Estimating the out-of-sample predictive ability of trading rules: a robust bootstrap approach

J. Hambuckers, C. Heuchenne

Abstract

In this paper, we provide a novel way to estimate the out-of-sample predictive ability of a trading rule. Usually, this ability is estimated using a sample splitting scheme, true out-of-sample data being rarely available. We argue that this method makes a poor use of the available data and creates data mining possibilities. Instead, we introduce an alternative .632 bootstrap approach. This method enables to build in-sample and out-of-sample bootstrap datasets that do not overlap but exhibit the same time dependencies. We show in a simulation study that this technique drastically reduces the mean squared error of the estimated predictive ability. We illustrate our methodology on IBM, MSFT and DJIA stock prices, where we compare 11 trading rules specifications. For the considered datasets, two different filter rule specifications have the highest out-of-sample mean excess returns. However, all tested rules cannot beat a simple buy-and-hold strategy when trading at a daily frequency.

JEL: C52, C53, C15, C18

Keywords: trading rules, bootstrap, bootstrap .632, out-of-sample, predictive ability, parameter uncertainty.

1 Introduction

Whether technical trading rules can consistently generate profits is a question that has been investigated by researchers, for a variety of reasons. In particular, it can be used to study if a market is efficient or not. Indeed, profitable technical trading rules would be in opposition with the market efficiency hypothesis, which states that all available information must be contained in the price of a security [Bajgrowicz and Scaillet, 2012]. Besides, the profitability of trading rules may also be used to detect a time-varying risk premium [Kho, 1996]. Moreover, as a high proportion of practitioners rely on technical trading rules to trade, it may be interesting to know if these investment strategies have any economic value [see Park and Irwin, 2007, for a review].

However, answering to this question poses several econometric challenges. Until now, the literature focused mainly on the multiple testing (also called data mining) issues (see Lo and MacKinlay [1990], White [2000], Sullivan et al. [1999], Hansen [2005], Romano and Wolf [2005], Hsu et al. [2010] for more considerations on this question), whereas it neglected the issue of computing an adequate estimator of the out-of-sample predictive ability of a trading rule. By out-of-sample predictive ability, we mean the ability of a rule, with its parameters determined *ex ante* on in-sample data, to generate buy-and-sell signals that correctly predict the future ups and downs of an asset price. Recently, Fang et al. [2014] stress out the need for fresh out-of-sample data, as it is considered to offer the strongest safeguard against possible statistical bias. Kuang et al. [2014] and Sullivan et al. [1999] point out that an out-of-sample analysis is often an effective way of detecting a data snooping bias. In practice, however, fresh data are not always available and a true out-of-sample analysis is rarely possible. Therefore, researchers use various sample splitting approaches. Kuang et al. [2014] and Bajgrowicz and Scaillet [2012] study the out-of-sample profits obtained with strategies based on *ex ante* selected trading rules parametrizations. In Kuang et al. [2014], they select all rules apparently profitable on a subsample and then they compute the out-of-sample performance on a second subsample. In Bajgrowicz and Scaillet [2012], the authors build monthly portfolios of rules using data of a single month and compute the out-of-sample performance over the following month. Bajgrowicz and Scaillet [2012] call this task *persistence analysis*. Both studies conclude that despite a superior in-sample performance of some rules (without taking the transaction costs into account), these results cannot be reproduced on out-of-sample data. According to Park and Irwin [2007], this is also the methodology followed by Lukac et al. [1988]. Allen and Karjalainen [1999] compute the out-of-sample performance of trading rules obtained via genetic algorithms in a similar way (four years of data to build the rules, two years to select the best ones and the remaining data to assess the performance).

It exists several drawbacks to this splitting technique. First, the results are very dependent from the cut-off point and are very volatile, because the estimation is based on a single realization of a stochastic process. Also, it is a suboptimal use of the available data: by setting aside some data for the purpose of validation, we do not use all the information at hand to select *ex ante* the best parametrizations of the rules (as an investor would likely do in the real life), and therefore we usually decrease the quality of this selection. Notice also that some authors [Bajgrowicz and Scaillet, 2012, Kuang et al., 2014] do not assume parameter uncertainty and consider all parameters as *a priori* fixed: they assume that each parametrization is a rule in itself, and their selection of the best rules (or of the profitable ones) is made across all rules specifications. Oddly, to the best of our knowledge and despite the apparent weaknesses of this technique, no study in the field of technical trading rules tries to focus on these issues and to solve them. This is surprising, as one could expect evolved investors to search for trading rules that forecast correctly the future, not the past.

In this work, we introduce a methodology that avoids these weaknesses. We propose a way to improve the basic sample splitting technique by adapting advanced cross-validation and bootstrap techniques to the time series context. Our goal here is to obtain a better measure of the out-of-sample predictive ability of a trading rule. This idea is briefly suggested in White [2000], when the author tells us that "(...) cross-validation represents a more sophisticated use of hold-out data. It is plausible that our methods may support testing that the best cross-validated model is no better than a benchmark". Cross-validation techniques are quite common in the field of neural networks modeling and classification problems but did not attract a lot of attention from researchers in the area of technical trading. Moreover, few adaptations to the time series context currently exist.

Among the best known techniques in the regression context, a first method is to split the sample not into two parts, but into k parts [Efron and Tibshirani, 1993]. Hence, we estimate the parameters on $k-1$ parts and compute the out-of-sample prediction error (or the value of the score function) on the last part. This operation is performed for the k different parts, before averaging to obtain our final estimator of the prediction error. This technique is called k-fold cross-validation. The roll-over month-by-month approach followed by Bajgrowicz and Scaillet [2012] and Taylor [2014] can be linked to this idea. Now, if k is equal to the size of the sample, we perform a leave-one-out cross-validation. A second method relies on the bootstrap [Efron and Tibshirani, 1993]. With a resampling procedure, we build a statistical world replicating the properties of the true world, where we are able to estimate B sets of parameters. Then, the initial sample is used as a validation (out-of-sample) set. It is as if we had at hand multiple realizations of the same stochastic process for estimation purpose and the opportunity to validate these results on the whole population. The disadvantage of this method is the big overlapping between training sets and validation sets, that causes a bias. To solve this issue, the bootstrap .632 and +.632 techniques [Efron, 1983, Efron and Tibshirani, 1993, 1997] can be used to compute estimators based on validation sets that do not overlap with the training sets. Overall, these techniques must be seen as generalizations of the sample splitting technique.

In Section 2, we present an adaptation to the time series context of the .632 bootstrap technique. Our procedure is based on the idea that, to correctly assess the predictive ability of a trading rule, we need a large number of the possible outcomes of the underlying stochastic process. Using the .632 resampling technique, we are able to build a large number of both training sets (in-sample data) and validation sets (out-of-sample data). Also, we can simultaneously control for non-overlapping datasets and keep intact the intrinsic time dependencies of the original data. In the bootstrap world, we fit the rules on the training sets and then use the bootstrap validation sets to compute an estimator of the out-of-sample predictive ability of the rules. The bootstrap training samples are generated like in a regular bootstrap procedure for time series data: by block, or using a nonparametric

