Reinforcement Learning for Automated Financial Trading: Basics and Applications

Francesco Bertoluzzo[1] and Marco Corazza[1,2]

[1] Department of Economics, Ca' Foscari University of Venice
[2] Advanced School of Economics of Venice
Sestiere Cannaregio n. 873, 30121 Venice, Italy
{fbertoluzzo,corazza}@unive.it

Abstract. The construction of automated financial trading systems (FTSs) is a subject of high interest for both the academic environment and the financial one due to the potential promises by self-learning methodologies. In this paper we consider Reinforcement Learning (RL) type algorithms, that is algorithms that real-time optimize their behavior in relation to the responses they get from the environment in which they operate, without the need for a supervisor. In particular, first we introduce the essential aspects of RL which are of interest for our purposes, second we present some original automatic FTSs based on differently configured RL-based algorithms, then we apply such FTSs to artificial and real time series of daily stock prices. Finally, we compare our FTSs with a classical one based on Technical Analysis indicators. All the results we achieve are generally quite satisfactory.

Keywords: Financial trading system, Reinforcement Learning, stochastic control, *Q*-learning algorithm, Kernel-based Reinforcement Learning algorithm, financial time series, Technical Analysis.

1 Introduction

According to the well-known weak form of the *Efficient Market Hypothesis* (EMH), it is not possible to systematically make profitable trading in financial markets. In fact, following this theory, the economic agents operating in these markets are rational, that is, through the law of the demand and the supply, they are able to instantaneously and appropriately vary the prices of the financial assets on the basis of the past and the current information. In this theoretical framework, the only source of (unpredictable) variations of the financial asset prices between two consecutive time instants can be the arrival of unexpected new information (see for more details [7]).

But, as the common sense suggests, human beings (and therefore economic agents) are often non rational when making decisions under uncertainty. In fact, since the 80s of the past century experimental economists have documented several departures of the real investors' behaviours from the ones prescribed by the EMH (see [12] and the references therein). The main implication coming from these departures consists in the fact that financial markets are not so rarely inefficient, and consequently that they offer, more or less frequently, possibilities of profitable trading.

S. Bassis et al. (eds.), *Recent Advances of Neural Network Models and Applications*,
Smart Innovation, Systems and Technologies 26,
DOI: 10.1007/978-3-319-04129-2_20, © Springer International Publishing Switzerland 2014

At this point, a (crucial) question naturally arises: How to take advantage of these possibilities of trading? Of course, the answer depends on the chosen reference theoretical framework. In our opinion, the currently most convincing attempt to reconcile the EMH with the empirical departures from it is given by the so-called *Adaptive Market Hypothesis* (AMH). Following this theory, a financial market can be viewed as an evolutionary environment in which different "species" (for instance, hedge funds, market makers, pension funds, retail investors, ...) interact among them in accordance with unknown and structurally time-varying dynamics in order to achieve the efficiency (see for more details [8] and [12]). Note that this evolutionary tending towards efficiency is not instantaneous and that it generally does not imply appropriate variations of the financial asset prices. Because of that, the AMH entails that ≪[f]*rom an evolutionary perspective, the very existence of active liquid financial markets implies that profit opportunities must be present. As they are exploited, they disappear. But new opportunities are also constantly being created as certain species die out, as others are born, and as institutions and business conditions change*≫ (from [12], page 24). So, coming back to the above question, it should be clear that an effective financial trading system (FTS) has to be a new specie:

- Able to real-time interact with the considered financial market in order to learn its unknown and structurally time-varying dynamics;
- Able to exploit this knowledge in a not emotive way in order to real-time detecting profitable financial trading policies.

Therefore, given the reference theoretical framework we have chosen (i.e. the AMH) and given the features of the FTS we have required in the two previous points, in this paper we resort to a self-adaptive machine learning known as *Reinforcement Learning* (RL) (see [1]), also known as *Neuro-Dynamic Programming* (see [4]), in order to develop a fully automated FTS. In particular, to this end we consider two different RL-based policy evaluation approaches: the *Temporal Difference* (TD) (see subsection 3.3) and the *Kernel-based Reinforcement Learning* (KbRL) (see subsection 3.4). Note also that, from a more traditional standpoint, the implementation of such a FTS can be viewed as a stochastic control problem in which the RL-based approaches have to discover the optimal financial trading strategies directly interacting with the financial market (so, any need to build a priori formal models for the description of the dynamics of the prices and/or the returns is eliminated).

The remainder of the paper is organized as follows. In the next Section we give a brief review of the prominent literature on the RL-based FTSs, and we describe the elements of novelties we present in our paper with respect to this literature. In Section 4 we present the essential aspects of the RL methodology which are of interest to our purposes. In Section 3 we present our RL-based FTSs, we provide the results of their applications to an artificial time series and to five real ones, and we compare these results with those coming from a classical FTS. In section 5 we propose some concluding remarks.

2 A Brief Review of the Literature

In this Section we provide a few review of the most influential papers about the RL-based FTSs. Our purpose is not primarily to be exhaustive, but rather that to highlight the main research directions. Then we describe the elements of novelties present in our paper.

Among the first contributions in this research field, we recall [13], [14], and [9]. In general, the respective Authors show that RL-based financial trading policies perform better than those based on supervised learning methodologies when market frictions are considered. In [13] a simplified version of the RL methodology, called *Direct Learning*, is proposed and used in order to set a FTS that, taking into account transaction costs, maximizes an appropriate investor's utility function based on a differential version of the well-known Sharpe ratio. Then, it is shown by controlled experiments that the proposed FTS performs better than standard FTSs. Finally, the Authors use the so developed FTS to make profitable trades with respect to assets of the U.S. financial markets. In [14], the Authors mainly compare FTSs developed by using various RL-based approaches with FTSs developed by using stochastic dynamic programming methodologies. In general they show by extensive experiments that the former approaches are better than the latter ones. In [9] the Author considers a FTS similar to the one developed in [13] and applies it to the financial high-frequency data, obtaining profitable performances. Also in [3] the Authors consider a FTS similar to the one developed in [13], but they consider as investor's utility function based on the differential version of the returns weighted direction symmetry index. Then, they apply this FTS to some of the most relevant world stock market indexes achieving satisfactory results. In [16] the Authors proposes a RL-based asset allocation strategy able to utilize the temporal information coming from both a given stock and the fund over that stock. Empirical results attained by applying such asset allocation strategy to the Korean stock market show that it performs better than several classical asset allocation strategies. In [10] two stock market timing predictors are presented: an actor-only RL and and actor-critic RL. The Authors show that, when both are applied to real financial time series, the latter generally perform better than the former. Finally, in [2] an actor-critic RL-based FTS is proposed, but in a fuzzy version. The Authors show that, taking into account transaction costs, the profitability of this FTS when apply to important world stock market indexes is consistently superior to that of other advanced trading strategies.

