expressed as follows:

$$J(t) = \sum_{k=0}^{\infty} \gamma^k U(t+k), \tag{1}$$

where $0 < \gamma < 1$ is a discount factor for finite horizon problems.

The target value for $J(t)$ can be $\gamma J(t+1) + U(t)$, allowing the critic to be trained forward in time, which is the key for real-time application.

In the financial area, high market volatility often creates difficulties for existing theories to provide clear explanations of price behaviour. The changing of the financial series results from the dynamics of the complex system. Supervised learning-based models can generalize the non-linear relationship between stock returns and various Micro/Macro financial factors when the market is "smooth and calm". Unfortunately, they lack the exploration ability to catch unexpected price movements in time. Essentially, all volatility is the direct consequence of the investment behaviours of people, no matter if the decisions of investors are made base upon technical or fundamental analysis. Obviously, the statistically significant structure of the financial time series found by supervised learning is highly difficult (if not impossible) to represent such evolutionary behaviour.

To account for such observation, we propose a financial series hybrid view in which the observed price p_t is the result influenced by the combination of inherent market inertia (*i.e.* "normal" market patterns that can be generalized from historical data) and the actions taken by investors at the immediate previous market "state" (resulting in the "unexpected" market response). The price series generalized from "normal" market patterns, rp_t, can be viewed as the observed underlying "rational price" process. The most recent investment actions (buy/sell/hold) of investors change the output from the market inertia. The final observed noisy financial series is the summation of the two separated processes:

$$p_t = rp_t + a_t + \varepsilon_t, \tag{2}$$

where a_t implicitly shows the extra investment policies other than those that follow the historical pattern, while ε_t is zero mean random noise.

III. Critic Adaptation Stock Price Forecasting Architecture

The proposed hybrid view for financial time series prediction provides the basis for further technical analysis using the reinforcement learning philosophy. It is natural to conclude that price prediction can be done by solving two types of non-linear mapping problems, *i.e.* the next-step price derived from the market inertia and next-step price resulting from "irrational" investment behaviours at the given time step. The latter price implicitly shows the very recent extra investment policies other than those following historical patterns.

Supervised learning methods are an extremely useful tool to catch the system patterns that keep repeating. Therefore, they will be used to obtain the characteristic of market inertia. Given that the short-term performance of the forecasting architecture can be immediately measured, a type of direct reinforcement learning approach is utilized in order to directly explore "irrational" policies. It is noted that forecasting for future price is totally based upon historical price values, so that the selection and pre-processing for input variables is unnecessary. Let the output price series from the

forecasting architecture be denoted as $\tilde{p}_t$. The architecture will update its parameters at the end of each time interval t in order to forecast p_{t+1}.

The ANNs model is constructed to generalize the unobserved underlying "rational" price series rp_t from market inertia. The results will be $\tilde{rp}_t$. The recurrent reinforcement learning algorithm is used for the adaptation of the critic network so that the additional price series $\tilde{a}_t$ that approximates the unexpected investment behaviours alone in time can be obtained. Thus, the final forecasting result is as following:

$$\tilde{p}_{t+1} = \tilde{rp}_{t+1} + \tilde{a}_{t+1} \tag{3}$$

A standard multi-layer feedforward perceptron model trained by various backpropagation algorithms can be used to get a fairly accurate representation for the underlying function F through which the $\tilde{rp}_t$ can be mapped from the lagged observed price series $(p_{t-1}, \ p_{t-2}, \ \cdots, \ p_{t-k})$.