resampling of the residuals, that are used to build recursively new data with the estimated model. The bootstrap validation samples, for their part, are drawn using the residuals (or blocks of data) *not used* in the training samples. Indeed, an elementary calculation tells us that, if we draw a sample of size n with replacement from an initial set of n observations, a single observation has roughly a probability $1 - \left(\frac{n-1}{n}\right)^n \simeq 0.368$ to not be selected in the resample. It means that, on average, around one third of the data would stay unused in each resample. We use these unselected observations to create bootstrap validation samples that do not overlap with the training samples. These validation samples can be used to assess the out-of-sample performance, avoiding an overfitting bias. When a good stationary time series model can be found, the residual-based bootstrap is preferred to the block bootstrap of Politis and Romano [1994]. Indeed, the residuals-based bootstrap has been shown to produce very good results when the hypotheses of the underlying model are met. Also, it is interesting to notice that our approach could easily be extended to all data frequencies (especially very high frequencies). However, our approach differs slightly from the traditional one. Here, we estimate the out-of-sample predictive ability of a trading rule with its best parametrization determined *ex ante* on in-sample data. In Brock et al. [1992], Allen and Karjalainen [1999], Bajgrowicz and Scaillet [2012] and Kuang et al. [2014], the authors are interested in computing the out-of-sample performance of the combinations parameters - trading rules that perform best in-sample. In other words, our perspective takes into account the parameter uncertainty around a trading rule, whereas the traditional approach assumes no parameter uncertainty. As an example, imagine that we consider two rules (let's say cross-over moving average and support and resistance rules), each with 10 possible parametrizations. We have 20 combinations parameters - trading rules. Whereas Brock et al. [1992], Sullivan et al. [1999] and Bajgrowicz and Scaillet [2012] try to find the combination(s) that generates the highest mean excess return (or the ones that generates a profit), we aim at finding, among the two rules, the one that can generate the highest out-of-sample mean excess return (see Section 2.1 for more comments regarding this perspective).

To sum up, our procedure allows:

- to select the optimal parameters of the rules on in-sample data (i.e. a training set), regarding a given scoring rule (e.g. the mean excess return of the positions in the asset, over time),
- to compute, for this rule with its optimal parameters, the value of the scoring rule on new out-of-sample data (i.e. a validation set built during the resampling procedure),
- and to average over all the bootstrap samples to get our final estimator of the predictive ability of the considered rules.

With this estimator, we can compare trading rules between each other and identify the one that has the highest predictive ability. We can also identify those that have a predictive

ability above a pre-specified threshold.

With this method, we provide an alternative to the sample splitting technique (with or without rolling window), traditionally used by researchers to estimate the out-of-sample performance of trading rules [see, e.g. Allen and Karjalainen, 1999, Lukac et al., 1988, Bajgrowicz and Scaillet, 2012, Kuang et al., 2014]. Our method has the advantage to not be dependent from some subjective cut-off points. Because we use a resampling procedure, we offer a convenient way to generalize the splitting of the dataset into multiple subsamples. Moreover, by working with a random resampling procedure instead of a subjective sample splitting procedure, we can incorporate, in our measure of the out-of-sample performance, new scenarios (i.e. new patterns of prices). These scenarios are useful to assess a trading rule on artificial but totally new data. Especially, using a residuals-based resampling, we are able to create new patterns of (stochastic) shocks and to keep simultaneously the time dependencies intact. In this configuration, we can combine a large number of different pasts (our bootstrap training sets) with a large set of different futures (the bootstrap validation sets), although these sequences of events do not exist in our original time series. Eventually, we also provide a way to create bootstrap in- and out-of-sample time series of interest rate data *via* the dynamic Nelson-Siegel model of Diebold and Li [2006]. This procedure can be helpful when we use the mean excess return criterion as a measure of the rules' performances.

In Section 3, we illustrate our methodology on three time series: the daily stock prices of IBM, Microsoft (MSFT) and the Dow Jones Industrial Average (DJIA), for the period 09/20/2002 - 05/16/2014. We compare 11 trading rule specifications, used by Sullivan et al. [1999] and Bajgrowicz and Scaillet [2012]. These rules rely on daily past prices. We do not consider rules based on high frequency prices or volumes. We compare the results of the bootstrap estimations to the ones obtained with a classical in-sample prediction error and a BH strategy. To resample the data, we use simple ARMA-GARCH models, as our data appear to exhibit heteroscedasticity and stationary mean processes. We show that despite a good in-sample performance (measured with a mean excess return criterion and a Sharpe ratio criterion), these rules do not have an out-of-sample performance that beats the BH strategy. Our results are in line with studies [Bajgrowicz and Scaillet, 2012, Sullivan et al., 1999, Brock et al., 1992, Kuang et al., 2014, Fang et al., 2014] that detect an in-sample superior performance of some trading rules but cannot reproduce this result on out-of-sample data. In particular, our results for the DJIA data are similar to the ones of previous studies focusing on this stock index. These results appear to be robust to the resampling method (either block or residuals-based) and to the different measures of the performance (either a mean excess return criterion or a Sharpe ratio criterion).

Finally, to investigate the validity of our results, we perform a simulation study (Section 4). We show that under the correct specification of the parametric model used in the

resampling procedure, our bootstrap estimator of the predictive ability has a mean squared error (MSE) that is up to 93.9% lower than the one obtained with a sample splitting procedure. Besides, the bootstrap distribution of the out-of-sample prediction error is very close to its true distribution (obtained by Monte Carlo simulation), suggesting that our approach is effective.

We conclude and discuss extensively these results in Section 5.

2 Methodology

2.1 Resampling procedure

In this subsection, we detail the resampling procedure used to compute our estimator of the out-of-sample predictive ability. The goal here is to build a statistical world where the time dependencies are similar to the ones in the true world. We want the asymptotic distribution of the considered statistic in the bootstrap world to be close to its distribution in the true world [Kreiss and Paparoditis, 2011]. Remind also that we want to build training sets and validation sets that do not overlap, to control for a possible overfitting bias.

While the initial bootstrap in Efron [1979] was developed for inference with i.i.d. data, many extensions have been provided in the context of stationary time series. Besides stationary block bootstrap methods [Kunsch, 1989, Politis and Romano, 1994], parametric residuals-based bootstrap methods are known to work well for stationary dependent data, when the assumptions of the underlying model are met [Kreiss and Paparoditis, 2011]. The general idea is to resample the residuals of the model, instead of the original data. Due to its simplicity, the absence of "tuning parameters" (in opposition to the average length of the blocks in the block bootstrap approach) and the availability of a lot of good models for stock returns, we focus on this kind of resampling procedures.

First, we apply some transformations to the initial time series, to obtain data that can be modeled by stationary time series models (for daily stock prices, taking the log difference is often enough). Assume that the data generating process (DGP) of these data (i.e. the stock log returns) has a representation of the form

$$R_t = \mu_{\theta_\mu}(R_{t-1}, \cdots, R_{t-p}) + \sigma_{\theta_\sigma}(R_{t-1}, \cdots, R_{t-q}) \cdot z_t,\ t \in \mathcal{Z}, \tag{1}$$

where μ_{θ_μ} is the conditional mean function with parameters θ_μ, σ_{θ_σ} the conditional variance function with parameters θ_σ and z_t are i.i.d. innovations with zero-mean and unit variance [see Politis and Romano, 1994, Kreiss and Paparoditis, 2011]. For the rest of the paper, we make the hypothesis that the parametric forms of μ_{θ_μ} and σ_{θ_σ} are given by an ARMA(p,q)-

GARCH(p',q') model:

$$
\begin{aligned}
R_t &= \mu_t + r_t, && (2)\\
r_t &= \sigma_t z_t, && (3)\\
\mu_t &= \omega^{(m)} + \sum_{i=1}^{p} \alpha_i^{(m)} R_{t-i} + \sum_{j=1}^{q} \beta_j^{(m)} r_{t-j}, && (4)\\
\sigma_t^2 &= \omega + \sum_{i=1}^{p'} \alpha_i r_{t-i}^2 + \sum_{j=1}^{q'} \beta_j \sigma_{t-j}^2, && (5)
\end{aligned}
$$

with $z_t \overset{iid}{\sim} f(\cdot)$, $E(z_t) = 0$ and $E(z_t^2) = 1$. We do not assume any particular distribution on $f(.)$, simply the existence of at least the four first moments, for inference purposes (in the bootstrap procedure, we perform a nonparametric resampling of the residuals). Using a nonparametric resampling of the residuals has the advantage to avoid wrong assumptions on $f(.)$, the choice of an error distribution being a delicate step in a modeling process [see, e.g. Engle and Gonzalez-Rivera, 1991, for a discussion on the subject]. If needed, we can use more complex parametric structures (EGARCH, GJR-GARCH, M-GARCH,...) to capture the dependencies of our data, without loss of generality. We follow the procedure of Hall and Yao [2003] to obtain i.i.d. residuals that may be resampled. For an initial sample $R = \{R_1, \cdots, R_n\}$ of size n:

1. obtain $\hat{\theta}_\mu = \{\hat{\omega}^{(m)}, \hat{\alpha}_1^{(m)}, \cdots, \hat{\alpha}_p^{(m)}, \hat{\beta}_1^{(m)}, \cdots, \hat{\beta}_q^{(m)}\}$ and $\hat{\theta}_\sigma = \{\hat{\omega}, \hat{\alpha}_1, \cdots, \hat{\alpha}_{p'}, \hat{\beta}_1, \cdots, \hat{\beta}_{q'}\}$, consistent estimates of $\theta_\mu = \{\omega^{(m)}, \alpha_1^{(m)}, \cdots, \alpha_p^{(m)}, \beta_1^{(m)}, \cdots, \beta_q^{(m)}\}$ and $\theta_\sigma = \{\omega, \alpha_1, \cdots, \hat{\alpha}_{p'}, \beta_1, \cdots, \hat{\beta}_{q'}\}$ respectively (e.g. using quasi-maximum likelihood techniques).

2. Compute $\hat{\mu}_t$ and $\hat{\sigma}_t$ for $t = 1, \cdots, n$, estimations of the conditional means and variances, based on the estimated parameters.

3. Compute $\hat{z}_t = (R_t - \hat{\mu}_t)/\hat{\sigma}_t$ for $t = 1, \cdots, n$, estimations of the true innovations z_t, and centred them.

4. Generate randomly $B \times n$ bootstrap innovations $z_{b,t}^*$, for $t = 1, \cdots, n$ and $b = 1, \cdots, B$, from the historical distribution of the estimated centred innovations $\{\hat{z}_1, \cdots, \hat{z}_n\}$, with replacement. Save the time indices of the selected innovations, $T(b, t)$ (these are used later).

5. Using equations 2 to 5, the estimated sets of parameters $\hat{\theta}_\mu$ and $\hat{\theta}_\sigma$, and the resampled innovations, build recursively B bootstrap time series $R_b^* = \{R_{b,1}^*, \cdots, R_{b,n}^*\}$ of size n using the following equation:

$$R_{b,t}^* = \mu_{b,t}^* + \sigma_{b,t}^* z_{b,t}^*, \qquad (6)$$

with

$$r^*_{b,t} = \sigma^*_{b,t} z^*_{b,t}, \tag{7}$$

$$\mu^*_{b,t} = \hat{\omega}^{(m)} + \sum_{i=1}^{p} \hat{\alpha}_i^{(m)} R^*_{b,t-i} + \sum_{j=1}^{q} \hat{\beta}_j^{(m)} r^*_{b,t-j}, \tag{8}$$

$$(\sigma^*_{b,t})^2 = \hat{\omega} + \sum_{i=1}^{p'} \hat{\alpha}_i r^{*,2}_{b,t-i} + \sum_{j=1}^{q'} \hat{\beta}_j (\sigma^*_{b,t-j})^2. \tag{9}$$

To initialize the recursion, we use $\hat{\omega}^{(m)}/(1 - \sum_{i=1}^{p} \hat{\alpha}_i^{(m)} - \sum_{j=1}^{q} \hat{\beta}_j^{(m)})$ for the starting conditional mean, $\hat{\omega}/(1 - \sum_{i=1}^{p'} \hat{\alpha}_i - \sum_{j=1}^{q} \hat{\beta}_j)$ for the starting conditional volatility, and observed values of R_t [Davidson and MacKinnon, 2006].

Each bootstrap sample is used to estimate the optimal parameters that minimize the in-sample prediction error, for a given trading rule. In the context of trading rules, the predictive ability can be measured throughout a prediction error like a misclassification rate or a score function. If the score function has a positive economic signification (e.g. the mean excess return), then we need to maximize it. Each rule $l = 1, \cdots, L$ is used to make $n - h$ one-step ahead predictions (h being the larger number of past values needed for a single forecast), using the parameters θ_l. A prediction error is associated to each forecast throughout a function $Q(\cdot)$. The value of θ_l (as suggested before, e.g., the window sizes of the moving-averages) that generates the lowest mean prediction error is adopted as an estimator $\bar{\theta}^*_{b,l}$ of the optimal parameters, for the b^{th} bootstrap sample and the l^{th} rule. Analytically, for $b = 1, \cdots, B$ and $l = 1, \cdots, L$, we compute

$$\bar{\theta}^*_{b,l} = arg \min_{\theta \in \Theta} \frac{1}{n-h} \sum_{t=h+1}^{n} Q(R^*_{b,t}, \theta). \tag{10}$$

The selection of $\bar{\theta}^*_{b,l}$ is performed over a finite set Θ. For a given rule l, by averaging $\frac{1}{n-h} \sum_{t=h+1}^{n} Q(R^*_{b,t}, \bar{\theta}^*_{b,l})$ over the B resamples, we would obtain an estimator of the expected in-sample prediction error, but we are interested in the true predictive ability of a rule, i.e. in the performance of a rule l with parameters $\bar{\theta}^*_{b,l}$, measured on a totally new dataset, ideally very large. How do we obtain new values of $R^*_{b,t}$, without an excessive overlapping with the training data? This is where the bootstrap procedure starts to make sense. For a given resample, the bootstrap innovations are drawn from $\{\hat{z}_1, \cdots, \hat{z}_n\}$ with replacement. A quick calculation shows that, for n fairly large, the probability of one single innovation (let's say innovation k) to not be selected in the resample is $\left(\frac{n-1}{n}\right)^n \simeq .368$. It means

that more than a third (on average) of the innovations are "wasted", for a single resample. Following the idea of the bootstrap .632 described in Efron [1983] and Efron and Tibshirani [1997], we propose to take advantage from the fact that we know exactly how the resamples are built. For each resample $b = 1, \cdots, B$, we use the V_b unselected innovations to create a new validation (out-of-sample) dataset, that does not overlap with the training set. More formally, we add the following steps to the initial resampling procedure:

6. for $b = 1, \cdots, B$, use the indices saved in step 4 to identify all V_b unselected innovations. Arrange them in increasing order of time index and form a new set of innovations (here re-indexed for clarity) $\{\epsilon^*_{b,1}, \cdots, \epsilon^*_{b,V_b}\}$.

7. For $b = 1, \cdots, B$, use $\hat{\theta}_\sigma$, $\hat{\theta}_\mu$ and the last values of the associated training resample as starting values to build the b^{th} out-of-sample datasets $\mathcal{R}^*_b = \{\mathcal{R}^*_{b,1}, \cdots, \mathcal{R}^*_{b,V_b}\}$. $\mathcal{R}^*_{b,t}$ are obtained using equations 2 to 5 with $\mathcal{R}^*_{b,t}$ replacing $R^*_{b,t}$ and $\epsilon^*_{b,t}$ replacing $z^*_{b,t}$.