With respect to the this literature, in this paper we do the following:

- As introduced in Section 1, we consider two different RL-based trading policy evaluation approaches: the TD and the KbRL ones. Generally, the *squashing function* used in the former is the well-known S-shaped logistic. In this paper we have substitute it with another well-known S-shaped squashing function, the hyperbolic tangent, in order to check its performances. Notice that, to the best of our knowledge, the hyperbolic tangent has been rarely used in this context. Then, as regards the KbRL, again to the best of our knowledge, it has never been used before for developing FTSs;
- In several papers the differential Sharpe ratio is used as performance indicator. In this paper we utilize the classical Sharpe ratio computed on the last $L \in \mathbb{N}$ trading days. This surely reduce the computational effort;

- Usually, the very very big majority of the classical FTSs and of the advanced ones consider two signals or actions: “sell” or equivalently “stay-short-in-the-market”, and “buy” or equivalently “stay-long-in-the-market”. In this paper we consider also a third signal or action: “stay-out-from-the-market”. By doing so, we give to our RL-based FTS the possibility to take no position, or to leave a given position, when the buy/sell signal appears weak. Note that the set of the action is finite and discrete;
- As state descriptors of the considered financial market we consider the simple current and some past returns. Generally, this is not the case for several FTSs. Anyway, we have made this choice in order to check the performance capability of our FTS also starting from not particularly sophisticated information. Notice that any state descriptor is a continuous variable over $\mathbb{R}$.

3 Some Basic on the RL

In this Section we present the basic aspects of the RL methodology which are of interest to our purposes.

To learn by directly interacting with the environment[1] without the need of a supervisor is likely the more immediate idea about the nature of the learning itself. The consequences of the actions[2] of the learning agent[3] lead she/he to choose what are the actions that allow to obtain the desired results and what are the ones to avoid. RL formalizes this kind of learning by maximizing a numerical reward[4] (see [1]). The agent has to discover which actions yield the most reward by trying them. RL is different from Supervised Learning (SL) in which the agent learns from past examples provided by an external supervisor. SL is an important kind of learning, but in interactive problems it is often impractical to obtain examples of desired current and future behavior that are representative of the situation in which the agent will have to act.

To formalize these first ideas, let us consider a system observed at discrete time in which the state[5] at time t, $s_t \in \mathscr{S}$, summarizes all information concerning the system available to the agent. In the RL framework it is assumed that the system satisfies the Markov property, that is that the probability of transition from the actual state s_t to the next one s_{t+1} depends only on the current state s_t. On the basis of s_t, the agent selects an action $a_t \in \mathscr{A}(s_t)$, where $\mathscr{A}(s_t)$ is the set of all the possible actions the agent can take given the state s_t. At time $t+1$ the agent receives a reward, $r(s_t, a_t, s_{t+1}) \in \mathbb{R}$, as consequence of her/his actions a_t and of the new state s_{t+1} in which she/he finds herself/himself. The reward is a numerical representation of the satisfaction of the agent. Generally, at time t the agent wish to maximize the expected value of some global return, $R(s_t)$, which is defined as function of the actual reward and of the future suitably discounted ones. This function can be written as:

[1] In our case a financial market.

[2] In our case stay-short-in-the-market, stay-out-from-the-market, and stay-long-in-the-market.

[3] In our case a FTS.

[4] In our case the performances of the FTS.

[5] In our case the current and some past returns.

$$R(s_t) = r(s_t, a_t, s_{t+1}) + \gamma r(s_{t+1}, a_{t+1}, s_{t+2}) + \gamma^2 r(s_{t+2}, a_{t+2}, s_{t+3}) + \ldots,$$

where $\gamma \in (0,1)$ is the discount factor.

In RL, a policy $\pi(s_t) = a_t$ is a mapping from states to actions defining the choice of action a_t given state s_t. In order to maximize the expected $R(s_t)$, RL searches for a suitable policy. Considering also the policy $\pi(\cdot)$, we can rewrite the global return as:

$$R^{\pi}(s_t) = r(s_t, \pi(s_t), s_{t+1}) + \gamma r(s_{t+1}, \pi(s_{t+1}), s_{t+2}) + \gamma^2 r(s_{t+2}, \pi(s_{t+2}), s_{t+3}) + \ldots.$$

3.1 The Value Functions

Generally, RL-based approaches[6] need the estimation of the so-called *value functions*. These functions of states (or of state-action pairs) attribute a value to each state s_t (or to each state-action pairs) proportional to the rewards achievable in the future from the current state s_t (or from the current state-action pairs). They evaluate how good is for the agent to be in a given state (or how good to perform a given action in a given state). The notion of "how good" is defined in terms of R_t. In particular, the value of a state $s_t = s$ following policy $\pi(\cdot)$ is the expected sum of the current and of the future discounted rewards when starting in state $s_t = s$, and thereafter following policy $\pi(\cdot)$, that is:

$$V^{\pi}(s) = \mathbb{E}\left[R^{\pi}(s_t) | s_t = s\right].$$

Similarly, the value of tacking action $a_t = a$ being in state $s_t = s$ under policy $\pi(\cdot)$ is the expected sum of the current and of the future discounted rewards starting from state $s_t = s$, taking action $a_t = a$, and thereafter following policy $\pi(\cdot)$, that is:

$$Q^{\pi}(s,a) = \mathbb{E}\left[R^{\pi}(s_t) | s_t = s, a_t = a\right].$$

A fundamental property of value functions is that they satisfy particular recursive relationships. In particular, for any policy $\pi(\cdot)$ and any state $s_t = s$, the following consistency condition holds between the value of s_t and the value of any possible successor state s_{t+1}:

$$V^{\pi}(s) = \mathbb{E}\left[r(s_t, \pi(s_t), s_{t+1}) + \gamma V^{\pi}(s_{t+1}) | s_t = s\right]. \qquad (1)$$

Equation (1) is the so-called *Bellman equation* for $V^{\pi}(s_t)$. It is possible to prove that the value $V^{\pi}(s)$ is the unique solution to its Bellman equation.