The decision function for $\tilde{a}_t$ is defined using the following format:

$$\tilde{a}_t = A\left(\tilde{a}_{t-1}, \theta_t, R_t\right), \tag{4}$$

where θ_t denotes the adjustable architecture parameters at time t and R_t is the observable of the system $\left\{p_{t-1}, p_{t-2}, p_{t-3}, \ldots; \tilde{rp}_{t-1}, \tilde{rp}_{t-2}, \tilde{rp}_{t-3}, \ldots; z_t, z_{t-1}, z_{t-2}, \ldots\right\}$.

z_t denotes any other external variables needed. The specific form of A at the starting point for a real problem can be set up by regression analysis based on a small group of training data, *e.g.* a simple autoregressive form may be valid as:

$$\tilde{a}_t = u\,\tilde{a}_{t-1} + v_0 p_{t-1} + v_1 p_{t-2} + \ldots w_0\,\tilde{rp}_{t-1} + w_1\,\tilde{rp}_{t-2} + \ldots + x, \tag{5}$$

Here, θ_t are the adjustable weights $\{u; v_1, v_2, \ldots; w_1, w_2, \ldots; x\}$. Such a format is used in the simulations described in the next section.

Financial series data are highly time-variant. Ultimately, minimizing the Mean Square Error is not necessary the most important objective because the goal of investors is to maximize the profit gain, or catch the market trend. The cost-to-go function needs to be designed properly in order to reflect such a trade-off. To accomplish this, we propose a new one-step utility function $U(\bullet)$, named the *policies-matching ratio,* as the reinforcement signal for the critic network:

$$U_t = \mu_1 \exp\left(-\left(\frac{\tilde{a}_{t-1} - \Delta p}{\sigma}\right)^2\right) + \mu_2 \frac{\left(\exp\left(\tilde{a}_{t-1}\Delta p\right) - \exp\left(-\tilde{a}_{t-1}\Delta p\right)\right)}{\left(\exp\left(\tilde{a}_{t-1}\Delta p\right) + \exp\left(-\tilde{a}_{t-1}\Delta p\right)\right)}, \tag{6}$$

$$\Delta p = p_{t-1} - \tilde{rp}_{t-1} \tag{7}$$

The ratio is constructed by considering the ability of series $\tilde{a}_t$ to both minimize prediction error (represented by the first Gaussian function term) and follow the market trend (demonstrated by latter hyperbolic tangent sigmoid term). Note that the weights $0 \le \mu_1 \le 1$ and $0 \le \mu_2 \le 1$ illustrate the degree of preference toward the two optimization objectives mentioned above. Their values depend upon the application emphasis.

The performance criterion at time t can now be expressed as a function of the sequence of utility.

$$J\left(U_t, U_{t-1,\ldots,} U_2, U_1\right) \tag{8}$$

Normally, the following format for the cost-to-go function will be used:

$$J_T = \sum_{k=1}^{T} \gamma^k U_k \tag{9}$$

For a decision function $A(\theta_t)$, the learning processes need to be implemented so that θ can be updated to maximize J_T. The RRL algorithm [8] is adopted for direct reinforcement.

$$\frac{dJ_T(\theta)}{d\theta} = \sum_{k=1}^{T} \frac{dJ_T}{dU_k}\left\{\frac{dU_k}{dA_k}\frac{dA_k}{d\theta} + \frac{dU_k}{dA_{k-1}}\frac{dA_{k-1}}{d\theta}\right\} \tag{10}$$

The gradient ascent is:

$$\Delta\theta = \alpha \frac{dJ_T(\theta)}{d\theta}, \tag{11}$$

where $0 \le \alpha \le 1$ is the learning rate, with a small value being preferred.

By considering only the most recently utility U_t, the on-line stochastic optimization can be obtained:

$$\frac{dJ_t(\theta)}{d\theta_t} \approx \frac{dJ_t}{dU_t}\left\{\frac{dU_t}{dA_t}\frac{dA_t}{d\theta_t} + \frac{dU_t}{dA_{t-1}}\frac{dA_{t-1}}{d\theta_{t-1}}\right\} \tag{12}$$

More specifically, by adopting the proposed *policies-matching ratio,* a more simple form follows:

$$\frac{dJ_t(\theta)}{d\theta_t} \approx \frac{dJ_t}{dU_t}\frac{dU_t}{dA_{t-1}}\frac{dA_{t-1}}{d\theta_{t-1}} \tag{13}$$

$$\Delta\theta_t = \alpha \frac{dJ_t(\theta_t)}{d\theta_t} \tag{14}$$

$$\frac{dA_t}{d\theta_t} \approx \frac{\partial A_t}{\partial \theta_t} + \frac{\partial A_t}{\partial A_{t-1}}\frac{dA_{t-1}}{d\theta_{t-1}} \tag{15}$$

Equations (13)-(15) constitute the critic network adaptive learning algorithm.