With these out-of-sample data at hand, we take the average of the sum of the scores over the B resamples to obtain a measure of the predictive ability of rule l:

$$S^*_l = \frac{1}{B} \sum_{b=1}^{B} \sum_{t=1}^{V_b} \frac{Q(\mathcal{R}^*_{b,t}, \bar{\theta}^*_{b,l})}{V_b}, \tag{11}$$

However, due to the independence between training and validation datasets, this estimator tends to be too large (for Q having a negative meaning, like a prediction error). To correct for this effect, Efron and Tibshirani [1993] and Efron and Tibshirani [1997] propose two ways to combine this quantity with the in-sample prediction error S^0_l obtained on the initial sample. Let's define $\bar{S}_l$, a new estimator of the out-of-sample prediction error of the l^{th} rule:

$$\bar{S}_l = .368 S^0_l + .632 S^*_l, \tag{12}$$

with

$$S^0_l = \frac{1}{n-h} \sum_{t=h+1}^{n} Q(R_t, \bar{\theta}_l), \tag{13}$$

$$\bar{\theta}_l = \min_{\theta \in \Theta} \frac{1}{n-h} \sum_{t=h+1}^{n} Q(R_t, \theta), \tag{14}$$

where $\bar{\theta}_l$ is the estimated optimal set of parameters for rule l, for the initial sample. The origin of the weights in $\bar{S}_l$ are discussed at the end of this section. Nevertheless, when the in-sample overfitting is high, these weights may not be appropriate anymore. Therefore, Efron and Tibshirani [1997] proposed also a second weighting between S^0_l and S^*_l, function of the in-sample overfitting. Let's define $\bar{S}^+_l$, also an estimator of the out-of-sample prediction error of the l^{th} rule:

$$\bar{S}^+_l = (1 - w) S^0_l + w \cdot S^*_l, \tag{15}$$

with

$$w = \frac{.632}{1 - .368J}, \tag{16}$$

and

$$J = \frac{S_l^* - S_l^0}{\Lambda - S_l^0}. \tag{17}$$

J is defined as the *relative overfitting rate* and is constrained to take values between 0 and 1. w takes values between 0.632 and 1 (hence, $\bar{S}_l^+$ is bounded by S_l^* and $\bar{S}_l$). Λ is defined as the *no information error rate*, i.e. the expected error rate of the rule when the predictors are independent from the predicted variables. Intuitively, Λ estimates the expected prediction error of the rule in a sort of worst case scenario, that is when there are no relationships between the explicative and the predicted variables. In fact, it is similar to the performance of a rule that predicts **randomly** the ups and downs of the considered time series. Hence, $\Lambda - S_l^0$ estimates the difference between what could be the performance of a single sequence of random predictions (S_l^0), particularly lucky, and the expectation of such a random prediction process (Λ). The closer $S_l^* - S_l^0$ (the discrepancy between the apparent error and the out-of-sample error) is to $\Lambda - S_l^0$, the more S_l^0 is influenced by a high overfitting and should not be used in $\bar{S}_l$. See Appendix D for more details regarding the computation of Λ. Our empirical study in the next section indicates that Λ is close to 1. Also, our simulations emphasize that $\bar{S}_l^*$ seems to be closer to the true predictive ability than $\bar{S}_l$. Therefore we decide to use preferably S_l^* or $\bar{S}_l^+$.

2.2 Some words regarding the notion of trading rule

Finally, some words regarding the notion of trading rule. In Brock et al. [1992], Sullivan et al. [1999] and Bajgrowicz and Scaillet [2012], the parameters of the considered trading rules are fixed *a priori*: there are no estimated parameters. When Sullivan et al. [1999] explained that they tested a universe of 7,846 trading rules, this universe stems, in fact, from 5 different **rule specifications** (see Appendix A of their study): filter rules, moving averages, support and resistance, channel breakouts and on-balance volume averages. Falbo and Pelizzari [2011] speak about classes of technical trading rules, or trading styles, instead of rule specifications. In our approach, "trading rule" must be seen as "a mathematical relationship with some parameters, producing buy and sell signals", i.e. a rule specifications in the sense of Sullivan et al. [1999]. By naming "trading rule" a particular parametrization of a rule specification, Sullivan et al. [1999] adopt a narrow notion of a rule. In our opinion, it seems more interesting to quantify the predictive ability of a rule specification with its best *ex ante* parametrization, because what really matter is to find the mathematical relationship that correctly predicts prices variations. By using a bootstrap procedure, we take into account the parameter uncertainty, considered as non-existent in other approaches. In fact, it is because we assume this parameter uncertainty that the

.632 resampling is useful. Indeed, the initial bootstrap training samples are used to get estimators of the best parameters, whereas the validation sets are used to estimate the out-of-sample performance. If the parameters were treated as fixed, then a classical bootstrap approach would have been sufficient to get estimators of the out-of-sample performance, for each parametrization.Also, we are well aware that most trading rules use prices (P_t) and not returns as inputs, to generate their buy and sell signals. Until now, we have only discussed the resampling of log returns, but the recursive computation of the prices is straightforward. Using an initial price P_0, we obtain series of prices $P = \{P_1, ..., P_n\}$ with the equation

$$P_t = P_{t-1}\, \mathrm{e}^{R_t}, \tag{18}$$

that is obtained from the definition of the log returns $R_t = \log(P_t/P_{t-1})$.

2.3 Alternative to the residuals-based bootstrap: the block bootstrap

As mentioned before, when no parametric model seems suitable to model the data, one can simply use the classical stationary block bootstrap approach of Romano and Wolf [2005], for strictly stationary and weakly dependent time series. Under these assumptions, for a broad class of nonlinear statistics, this technique provides consistent estimators of the considered statistics [see Kreiss and Paparoditis, 2011, for more details]. Starting from the initial series of log returns R, we build block bootstrap training samples by executing the following steps [Sullivan et al., 1999, Bajgrowicz and Scaillet, 2012]:

For $b = 1, \cdots, B$

1. Define a such that $1/a$ is equal to the average length of a block.
2. Set $t = 1$ and draw randomly $R^*_{b,t}$ from R. Let $T(b,t)$ be the time index of the t^{th} selected observation (in the bootstrap sample b), among R. Save it.
3. Set $t = t + 1$. If $t > n$, stop. Otherwise, draw a random variable U from the uniform distribution.
   (a) If $U < a$, start a new block by drawing randomly $R^*_{b,t}$ from R. Save $T(b,t)$.
   (b) If $U \geq a$, set $T(b,t) = T(b,t-1) + 1$ and $R^*_{b,t} = R_{T(b,t)}$. If $T(b,t) > n$, set $T(b,t) = 1$.
4. Repeat step 3.

Similarly to the procedure explained in Section 2, for each bootstrap sample $b = 1, \cdots, B$, we use the V_b unselected observations to build validation sets. Because the drawing is made

by blocks, the unselected observations are also a set of blocks of average length $1/a$, but with their original order of appearance conserved. Here, we select $a = 0.1$ as in Sullivan et al. [1999] and Bajgrowicz and Scaillet [2012], after a trial-and-error phase. This stage could be improved by using an automatic procedure of selection, as in Politis and White [2004], but this question is beyond the present research.

A notable difference (even if it should not impact significantly the final conclusions) with the residuals-based resampling is that the probability of being in the training sample is now different. Indeed, by performing a block resampling, this probability increases. Figure 1 shows Monte Carlo estimations of this probability as a function of the block length, for samples of size 2000. This difference seems minimal for small blocks (up to block of average length 10, we observe a difference in probability of around .1%), but for large average block lengths (i.e. above 10), it increases. Here, we use an average block length of 10 but if one wishes to use larger blocks, we propose to use a Monte Carlo estimator of this probability instead of the weights .368 and .632, in equations (12) and (16).