3.2 The Generalized Policy Iteration

The task of the RL methodologies consists in finding an optimal policy. The optimal policy identifies the values $V^*(s)$ and $Q^*(s,a)$ such that

$$V^*(s) = \max_{\pi} V^{\pi}(s) \text{ and } Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a) \ \forall \ s \in \mathscr{S} \text{ and } \forall \ a \ \in \mathscr{A}(s). \qquad (2)$$

[6] In our case the financial trading strategies.

Since $V^*(s)$ is the value function for a policy, it satisfy the Bellman equation (1). As it is also the optimal value function, $V^*(s)$'s Bellman condition can be written in a special form without reference to any specific policy. This form is the Bellman equation for $V^*(s)$, or the *Bellman optimality equation*, which expresses the fact that the value of a state under an optimal policy must equal the expected global return for the best action from the state itself, that is:

$$V^*(s) = \max_a Q^*(s,a) = \mathbb{E}\left[R^*(s_t)|s_t = s, a_t = a\right]. \tag{3}$$

With equivalent arguments, the Bellman optimality equation for $Q^*(s,a)$ is:

$$Q^*(s,a) = \mathbb{E}\left[r(s_t,a_t,s_{t+1}) + \gamma \max_{a'} Q^*(s_{t+1},a')|s_t = s, a_t = a\right].$$

At this point, it is possible to iteratively calculate the value function for a state (or for a state-action pair). Let $V_0^\pi(s_t)$ for all $s_t = s \in \mathscr{S}$ be an arbitrarily initialization of the state value function. Each successive approximation is obtained by using as follows the Bellman equation in an update rule:

$$\widehat{V}_{k+1}^\pi(s_t) = \mathbb{E}\left[r(s_t,a_t,s_{t+1}) + \gamma \widehat{V}_k^\pi(s_{t+1})\right] \ \vee s_t = s \in \mathscr{S}. \tag{4}$$

If the expectation $\widehat{V}_{k+1}^\pi(s_t)$ exists, then $\lim_{k \to +\infty} \widehat{V}_k^\pi(s) = V^\pi(s)$.

The reason for computing the value functions for a policy is to find better policies in order to increase the expected value of the global returns. This process is called *policy improvement.*

As described, the policy improvement process requires the evaluation of the previous policy. This evaluation can be made by (4), which is itself an iterative process that converges in the limit. Fortunately, there is no need to wait for the exact convergence or for a particularly high level of convergence. In fact one can stop the policy evaluation iteration in several ways without losing the convergence (see [1]). An important special case is when policy evaluation is stopped after just one step. In particular, it is possible to combine the evaluation process and the improvement one by stopping the policy evaluation process at each step and then by improving the policy itself. This mixed algorithm is called *generalized policy iteration.* It can be written as:

$$\widehat{V}_{k+1}^\pi(s_t) = \max_a \mathbb{E}\left[r(s_t,a,s_{t+1}) + \gamma \widehat{V}_k^\pi(s_{t+1})\right], \tag{5}$$

where $\widehat{V}_{k+1}^\pi(s_t)$ is the update estimate with the improved policy at step $k+1$ with respect to the previous estimate and the previous policy at step k.

In order to improve the policy we choose an approach, among the ones presented in literature, which may produce increasing of the global return in the long run. Following such an approach, the choice of the action at each time t is given by:

$$a_t = \begin{cases} \pi'(s_t) & \text{with probability } 1-\varepsilon \\ a \in \mathscr{A}(s_t) & \text{with probability } \varepsilon \end{cases},$$

where $\varepsilon \in (0,1)$ and $\pi'(s_t)$ is the candidate action which maximizes $Q^\pi(s,a)$.

In the next subsections we shortly introduce the two different approaches we use in our FTSs for calculating the expected value (5). Notice that we can not take into account other kinds of approaches like the *Dynamic Programming*-based ones and the *Monte Carlo*-based ones. In fact:

- The former need models to calculate the true probabilities of transition from a state to another one, whereas in financial trading such a model is generally not known or not available;
- The latter, in order to improve the policy, need to wait for until the end of all the trades, whereas a real FTS trades an a priori indefinite number of times.

3.3 TD Methods and the Q-Learning Algorithm

In this subsection we present a class of (trading) policy evaluation algorithms which update step by step the estimate of $V(s_t)$. First of all one puts in evidence that it is possible to write $\widehat{V}_{k+1}(s_t)$ in the following recursive way:

$$\begin{aligned}\widehat{V}_{k+1}(s_t) &= \frac{1}{k+1}\sum_{j=1}^{k+1} R_j(s_t) = \frac{1}{k+1}\left[R_{k+1}(s_t) + \sum_{j=1}^{k} R_j(s_t)\right] = \cdots = \\ &= \widehat{V}_k(s_t) + \alpha_k\left[R_{k+1}(s_t) - \widehat{V}_k(s_t)\right],\end{aligned}$$

where $\alpha_k = 1/(k+1)$ is the so-called learning rate.