Figure 1 presents the architecture for the whole forecast learning process. Basically, the complete adaptive learning process for the proposed architecture will consist of two groups of parameter updating, *i.e.* θ for the critic network and the weights W for the ANN model. Obviously the most recent "unexpected" investment polices keep changing over time. Moreover, the unobserved market inertia is also likely to change frequently. To account for this assumption, a rolling-training process will be applied to both supervised and reinforcement leaning at the end of each time interval t.

For the ANN model trained by the backpropagation algorithm, the perfect data fitting for the training set can easily be obtained if sufficient lagged values are included and if the number of neurons in the hidden layer is also large. However, the optimal training balance is hard to get and the ANNs is well known for its tendency to over-fit. To overcome the over-training phenomenon common in supervised learning, a cross-validation procedure will be included.

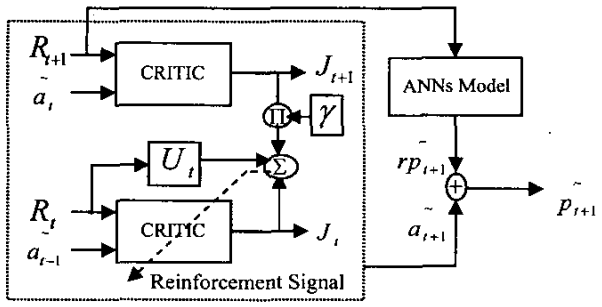

Fig.1. Critic Adaptation Forecasting Architecture for Stock Price

Note that here the exploration ability of architecture to search unknown investment strategies (which is crucial) is promised not only by the adopted on-line stochastic RRL optimization algorithm, but also by the characteristic of financial series: intrinsic noisy and uncertain. In the light of above reasons, the noise variable will not be incorporated in the reinforcement learning.

IV. Empirical Results

This section presents an overview of experimental work for applying the proposed critic adaptation forecasting architecture to predict future stock prices. Two daily stock index series (S&P 500 and NAS/NMS Composite) are studied. The goal is to forecast the corresponding index's next day closing price based simply upon its historical daily closing price. A 5-year test period starting from year 1998 to 2003 was selected and used for both cases.

The standard three-layer feedforward perceptron trained by the Levenberg-Marquardt algorithm is adopted to generalize the individual market inertia for each index. As mentioned in section 3, ANN models are prone to over-fitting. In order to construct relatively accurate ANN models without adding too much extra work, each case experimented with different values of k (the number of lagged values) ranging from 3 to 10 with an increment of 1, along with n (hidden neuron number) ranging from 2 to 8 with an increment of 2. Such a combination resulted in 32 candidate models. The method for choosing k and n employed standard cross-validation verification. A small group of training data (20 daily closing prices for S&P 500 started from 11-Feb-98 and 20 daily closing prices for Nasdaq started from 1-Jun-98) will be available at the beginning of each experiment. In order to let the ANN model reflect the market inertia's changes as timely as possible, rolling-training and data resampling were implemented, *i.e.* 10 input-target pairs $(\{(p_{t-1}, p_{t-2}, \cdots p_{t-k}) \leftrightarrow (p_t)\})$ are sequentially picked up each time as the training data for the construction of the network models. As such, 10 data pairs were divided into a training set (7 pairs) and validation set (3 pairs). After the ANNs model was set up and the prediction for the next day was completed, the new actual value of the stock closing price is added to the training data once it becomes available. The oldest input-target pair is eliminated. Then the training process is restarted. Throughout the experiment, most of the time the selected structure for the S&P 500 has 7 hidden layer units and 3 lagged values, while the structure for Nasdaq has 2 hidden layer units and 3 lagged values.

Meanwhile, the adaptation for the parameter θ of the decision function A keeps repeating at the end of each day based on the new reinforcement signal. The learning episodes will be increased gradually until the approximate policy convergence is reached.