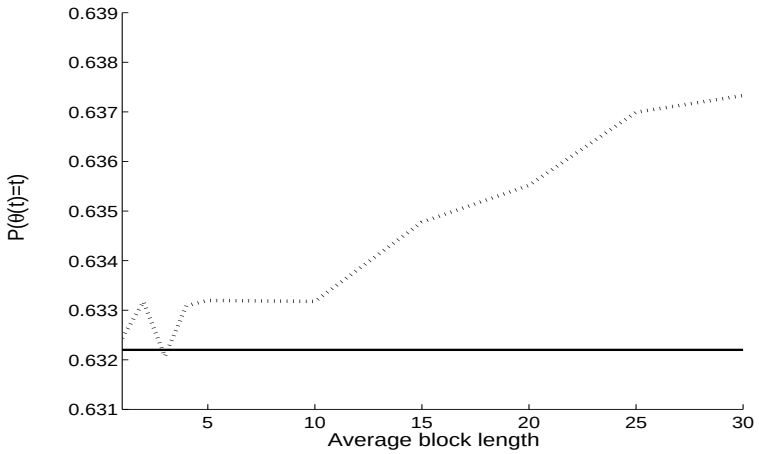

Figure 1: Solid line: Probability for an observation, among a sample of size 2000, to be in a resample of the same size, drawn with replacement. Dashed: Monte Carlo estimation of the probability, for an observation from a sample of size 2000, to be in a resample of same size, drawn by block of average length $1/a$. The size of the Monte Carlo simulations is 200.

3 Empirical illustration

In this section, we illustrate our methodology by applying it on three time series: IBM, MSFT and DJIA stock prices. We use data from the period 09/20/2002 - 05/16/2014. We compare 11 different rule specifications with the goal to find the one that has the highest predictive ability. We also want to know if some rules can generate a profit

higher than a simple BH strategy. We focus on the rule specifications used in Sullivan et al. [1999]. We don't treat the case of on-balance volume averages specifications and focus on price-dependent rules only. To the best of our knowledge, no formal mathematical descriptions of the rules are available in the literature. Therefore, our implementations of the rules could vary a bit compared to other studies on the subject even though we mainly reproduced the implementation of Bajgrowicz and Scaillet [2012], available at http://jfe.rochester.edu/data.htm. Overall, we implemented 11 rule specifications (4 filter rules, 2 moving averages rules, 4 support and resistance rules and one channel breakout rule), with a total of 11,668 parametrizations (4320 for the filter rules, 3780 for the moving average rules, 1008 for the support and resistance rules, and 2560 for the channel breakout rule). A brief description of the considered rules specifications (largely inspired by the ones made in Sullivan et al. [1999] and Bajgrowicz and Scaillet [2012]) can be found in Appendix D. For all selected specifications, we use the same sets of parameters as in Sullivan et al. [1999]. Those parameters are also reproduced in Appendix A. First, we discuss the two different score functions used in our application: a mean excess return criterion and a Sharpe ratio criterion. Then, we present the data and finally we apply our procedure. For the resampling of interest rate data, we use the historical correlation structure between risk-free rates and log returns thanks to a joint resampling procedure. We check if the results obtained with the residuals-based bootstrap are similar to the ones obtained with a block bootstrap procedure.

The interest in working with these three time series are twofold: first, DJIA, IBM and MSFT are popular stocks, frequently traded by private investors and fund managers that may rely on these kind of rules to take investment decisions. It is therefore of interest to know if they can really expect significant profits from such investment techniques or if excess returns are solely due to luck. Second, working with the DJIA enables us to compare our results (in a certain extend) with those obtained previously in popular studies [mainly, Sullivan et al., 1999, Bajgrowicz and Scaillet, 2012].

3.1 Prediction error and optimality criterion

An important question in the analysis of trading rules is to determined what kind of prediction error (or score function) $Q(\cdot)$ is used as an optimality criterion. Sullivan et al. [1999], Bajgrowicz and Scaillet [2012] and Kuang et al. [2014] use the mean excess return over the risk-free rate, as an economic indicator of the performance of a trading rule. More formally, for r_t^f the risk-free rate at time t, s_t taking value 1, 0 or -1 according to the fact that we are long, neutral or short in the asset at time t, $\mathbb{1}(s_t \neq 0)$ taking value 1 if we are not neutral, 0 otherwise, we define

$$Q(R_t, \theta) = \mathbb{1}(s_t \neq 0)(s_t R_t' - r_t^f), \tag{19}$$

where $R_t^{'}$ is the arithmetic return at time t. This definition is used in equations (10) to (14). In this paper, we use the same risk-free rate as in Sullivan et al. [1999] and Bajgrowicz and Scaillet [2012], i.e. the daily federal funds rate (available on the web site of the Federal Reserve Bank of New York, http://www.newyorkfed.org). An issue with this criterion is that, in the bootstrap world, we don't have values of the risk-free rate at our disposal. This said, how can we proceed to compute values of the criteria on the training samples ($Q(R_{b,t}^*, \theta_{b,l}^*)$) and on the validation samples ($Q(\mathcal{R}_{b,t}^*, \bar{\theta}_{b,l}^*)$)? A solution would be to use bootstrap values $r_{b,t}^{f,*}$ of r_t^f. We can draw these bootstrap data exactly in the same way we draw $R_{b,t}^*$ and $\mathcal{R}_{b,t}^*$: either we suppose a parametric model for r_t^f and we perform a residuals-based bootstrap, building recursively time series using resampled residuals, or we perform a block bootstrap resampling as in Section 2.3. If we suppose that r_t^f and R_t are independent processes, then the resampling procedures are made independently. At the contrary, if we assume some common driving factors, we can perform a paired resampling of the residuals. Mathematically, for a residuals-based paired resampling procedure, we build risk-free rate training samples throughout the following steps:

For $b = 1, ..., B$

1. Set $t = 1$.
2. Set $u_{b,t}^* = \hat{u}_{T(b,t)}$, were $\hat{u} = \{\hat{u}_1, \cdots, \hat{u}_n\}$ are estimated residuals of the risk-free rate model and $T(b,t)$ the time index of the t^{th} selected innovation of the prices.
3. Build $r_{b,t}^{f,*}$ recursively using the estimated parameters of the supposed interest rate model. Set $t = t + 1$. If $t > n$, stop. Otherwise go to step 2.

Similar to what is proposed in Section 2.1, risk-free rate validation sets are built using the unselected innovations. Transposition of these steps to a block bootstrap procedure is straightforward. Eventually, in this application, we use the Nelson-Siegel model with autoregressive latent factors of Diebold and Li [2006]. This model is parsimonious and the estimation of its parameters is straightforward. Other models (as the Cox-Ingersoll-Ross model [Cox et al., 1985], or simple ARMA-(E)GARCH models) can be used without any loss of generality. An excellent review on the subject of short term interest rates dynamics, providing comparisons between a large set of models, can be found in Bali and Wu [2006]. Also, Sarno et al. [2005] studied specifically the daily federal fund rate (for another period, though) and concluded that a simple univariate reaction function has the best predictive ability. Here, we do not use their model but stick to their recommendation by choosing a simple model. See Appendix C for details regarding the selected model.

As an optimality criterion, one could also use some kind of Sharpe ratio criterion, where the mean excess return is divided by the standard deviation of the excess returns over the

considered period:

$$\sigma_Q = \sqrt{\frac{1}{n-h}\sum_{t=h+1}^{n}(Q(R_t,\theta))^2 - \left(\frac{1}{n-h}\sum_{t=h+1}^{n}Q(R_t,\theta)\right)^2} \tag{20}$$

This criterion has been used by Bajgrowicz and Scaillet [2012] and Sullivan et al. [1999]. Both criteria have also been used in Kuang et al. [2014]. This criterion is interesting, because rational investors are expected to maximize their ratio risk/return. This criterion can be a good proxy of it.