The TD methods can update the estimate $\widehat{V}_{k+1}(s_t)$ as soon as the quantity

$$d_k = R_{k+1}(s_t) - \widehat{V}_k(s_t) = r(s_t, s_{t+1}) + \gamma\widehat{V}_k(s_{t+1}) - \widehat{V}_k(s_t),$$

becomes available. Therefore the above recursive relationship can be rewritten as:

$$\widehat{V}_{k+1}(s_t) = \widehat{V}_k(s_t) + \alpha_k\left[r(s_t, s_{t+1}) + \gamma\widehat{V}_k(s_{t+1}) - \widehat{V}_k(s_t)\right].$$

With regard to the convergence of the TD methods, it is possible to prove that, if the sufficient conditions $\sum_{k=1}^{+\infty}\alpha_k = +\infty$ and $\sum_{k=1}^{+\infty}\alpha_k^2 < +\infty$ hold, then $\lim_{k\to+\infty}\Pr\left\{\left|\widehat{V}_k(s_t) - \right.\right.$ $\left.\left.-V(s_t)\right| < \varepsilon\right\} = 1$, for any $\varepsilon > 0$ (see [4]).

The TD methods are "naturally" developed in an incremental and on-line fashion which make them particularly appealing for building FTSs.

In literature there exist several different TD methods, the most widespread of whom is the *Q-Learning algorithm* (QLa). The QLa is an *off*-policy control method, where "off" indicates that two different policies are used in the policy improvement process: A first one is used to estimate the value functions, and another one is used to control the improvement process. It is possible to prove that the so-obtained state-action value function is given by:

$$\widehat{Q}_{k+1}(s_t, a_t) = \widehat{Q}_k(s_t, a_t) + \alpha_k\left[r_{t+1} + \gamma\max_a \widehat{Q}_k(s_{t+1}, a_{t+1}) - \widehat{Q}_k(s_t, a_t)\right]. \qquad (6)$$

At this point we recall that the states of interest taken into account for the development of our FTSs are continuous variables over $\mathbb{R}$. In this case it is possible to prove that the value function at step k, $\widehat{V}_k(s_t)$, can now be approximated by a parameterized functional form with parameter vector θ_k at step k. It involves that the associated value function $\widehat{V}_k(s_t) = \widehat{V}_k(s_t;\theta_k)$ totally depends on θ_k, which varies step by step. In order to determine the optimal parameter vector, θ^*, which minimizes the "distance" between the unknown $V^\pi(s_t)$ and its estimate $\widehat{V}^\pi(s_t;\theta_k)$, in most machine learning approaches the minimization of the mean square error is used, that is:

$$\min_{\theta_k} \sum_s \left[V^\pi(s) - \widehat{V}^\pi(s;\theta_k) \right]^2 .$$

The convergence of θ_k to θ^* is proven for approximators characterized by simple functional forms like the affine ones (see [4]). Among the linear functional forms, in building our FTSs we use the following one:

$$\widehat{V}^\pi(s;\theta) = \sum_{i=1}^{n} \theta_i \phi_i(s_i) = \theta' \phi(s), \qquad (7)$$

where n is the number of considered states, and $\phi_i(\cdot)$ is a suitable transformation of the i-th state.

Under mild assumptions it is possible to prove that the update rule to use for estimating the state-action value function in the case of continuous states becomes:

$$\begin{aligned} d_k &= r(s_t,a_t,s_{t+1}) + \gamma \max_a \widehat{Q}(s_{t+1},a;\theta_k) - \widehat{Q}^\pi(s_t,a_t;\theta_k) \text{ and } \theta_{k+1} = \\ &= \theta_k + \alpha d_k \nabla_{\theta_k} \widehat{Q}^\pi(s_t,a_t;\theta_k). \end{aligned} \qquad (8)$$

3.4 The KbRL Algorithm

Another method for approximating $Q^\pi(s,a)$, alternative to the QLa and also usable in case of continuous states, is the nonparametric regression-based one known as KbRL algorithm (see [17] and [18]). Given a kernel $K(\cdot)$ and defined $q_t = (s_t \; a_t)$, the KbRL algortihm estimates $Q(s,a)$ as follows:

$$\widehat{Q}_k(q_t) = \sum_{i=1}^{t-1} p_i(q_t) \widehat{Q}_k(q_i),$$

where $p_i(q_t) = K\left(\frac{q_t - q_i}{h}\right) \Big/ \sum_{i=1}^{t-1} K\left(\frac{q_t - q_i}{h}\right)$.

Note that the choice of the kernel $K(\cdot)$ is not crucial, whereas the choice of the bandwidth h has to be done carefully (see [6]).

In this new reference methodological framework, the approximation relationship for $\widehat{Q}_{k+1}(q_t)$ is given by:

$$\widehat{Q}_{k+1}(q_i) = \widehat{Q}_k(q_i) + p_i(q_t) \left[\widehat{Q}_{k+1}(q_t) - \widehat{Q}_k(q_i) \right], \; i = 1,2,\ldots,t-1. \qquad (9)$$

This approach is particularly interesting because constitutes a kind of minimization method without derivatives (see [5]). Further, it is also usable in non-stationary contexts as the update relationship (9) is based on the current values of state-action pairs.

4 The RL-Based FTSs

In this Section we present our RL-based FTSs, we provide the results of their applications to an artificial time series and to five real ones, and we compare these results with those coming from a classical FTS.

4.1 Our RL-Based FTSs and the Applications

In this Subsection we use the QLa and the KbRL algorithm for developing daily FTSs. First one has to identify the quantities which specify the states, the possible actions of the FTSs, and the reward function.

With regards to the states, recalling that we are interested in checking the performance capability of our FTSs also starting from not particularly sophisticated information, we simply use as states the current and the past four percentage returns of the asset to trade. So, given the current price of the asset, p_t, the state of the system at the time t is given by the following vector:

$$s_t = \left(e_{t-4},\, e_{t-3},\, e_{t-2},\, e_{t-1},\, r_t\right),$$

where $e_\tau = \dfrac{p_\tau - p_{\tau-1}}{p_{\tau-1}}$.

Further, as we are also interested in checking the applicability of the considered RL-based methods to the development of effective FTSs, we do not consider frictional aspects.