Figure 2 show the results for the S&P 500 within a certain time window. For the *policies-matching ratio*, $\mu_1 = 0.9$ and

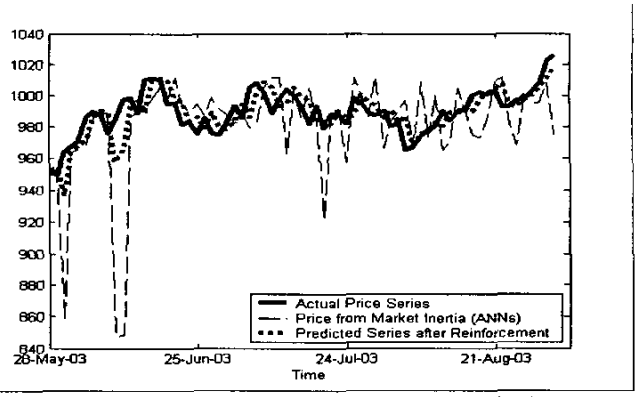

Fig.2. Time Series for S&P 500 Index Simulation within Certain Timeframe

$\mu_2 = 0.1$ are used for this experiment so the preference this case is to minimize Mean Square Error. Other settings include $\alpha = 0.01$, $\sigma = 0.95$, and number of episodes is equal to 500.

Assume investors are more interested in the market trend in terms of NAS/NMS Composite ($\mu_1 = 0.1$ and $\mu_2 = 0.9$ is used to address this assumption). Figure 3 presents the related simulation results. Here, $\alpha = 0.005$, $\sigma = 0.75$, and number of episodes for obtaining stable policy is 450.

The extra investment decisions, other than just following the market inertia, are evolved alone over time. The same evolution occurs for its implicit representation θ. Figure 4 uses v_0 as an example to demonstrate such evolution. Figure 4 also shows the difference between the unstable learning policy (400 learning episodes) and the convergent policy learning (500 learning episodes).

More precise comparison results are reported in Table 1 in which five performance measures are used for a total of 1,268 predicted daily prices. These measures include Root Mean Squares Error *RMSE* , Mean Absolute Error *MAE* , Mean Absolute Percentage Error *MAPE* , and the Direction Accuracy Indicators *DA*1 and *DA*2 (the correction percentage forecast for an up and down market). Note that the test goal for the S&P 500 is RMSE reduction and for the Nasdaq is market profitability. The definitions for *DA*1 and *DA*2 are as following:

$$DA1 = \frac{1}{T}\sum I\left[\left(p_t - p_{t-1}\right)\left(\tilde{p}_t - p_{t-1}\right)\right] \tag{16}$$

$$DA2 = \frac{1}{T}\sum I\left[\left(p_t - p_{t-1}\right)\left(\tilde{p}_t - \tilde{p}_{t-1}\right)\right] \tag{17}$$

where $I(x) = 1$ if $x > 0$ and $I(x) = 0$ if $x \le 0$.

The most important observation from Table 1 is that in either case the proposed hybrid learning model can correctly forecast market direction over 50 percent of the time. Such timing ability should result in market profitability.