3.2 Data

We apply our methodology on the three time series (IBM, MSFT and DJIA). Figure 2 shows the evolution of these prices for a period ranging from 09/20/2002 to 05/16/2014, such as their log returns. For IBM log returns, no sample autocorrelation function seems to be significantly different from 0 (taking into account the heteroscedasticity). The mean is assumed to be a constant. However, a typical GARCH effect (volatility clustering, significant sample autocorrelations of the squared returns at multiple lags) is observed. A simple GARCH(1,1) model seems to be a reasonable assumption for this series. For MSFT log returns, we observe a significant sample autocorrelation functions at lag 1 and 2, as well as a GARCH effect. After testing various AR(p), MA(q) and ARMA(p,q) models (with p and q taking values from 0 to 2), it appears that an ARMA(2,2)-GARCH(1,1) specification could be the best model. For DJIA data, we select an ARMA(3,3)-GARCH(2,1) specification. This specification seems to remove all time dependencies. We obtain consistent estimations of θ_σ and θ_μ using quasi-maximum likelihood techniques. The obtained values are displayed in Table 1. Figure 3 shows the estimated volatility and the residuals of the model for all time series. Additional information regarding the model selection can be found in Appendix B.

If we apply the rules on the original time series only, we obtain various combinations of optimal parameters for each rule. Tables 2 to 7 give the optimal parametrizations for each rule, all time series and both criteria (mean excess returns and Sharpe ratios). Notice that searching for good parameters can be difficult, due to the non-linear and non-convex nature of the score function. Figure 4 shows the response surface of the mean return criterion of a simple filter rule, obtained using the IBM data. We observe multiple local extrema. It illustrates that the delimitation of the parameters space strongly influences the results. Therefore, it is important to take into account the estimation procedure of the parameters.

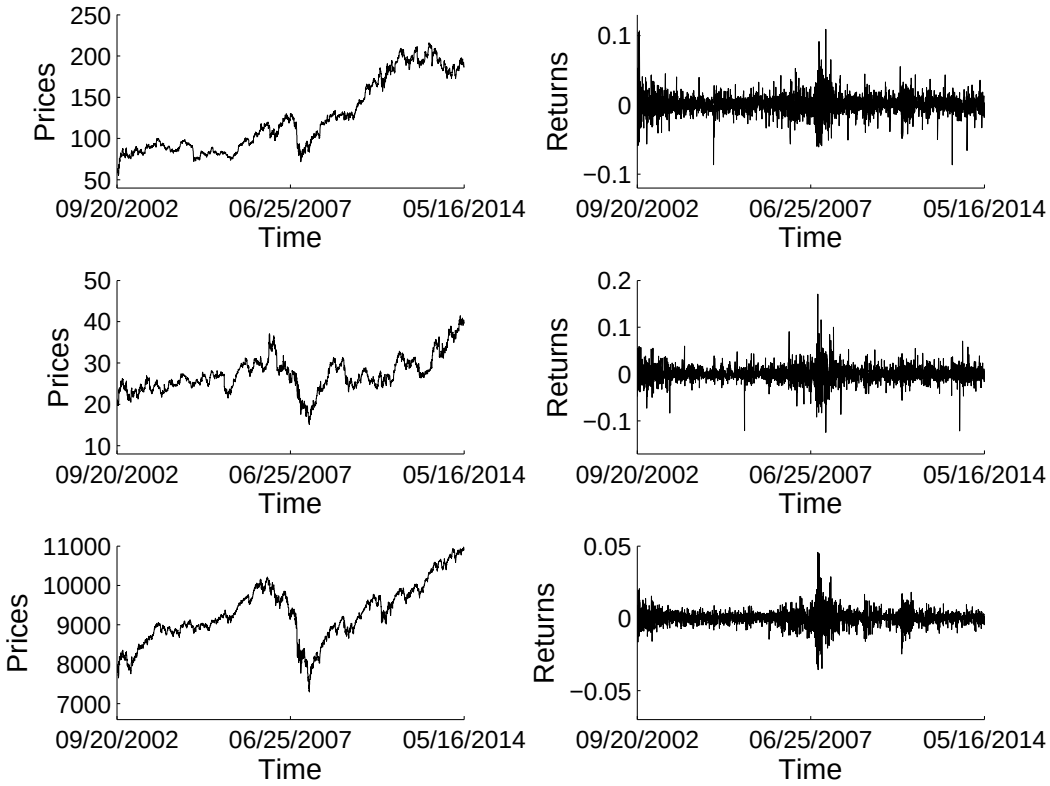

Figure 2: IBM (up), MSFT (middle) and DJIA (down) stock prices (left) and log returns (right) for the period 09/20/2002 - 05/16/2014.

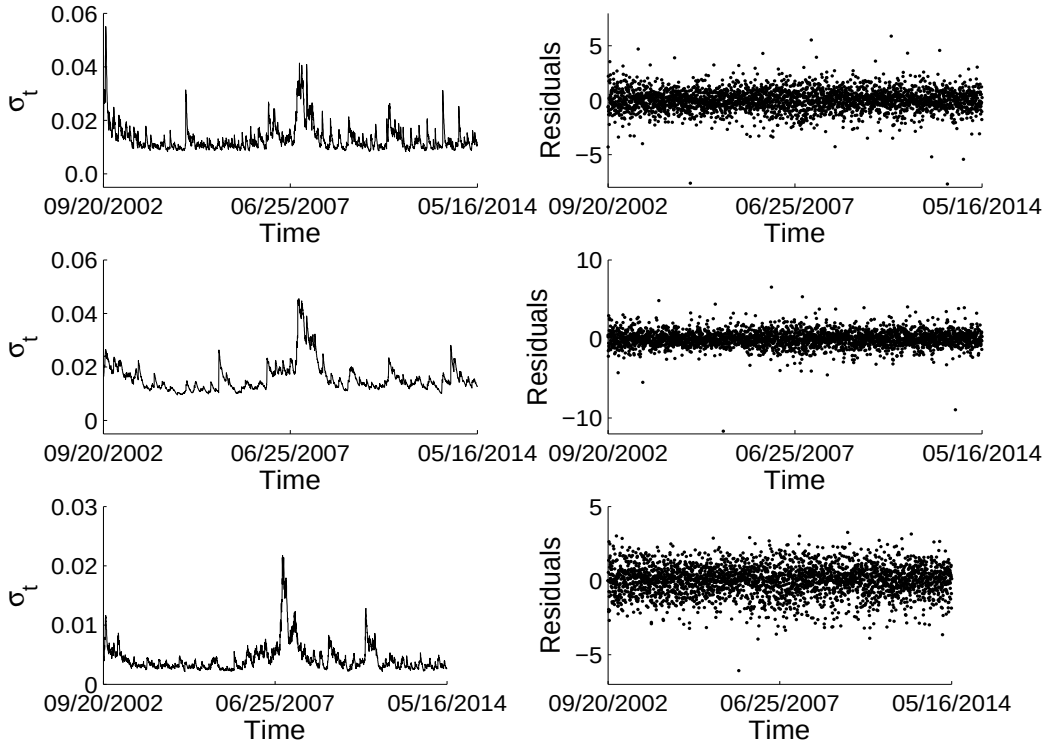

Figure 3: IBM (up), MSFT (middle) and DJIA (down) estimated standard deviation (left) and associated residuals (right) for the period 09/20/2002 - 05/16/2014.

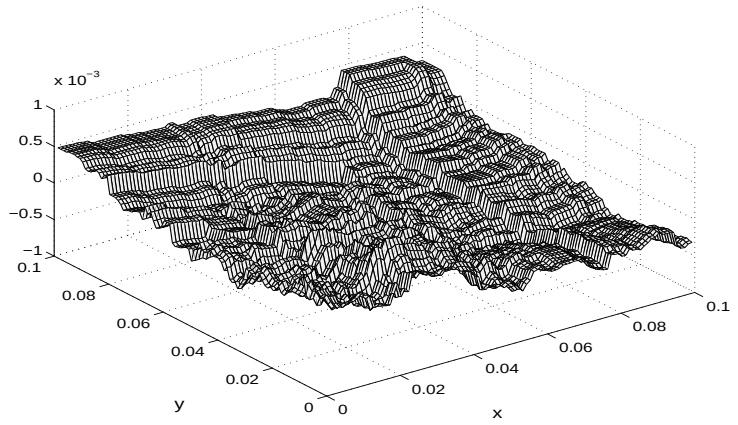

Figure 4: Response surface for the mean excess return of a simple filter rule with x and y parameters ranging from 0 to 0.10, for IBM daily prices. Parameter e is set up to 5.