Concerning the possible action of the FTS, we utilize the three following ones:

$$a_t = \begin{cases} -1 \text{ (sell or stay-short-in-the-market signal)} \\ \ \ 0 \text{ (stay-out-from-the-market signal)} \\ \ \ 1 \text{ (buy or stay-long-in-the-market signal)} \end{cases},$$

in which the stay-out-from-the-market implies the closing of whatever previously open position (if any)[7].

Finally, with reference to the reward function, following [14] we take into account the well known Sharpe ratio, that is:

$$Sr_t = \frac{\mathrm{E}_L\left[g_{t-1}\right]}{\sqrt{\mathrm{Var}_L\left[g_{t.1}\right]}},$$

where $\mathrm{E}_L(\cdot)$ and $\mathrm{Var}_L(\cdot)$ are, respectively, the sample mean operator and the sample variance one calculated in the last L stock market days, and $g_t = a_{t-1}e_t$ is the gained/lost percentage return obtained at time t as a consequence of the action undertaken by the FTS at time $t-1$.

In particular, as we wish an indicator that reacts enough quickly to the consequences of the actions of the considered FTSs, we calculated Sr_t only by using the last $L = 5$ stock market days (a stock market week), and the last $L = 22$ stock market days (a stock market month).

[7] Notice that in most of the prominent literature, like for instance in [14], only the sell and the buy signals are considered.

Now let us pass to the two RL-based approaches we consider: The QLa and the KbRL algorithm. With respect the QLa, the kind of linear approximator of the state-action value function we choose is:

$$Q(s_t, a_t; \theta_k) = \theta_{k,0} + \sum_{n=1}^{5} \theta_{k,n} \tanh(s_{t,n}) + \theta_{k,6} \tanh(a_t),$$

in which $\tanh(\cdot)$ plays the role of transformer of the state.
Then, the ε-greedy function we consider is:

$$a_t = \begin{cases} \arg\max_{a_t} Q(s_t, a_t; \theta_k) & \text{with probability } 1 - \varepsilon \\ u & \text{with probability } \varepsilon \end{cases},$$

in which $\varepsilon \in \{2.5\%, 5.0\%, 7.5\%\}$, and $u \sim \mathscr{U}_d(-1, 1)$.

Concerning the KbRL algoritm, we follow [6] and [17] (a reasonable setting of the bandwidth h has be done through some trial-and-error experiments).

Summarizing, we consider two different RL-based approaches (QLa and KbRL algorithm), three different values of ε (2.5%, 5.0% and 7.5%), and two different values of L (5 and 22), for a total of twelve different configurations.

We apply the so specified RL-based FTSs to six different time series of daily prices: An artificial one and five real ones. With reference to the artificial time series, as in [14] we generate log-price series as a random walks with autoregressive trend processes. In particular, we used the model:

$$p_t = \exp\left\{ \frac{z_t}{\max z - \min z} \right\},$$

where $z_t = z_{t-1} + \beta_{t-1} + 3a_t$, in which $\beta_t = 0.9\beta_{t-1} + b_t$, $a_t \sim \mathscr{N}(0, 1)$, and $b_t \sim \mathscr{N}(0, 1)$. The length of the so-generated series is $T = 5{,}000$.
This artificial price series shows features which are often present in real financial price series. In particular, it is trending on short time scales and has a high level of noise.

As far as the real time series regards, we utilize the closing prices of Banca Intesa, Fiat, Finmeccanica, Generali Assicurazioni, and Telecom Italia (from the Italian stock market), from January 1, 1973 to September 21, 2006. The length of these series is $T = 5{,}400$, more than 21 stock market years.

At this point we can present the results of the applications of the variously configured FTSs. In all the applications we set $\alpha = 0.8$ and $\gamma = 0.7$.

In figure 1 we graphically report the results of the application of the QLa-based FTS to the real price series of Banca Intesa, with $\varepsilon = 5\%$ and $L = 5$. In particular: The first panel shows the price series, the second panel shows the actions taken by the FTS at each time, the third panel shows the rewards, that is the Sharpe ratios, at each time, and the fourth panel shows the cumulative return one should obtain by investing the same monetary amount at each time. At the end of the trading period, $t = T$, the cumulative return is 271.42%.

It is important to note that at the beginning of the trading period, $t = 0$, the vector of the parameters used in the linear approximator, θ_k, is randomly initialized. Because of it, by repeating more times the same application we observe a certain variability in the final cumulative return. With regards to the application to the real price series of Banca

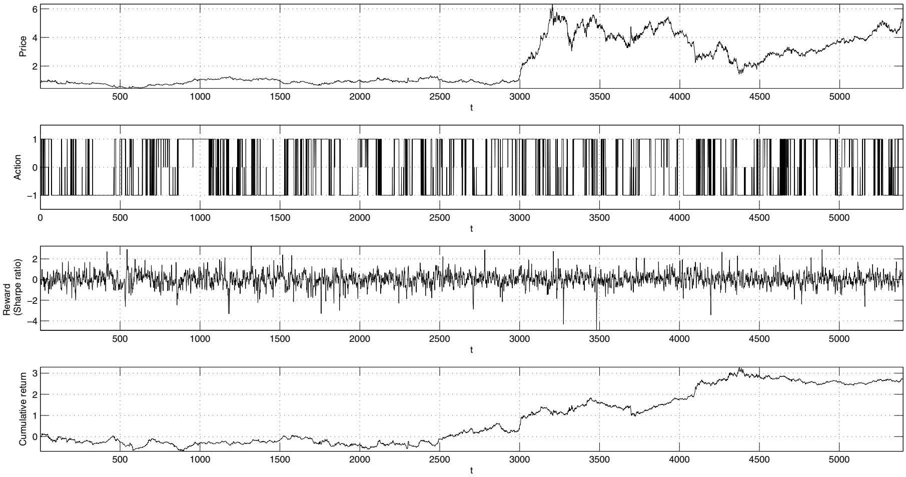

Fig. 1. Results of the *Q*La-based FTS applied to the Banca Intesa price series, with $\varepsilon = 5\%$ and $L = 5$. Final cumulative returns: 271.42%.