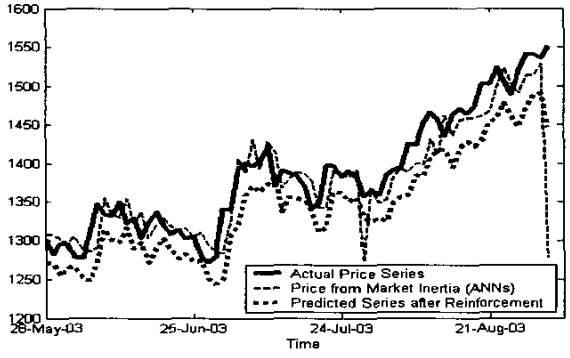

Fig.3. Time Series for Nasdaq Index Simulation within Certain Timeframe

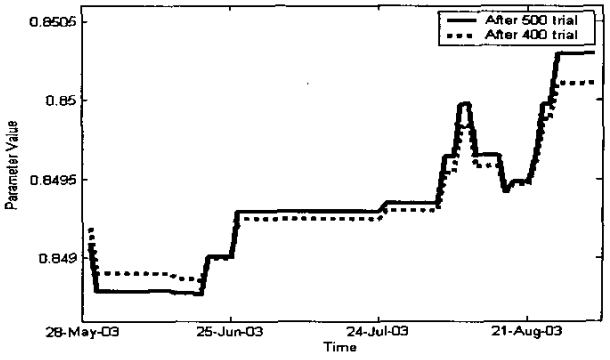

Fig. 4. Example of θ's Evolution within Certain Timeframe

TABLE 1: Comparison of Performance for Different Forecasting Architectures

ANNs Model vs. Hybrid Learning	S&P 500	Nasdaq
RMSE	1093.1 / 91.39	1815.1 / 2109.9
MAE	19.26 / 6.71	24.85 / 40.64
MAPE	1.95% / 0.78%	1.76% / 2.89%
DA1	50.75% / 51.76%	44.78% / 61.22%
DA2	41.79% / 53.3%	41.79% / 79.76%

V. Conclusions

This paper presents a novel hybrid view for predicting series-based financial data in which a reinforcement learning philosophy is used to solve a financial forecasting problem. Previous work [9] showed that the prediction model using value function-based reinforcement can outperform pure supervised learning type methods. In this paper, the more robust and practical forecasting schema derived from adaptive critic designs are proposed. The "curse of dimensionality" is avoided via direct reinforcement and the policies learned tend to be stable. A new utility function is also provided to reflect the requirements for both error reduction and market profitability. The experimental results demonstrate the compelling advantages offered by this research. To our knowledge, the critic adaptation financial series forecasting architecture that we propose here is unique. More research base upon this framework is now in progress.

References

[1] P.G McCluskey, "Feedforward and recurrent neural networks and genetic programs for stock market and time series forecasting," Technical Report CS-93-96, Brown University, 1993.

[2] R.J. Kuo, "A decision support system for the stock market through integration of fuzzy neural networks and fuzzy Delphi," *Applied Artificial Intelligence*, vol. 6, pp. 501-520, 1998.

[3] Marco Costantino, Russell J. Collingham and Richard G. Morgan, Qualitative Information in Finance: Natural language Processing and Information Extraction, University of Durham, UK, 1996.

[4] Jung-Hua Wang and Jia-Yann Leu, "Stock market trend prediction using ARIMA-based neural networks," *IEEE International Conference on Neural Networks*, 1996, Vol. 4, pp. 2160-2165.

[5] M.J. Brennan, E.S. Schwartz and R. Lagnado, "Strategic asset allocation," *J. Economic Dynamics Contr.*, vol. 21, pp. 1377-1403, 1997.

[6] R.C.Merton, *Continuous-Time Finance*. Oxford, U.K.: Blackwell, 1990.

[7] J.C.Cox, S.A.Ross and M.Rubinstein., "Option pricing: A simplified approach," *J. Financial Economics*, vol. 7, pp. 229-263, Oct. 1979.

[8] J.Moody, L.Wu, Y.Liao and M.Saffell., "Performance functions and reinforcement learning for trading systems and portfolios," *J. Forecasting*, vol. 17, pp. 441-470, 1998.

[9] Hailin Li and Cihan H. Dagli., "Synthesis of reinforcement learning and artificial neural networks applied to forecast real time financial series," *24th ASEM annual conference*, St.louis, USA, 2003, pp. 493-499.

[10] T.X.Brown, "Policy vs. value function learning with variable discount factors," *Proc. NIPS 2000 Workshop Reinforcement Learning: Learn the Policy or Learn the Value Function?*, Dec. 2000.

[11] W.T.Miller, R.S.Sutton and P.J.Werbos, *Neural Networks for Control*, Cambridge, MA: MIT Press, 1990.

[12] D.Prokhorov, R.Santiago and D.Wunsch., "Adaptive critic designs: A case study for neurocontrol," *Neural Networks*, vol. 8, pp. 1367-1372, 1995.