3.3 Results of the bootstrap procedure

Tables 8 to 10 show the values of the mean excess return and the Sharpe ratio computed with the various formulas. We set $B = 500$. The bootstrap samples are drawn with a residuals-based bootstrap. Results obtained with a stationary block bootstrap procedure can be found in Appendix D, Tables D.2 to D.4. For the residuals-based bootstrap, we use the models selected in the previous section. For the resampling of interest rates data, we use Diebold and Li [2006] approach. See Appendix C for more details regarding the estimation procedure and the values of the parameters for the interest rate time series. The resampling of the innovations for both the prices and the interest rates are paired, to take into account the empirical structure of the correlation. Figure 5 shows some of the generated patterns, in-sample and out-of-sample.

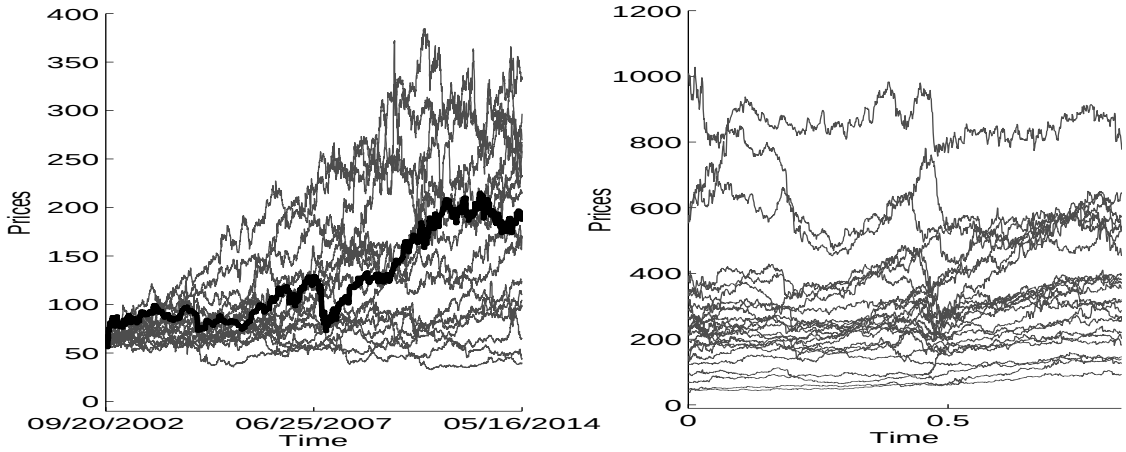

Figure 5: Left: bootstrap training samples of prices, for IBM data (solid grey) and initial time series (solid black). Right: bootstrap validation samples for the same time series.

We observe that the out-of-sample mean excess returns are lower than the in-sample values, for all time series. This is the case for all rules. Also, we see that the ranking between trading rules can vary between in-sample and out-of-sample results: for IBM data, whereas the filter rule with modified extrema has the highest in-sample value of the score function, S_l^* and $\bar{S}_l$ reveal that the highest out-of-sample value of the score function is obtained with the simple filter rule. For MSFT data, the highest in-sample mean excess return is obtained for the filter rule with a constant holding period, whereas the best out-of-sample value is obtained with the simple filter rule. For DJIA data, the best rule in-sample is the modified filter rule, whereas the best rule out-of-sample is the simple filter rule. Thus, being good in-sample does not guarantee to be good out-of-sample. The in-sample mean excess returns are positive for all considered rules except for the Channel Breakout rule

for IBM data. At the contrary, the out-of-sample mean excess returns of respectively 8, 11 and 9 rules are negative for IBM, MSFT and DJIA time series. Using the $\bar{S}_l$ estimator, these numbers become 1, 3 and 2 but the out-of-sample mean excess returns are still way lower than the in-sample results. To obtain the $\bar{S}_l^+$ estimator, we compute the Λ parameters throughout a Monte Carlo simulation (their values and more details are displayed in Appendix D). Due to its construction, $\bar{S}_l^+$ should take values between $\bar{S}_l$ and S_l^*, but the relative overfitting is so high that $\bar{S}_l^+ \simeq S_l^*$ for almost all rules. Thus, it seems more appropriate to use $\bar{S}_l^+$ or S_l^* instead of $\bar{S}_l$, as a measure of the out-of-sample predictive ability. Overall, it seems that all the tested rules have a poor out-of-sample performance. The out-of-sample mean excess return (measured by S_l^*) of the best rule in-sample for IBM is equivalent to an annual performance of 2.03% (these values are -2.62% for MSFT data and -0.27% for DJIA data), a poor performance regarding the computational efforts. Do these rules can beat a simple BH strategy? For IBM, the lowest bootstrap out-of-sample estimation of the BH performance is 16% (11.86% for MSFT data and 2.55% for DJIA data). The mean excess returns computed using this rule, over the initial datasets and in the bootstrap worlds, are displayed in Table 11. Focusing only on in-sample results, most of the rules would be able to beat the BH strategy. For IBM data, 9 rules perform better (if we use the bootstrap in-sample measure of the performance). For MSFT and DJIA, respectively 10 and 8 rules provide better results. However, whereas the BH strategy exhibits always a positive mean excess return, most strategies have a negative out-of-sample mean excess return when measured by S_l^* or $\bar{S}_l^+$(Figures 6). We do not have a formal statistical test to compare the strategies between each other, but the BH strategy has a (daily) mean excess return twice as big as the highest (out-of-sample) positive mean excess return ($\bar{S}_l^+$). It seems to indicate that none of the tested rules are able to beat this simple benchmark. It would be almost certain if we had included transaction costs. Overall, these results indicate that selecting *a priori* a rule and its optimal parametrization using the mean excess return criterion does not lead to a good out-of-sample profit. If some investors can make large profits using this approach, it may be due mostly to luck. These conclusions are the same as the ones obtained by Bajgrowicz and Scaillet [2012] (for the DJIA) and Kuang et al. [2014], who could not find any trading rule with a good out-of-sample performance. However, as pointed out in Bajgrowicz and Scaillet [2012], our results say little little about the existence of profitable strategies in other markets, using different trading frequencies or more sophisticated rules.

Using a Sharpe ratio criterion instead of the mean excess return, results are similar: the out-of-sample performances of all rules are lower than the one of a BH strategy, and negative most of the time. For IBM and DJIA, the best rules stay the same in-sample as well as out-of-sample, compared to the mean excess return criterion. For MSFT, the rule with the best out-of-sample Sharpe ratio is the modified filter rule, even though the simple filter rule and filter rule with a constant holding period have a performance very similar.

Regarding a possible difference between the residuals-based resampling and the block bootstrap procedure, results are globally alike. For the IBM data, we only observe a large difference for the moving average rules (both with band filter and with time delay), where the block bootstrap gives out-of-sample values way lower than the ones obtained with the residuals-based method. For the MSFT data, a similar effect happens for the simple filter rule. For the DJIA data, the block bootstrap procedure gives us way lower out-of-sample values for the simple filter rule than the residuals-based bootstrap procedure. Overall, these differences do not impact the final conclusions that all rules cannot beat the BH strategy if we look at their out-of-sample performance, measured by S_l^*.