Intesa, some of the final cumulative returns we obtain are: -158.52%, 174.91%, and -26.38%. This shows that the influence of the random initialization heavily spreads overall the trading period instead to soften as time step increases. To check the effects of this random initialization, we repeated 1,000 times the application of each of the investigated configurations. When the application was to the artificial time series, the series has been taken the same in all the repetition. In tables 1 to 4 we report some statistics concerning the final cumulative returns.

With reference to the KbRL algorithm, we obtain results similar to the one related to the *Q*La, although a bit less performing. In figure 2 we graphically report the results of the application of the KbRL-based FTS to the real price series of Fiat, with $\varepsilon = 7.5\%$ and $L = 22$ (at the end of the trading period, $t = T$, the cumulative return is 135.47%). In particular, note that for the KbRL algorithm there is not parameter vector to randomly initialize, but at the beginning of the trading period, $t = 0$, it is necessary to randomly initialize a suitable set of state-action pairs (see [17]). So, also in this approach one puts the question of the variability of the results. As for the *Q*La, for the KbRL algorithm too we repeated 1,000 times the application of each of the investigated configurations (see tables 1 to 4 for some statistics about the final cumulative returns).

The main facts detectable from tables 1 to 4 are the following ones:

- Given the values of the means:
  - ○ Generally, the investigated configurations appears to be profitable, in fact the most of the means (88.89%) are positive. Further, the *Q*La approach seems more performing than the KbRL one;
  - ○ It appears that the value of L has a significant impact on the performances of the FTSs. In particular, with reference to the artificial time series, all the checked FTS configurations are better performing when $L = 5$, whereas, with reference

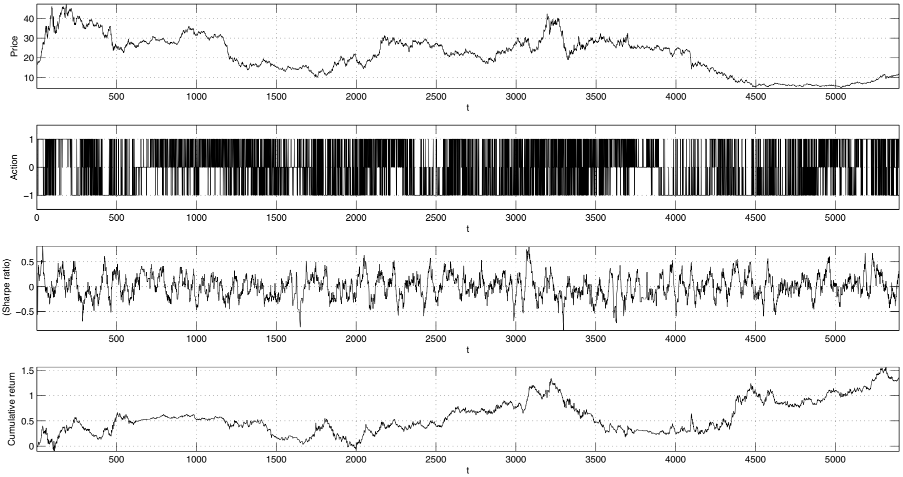

Fig. 2. Results of the KbRL-based FTS applied to the Fiat price series, with $\varepsilon = 7.5\%$ and $L = 22$. Final cumulative returns: 135.47%.

  to all the real time series, the most of the checked FTS configurations (75.00%) are better performing when $L = 22$;
  - It appears that the value of ε too has a significant impact on the performances of the FTSs. In particular, although with reference to the artificial time series empirical regularities does not come into view, with reference to all the real time series, the most of the checked FTS configurations (85.00%) are better performing when $\varepsilon = 7.5\%$;
- Given the values of the standard deviations, the results of all the considered variously configurated FTSs are characterized by a certain level of variability. In particular, such values emphasize that the question of the influence of the random initialization on the results is mainly true for the real financial time series.

4.2 The Comparison

In this Subsection we compare the results coming from our RL-based FTSs with those coming from a classical FTS based on Technical Analysis (TA) indicators. The choice of the latter FTS as term of comparison is due to (at least) two reasons: first, TA, together with Fundamental Analysis, is one of the most used "toolbox" among professionals and practitioners for the building of FTSs; second, in the last 10-15 years also the academic world has begun to recognize the soundness of some of the features related to the TA (see [11]).

Table 1. Some statistics about the final cumulative returns

Method	ε	L	Statistics	Artificial	Banca Intesa	Fiat
				time series	time series	time series
*Q*La	2.5%	5	μ	443.62%	−3.77%	40.58%
			σ	37.83%	127.48%	139.35%
			Conf. interval	[369.48%, 517.76%]	[−253.53%, 246.08%]	[−232.55%, 313.70%]
*Q*La	2.5%	22	μ	307.03%	70.85%	101.21%
			σ	43.02%	146.34%	144.27%
			Conf. interval	[222.71%, 391.35%]	[−215.97%, 357.68%]	[−181.55%, 383.98%]
*Q*La	5.0%	5	μ	472.02%	40.00%	91.26%
			σ	32.79%	139.09%	137.30%
			Conf. interval	[407.76%, 536.29%]	[−226.73%, 306.73%]	[−177.85%, 360.37%]
*Q*La	5.0%	22	μ	337.68%	92.92%	124.69%
			σ	40.80%	149.28%	144.89%
			Conf. interval	[257.71%, 417.64%]	[−199.66%, 385.50%]	[−159.29%, 408.68%]
*Q*La	7.5%	5	μ	467.48%	49.15%	114.41%
			σ	31.11%	139.71%	142.42%
			Conf. interval	[406.50%, 528.46%]	[−224.67%, 322.97%]	[−164.73%, 393.56%]
*Q*La	7.5%	22	μ	339.04%	99.66%	133.81%
			σ	39.60%	146.74%	147.53%
			Conf. interval	[261.43%, 416.66%]	[−187.95%, 387.28%]	[−155.34%, 422.96%]

Table 2. Some statistics about the final cumulative returns

Method	ε	L	Statistics	Artificial	Banca Intesa	Fiat
				time series	time series	time series
KbRL	2.5%	5	μ	483.71%	−26.16%	73.33%
			σ	60.10%	141.37%	144.00%
			Conf. interval	[365.91%, 601.51%]	[−303.25%, 250.93%]	[−208.90%, 355.56%]
KbRL	2.5%	22	μ	237.64%	15.90%	75.37%
			σ	63.55%	159.92%	131.08%
			Conf. interval	[113.09%, 362.20%]	[−297.54%, 329.34%]	[−181.55%, 332.28%]
KbRL	5.0%	5	μ	435.42%	−7.99%	77.32%
			σ	41.13%	131.26%	130.42%
			Conf. interval	[354.81%, 516.02%]	[−265.26%, 249.28%]	[−178.30%, 332.95%]
KbRL	5.0%	22	μ	216.61%	13.20%	71.98%
			σ	49.98%	153.34%	136.04%
			Conf. interval	[118.64%, 314.58%]	[−287.34%, 313.74%]	[−194.66%, 338.61%]
KbRL	7.5%	5	μ	401.76%	−0.63%	67.55%
			σ	39.80%	130.40%	128.00%
			Conf. interval	[323.75%, 479.77%]	[−256.23%, 254.96%]	[−183.33%, 318.42%]
KbRL	7.5%	22	μ	197.78%	35.36%	74.82%
			σ	42.55%	143.51%	132.06%
			Conf. interval	[114.38%, 281.18%]	[−245.91%, 316.63%]	[−184.02%, 333.65%]

Table 3. Some statistics about the final cumulative returns

Method	ε	L	Statistics	Finmeccanica time series	Generali Assicurazioni time series	Telecom Italia time series
*Q*La	2.5%	5	μ	177.03%	1.04%	288.79%
			σ	171.97%	99.12%	168.71%
			Conf. interval	[−160.03%, 514.09%]	[−193.24%, 195.31%]	[−41.88%, 619.46%]
*Q*La	2.5%	22	μ	161.77%	31.70%	246.44%
			σ	176.86%	103.50%	165.71%
			Conf. interval	[−184.87%, 508.40%]	[−171.15%, 234.55%]	[−78.35%, 571.23%]
*Q*La	5.0%	5	μ	269.77%	2.69%	334.86%
			σ	174.92%	101.54%	167.66%
			Conf. interval	[−73.08%, 612.62%]	[−196.33%, 201.70%]	[6.25%, 663.48%]
*Q*La	5.0%	22	μ	217.58%	30.60%	256.90%
			σ	173.14%	110.94%	179.84%
			Conf. interval	[−121.77%, 556.93%]	[−186.85%, 248.05%]	[−95.59%, 609.39%]
*Q*La	7.5%	5	μ	295.33%	21.69%	335.63%
			σ	163.18%	104.54%	163.96%
			Conf. interval	[−24.50%, 615.16%]	[−183.21%, 226.59%]	[14.26%, 657.00%]
*Q*La	7.5%	22	μ	227.32%	47.56%	279.09%
			σ	168.45%	110.49%	168.09%
			Conf. interval	[−102.85%, 557.49%]	[−169.00%, 264.12%]	[−51.54%, 609.73%]

Table 4. Some statistics about the final cumulative returns

Method	ε	L	Statistics	Finmeccanica time series	Generali Assicurazioni time series	Telecom Italia time series
KbRL	2.5%	5	μ	61.20%	−13.35%	77.46%
			σ	165.83%	122.98%	190.70%
			Conf. interval	[−263.81%, 386.22%]	[−254.40%, 227.69%]	[−296.31%, 451.22%]
KbRL	2.5%	22	μ	6.41%	−26.78%	26.82%
			σ	170.46%	126.01%	183.79%
			Conf. interval	[−327.68%, 340.50%]	[−273.76%, 220.20%]	[−333.40%, 387.05%]
KbRL	5.0%	5	μ	80.34%	10.77%	93.43%
			σ	162.01%	120.89%	187.08%
			Conf. interval	[−237.19%, 397.87%]	[−226.17%, 247.71%]	[−273.25%, 460.12%]
KbRL	5.0%	22	μ	29.58%	−9.44%	42.94%
			σ	170.01%	120.57%	188.15%
			Conf. interval	[−303.65%, 362.81%]	[−245.75%, 226.88%]	[−325.82%, 411.71%]
KbRL	7.5%	5	μ	107.97%	20.62%	102.27%
			σ	155.40%	119.64%	189.44%
			Conf. interval	[−196.61%, 412.55%]	[−213.89%, 255.12%]	[−269.04%, 473.57%]
KbRL	7.5%	22	μ	38.24%	−3.31%	34.85%
			σ	158.94%	121.46%	190.09%
			Conf. interval	[−273.28%, 349.75%]	[−241.38%, 234.75%]	[−337.73%, 407.43%]

The TA-based FTS we consider as benchmark utilizes the following five classical IF-THEN-ELSE rules:

Rule 1

IF $EMA3 > EMA12$ THEN (buy OR stay-long-in-the-market)
ELSE IF $EMA3 < EMA12$ THEN (sell OR stay-short-in-the-market)
ELSE stay-out-of-the-market

where *EMAm* is the exponential moving average calculated by using the last *m* daily stock prices;

Rule 2

IF $(MACD > 0$ AND $MACD > Signal\ line)$ THEN (buy OR stay-long-in-the-market)
ELSE IF $(MACD < 0$ AND $MACD < Signal\ line)$ THEN (sell OR stay-short-in-the-market)
ELSE stay-out-of-the-market

where $MACD = EMA12 - EMA26$ is the moving average convergence/divergence, and $Signal\ line = EMA9$;

Rule 3

IF $RSI < 30$ THEN (buy OR stay-long-in-the-market)
ELSE IF $RSI > 70$ THEN (sell OR stay-short-in-the-market)
ELSE stay-out-of-the-market

where *RSI* is the relative strength index calculated by using the last 14 daily stock prices;

Rule 4

IF $ROC < -1$ THEN (buy OR stay-long-in-the-market)
ELSE IF $ROC > 1$ THEN (sell OR stay-short-in-the-market)
ELSE stay-out-of-the-market

where *ROC* is the rate of change calculated by using the last 12 daily stock prices;

Rule 5

IF $TSI > 0$ THEN (buy OR stay-long-in-the-market)
ELSE IF $TSI < 0$ THEN (sell OR stay-short-in-the-market)
ELSE stay-out-of-the-market

where TSI is the true strength index calculated by using the last 13 and the last 25 daily stock prices.

If three or more of the listed rules "propose" the same action then the considered TA-based FTS takes that action, else it stays-out-of-the-market (see for more details on TA [15]).

In table 5, in the second row we report the results (in terms of final returns) of the applications of the specified TA-based FTS to the six time series of daily stock prices considered in Subsection 4.1, and in the third row we report, for each of the same time series, the percentage of times in which the variously configured RL-based FTSs have over-performed the TA-based FTS.

Table 5. Results about the comparison

	Artificial time series	Banca Intesa time series	Fiat time series	Finmeccanica time series	Generali A. time series	Telecom Italia time series
Final return	2.11%	2.06%	2.40%	1.74%	0.53%	4.57%
%	100.00%	66.67%	100.00%	100.00%	66.67%	100.00%

Given these results:

- Only for two time series of daily stock prices, the ones related to Banca Intesa and to Generali Assicurazioni, the TA-based FTS over-performs some of the configurations of the RL-based FTSs;
- Considering all the results as a whole, the RL-based FTSs over-perform the TA-based FTS in the 88.89% of the cases.

All this seems to show the goodness of our approaches, at least with respect to the chosen TA-based benchmark.

5 Some Concluding Remarks

In this paper we have developed and applied some original automated FTSs based on differently configured RL-based algorithms. Here we have presented the results coming out from the current phase of our research on this topic. Of course, many questions have again to be explored. In particular:

- The choice of percentage returns as states is a naive choice. Now we are beginning to work to specify some new indicators to use as states (in the first experimentations they have provided interesting results);
- As known, the Sharpe ratio as performance measure suffers several financial limits. Currently, as reward function we are considering alternative and more realistic performance measures;
- The management of the learning rate, α, we have used here is appropriate for stationary systems. But generally financial markets are non-stationary. Because of that, we are beginning to work to develop methods for the dynamic management of the learning rate in non-stationary contexts;
- In order to deepen the valuation about the capabilities of our FTSs, we wish to apply them to more and more financial price series coming from different markets;
- Finally, when all the previous questions will be explored, transaction costs and other frictions will be considered.

Acknowledgements. The authors wish to thank the Department of Economics of the Ca' Foscari University of Venice for the support received within the research project *Machine Learning adattativo per la gestione dinamica di portafogli finanziari* [*Adaptative Machine Learning for the dynamic management of financial portfolios*].

References

1. Barto, A.G., Sutton, R.S.: Reinforcement Learning: An Introduction. Adaptive Computation and Machine Learning. The MIT Press (1998)
2. Bekiros, S.D.: Heterogeneous trading strategies with adaptive fuzzy Actor-Critic reinforcement learning: A behavioral approach. Journal of Economic Dynamics & Control 34, 1153–1170 (2010)
3. Bertoluzzo, F., Corazza, M.: Making financial trading by recurrent reinforcement learning. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007/WIRN 2007, , Part II. LNCS (LNAI), vol. 4693, pp. 619–626. Springer, Heidelberg (2007)
4. Bertsekas, D.P., Tsitsiklis, J.N.: Neuro-Dynamic Programming. Athena Scientific (1996)
5. Brent, R.P.: Algorithms for Minimization without Derivatives. Prentice-Hall (1973)
6. Bosq, D.: Nonparametric Statistics for Stochastic Processes. Estimation and Prediction, vol. 110. Springer (1996)
7. Cuthbertson, K., Nitzsche, D.: Quantitative Financial Economics. Wiley (2004)
8. Farmer, D., Lo, A.W.: Market force, ecology and evolution. Industrial and Corporate Change 11, 895–953 (2002)
9. Gold, C.: FX trading via recurrent Reinforcement Learning. In: Proceedings of the IEEE International Conference on Computational Intelligence in Financial Engineering, pp. 363–370 (2003)
10. Li, H., Dagli, C.H., Enke, D.: Short-term stock market timing prediction under reinforcement learning schemes. In: Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning, pp. 233–240 (2007)
11. Lo, A.W., Mamaysky, H., Wang, J.: Foundations of technical analysis: Computational algorithms, statistical inference, and empirical (2000)
12. Lo, A.W.: The Adaptive Markets Hypothesis. Market efficiency from an evolutionary perspective. The Journal of Portfolio Management 30, 15–29 (2004)
13. Moody, J., Wu, L., Liao, Y., Saffel, M.: Performance functions and Reinforcement Learning for trading systems and portfolios. Journal of Forecasting 17, 441–470 (1998)
14. Moody, J., Saffel, M.: Learning to trade via Direct Reinforcement. IEEE Transactions on Neural Network 12, 875–889 (2001)
15. Murphy, J.J.: Technical Analysis of the Financial Markets. A Comprehensive Guide to Trading Methods and Applications. New York Institute of Finance (1999)
16. Jangmin, O., Lee, J., Lee, J.W., Zhang, B.-T.: Adaptive stock trading with dynamic asset allocation using reinforcemnt learning. Information Sciences 176, 2121–2147 (2006)
17. Ormonet, D.: Kernel-Based Reinforcement Learning. Machine Learning 49, 161–178 (2002)
18. Smart, W.D., Kaelbling, L.P.: Practical Reinforcement Learning in continuous spaces. In: Proceedings of the 17th International Conference on Machine Learning, pp. 903–910 (2000)