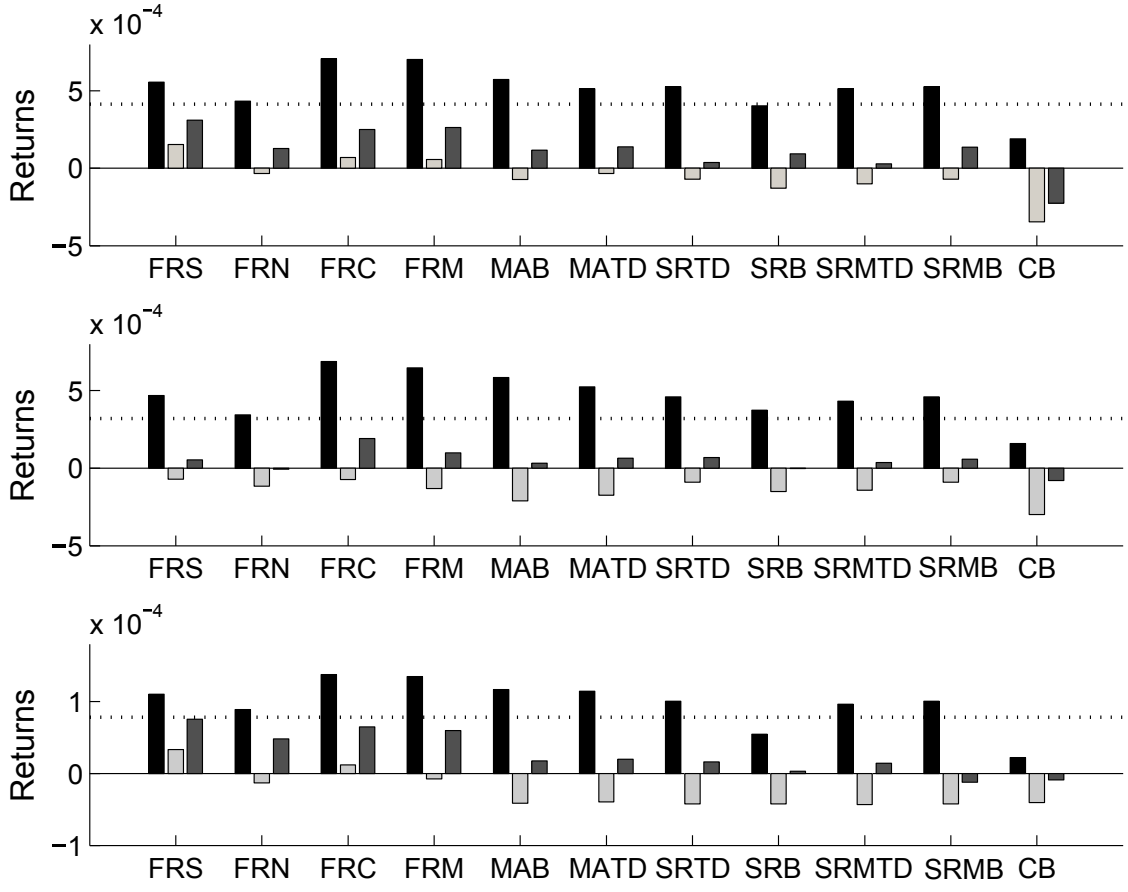

Figure 6: Comparisons of the different measures of the predictive ability (obtained with a residuals-based bootstrap), for IBM (top), MSFT (middle) and DJIA (bottom) data. Black: bootstrap in-sample performance. Light grey: S_l^*. Dark grey: $\bar{S}_l$. Black dotted: bootstrap in-sample performance of a BH strategy (i.e. mean excess return of the generated data).

4 Simulations

Eventually, we perform a small Monte Carlo study. The goal here is to determine if our bootstrap procedure provides an improvement compared to the sample splitting procedure, and to assess if our bootstrap estimators of the predictive ability are close the true predictive ability of the tested rules.

4.1 Simulation set-up

We simulate samples of prices where the log returns follow some ARMA-GARCH models. We assume that the innovations follow a standardized Student's t-distribution, to reflect the excess kurtosis found in empirical residuals [see, Bai et al., 2003, for a discussion on the subject]. We generate the data using the same models and the same parameters as the ones estimated for IBM, MSFT and DJIA data:

- DGP1: GARCH(1,1) with T(5) innovations.
- DGP2: ARMA(2,2)-GARCH(1,1) with T(4) innovations.
- DGP3: ARMA(3,3)-GARCH(2,1) with T(8) innovations.

For each DGP, we generate $N = 250$ samples of size $n = 1500$. The first 1000 observations are used as in-sample data ($n_{is} = 1000$) whereas the 500 last observations are used as out-of-sample data ($n_{oos} = 500$). For each sample, we select the optimal parameters of each rule using the pseudo in-sample data and we compute an estimator of the predictive ability for each rule with the pseudo out-of-sample data. By averaging over the N samples, we obtain a Monte Carlo estimator of the true predictive ability:

$$S_l^{MC} = \frac{1}{n_{oos} \times N} \sum_{i=1}^{N} \sum_{t=1001}^{n} Q(R_{i,t}, \bar{\theta}_{i,l}^{MC}) \tag{21}$$

where $Q(R_{i,t}, \bar{\theta}_{i,l}^{MC})$ is the value of the score function associated to the t^{th} trading day of the i^{th} sample, with the optimal parameters $\bar{\theta}_{i,l}^{MC}$ of the l^{th} rule estimated on the i^{th} pseudo in-sample dataset. We use this quantity as a benchmark: it is the best possible estimation of the predictive ability that we could obtain, if we were sure that the data follow exactly the supposed model. In practice, the sample splitting procedure is similar to using a single run of the Monte Carlo simulation to estimate the predictive ability. The Mean Squared Error (MSE) between the Monte Carlo estimator and the estimations of the predictive ability based on single samples can be found in Table 12 (MSE^{SS}). This quantity measures how far we are (on average) from the true predictive ability when we use the splitting procedure. It is given by

$$MSE_l^{SS} = \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1001}^{n} Q(R_{i,t}, \bar{\theta}_{i,l}^{MC})/n_{oos} - S_l^{MC} \right)^2. \tag{22}$$

For the score function $Q(\cdot)$, we use the mean excess return criterion with an annual risk-free rate of 0.7% (0.002% on a daily basis), assumed constant for simplicity.

What is the gain obtained with our bootstrap procedure, in term of MSE? For each sample, we use a residuals-based resampling procedure with $B = 100$ (we keep the number of bootstrap samples low to limit the computing time). We use the true models but the parameters are estimated on the pseudo in-sample data (i.e. on the first 1000 observations). Notice that, by doing so, we use less data than the splitting procedure (i.e. we use n_{is} observations instead of n observations). In practice, both the sample splitting and our procedure would rely on the same number of observations. Therefore, we can expect our procedure to be more efficient. We use both S_l^* (given by equation 11) and $\bar{S}_l$ (given by equation 12) as bootstrap estimators of the predictive ability. We compute:

$$MSE_l^{S^*} = \frac{1}{N}\sum_{i=1}^{N}\left(S_{i,l}^* - S_l^{MC}\right)^2, \tag{23}$$

$$MSE_l^{\bar{S}} = \frac{1}{N}\sum_{i=1}^{N}\left(\bar{S}_{i,l} - S_l^{MC}\right)^2. \tag{24}$$

4.2 Simulation results

Table 12 shows the MSE obtained with both procedures and the sample splitting approach. The columns RMSE give the ratio of the mean squared errors computed with the bootstrap approach and the sample splitting approach. They are computed in the following way:

$$RMSE_l^{S^*} = MSE_l^{S^*}/MSE_l^{SS}, \tag{25}$$

$$RMSE_l^{\bar{S}} = MSE_l^{\bar{S}}/MSE_l^{SS}. \tag{26}$$

A ratio below 1 indicates that our bootstrap approach has a MSE lower than the one obtained with the sample splitting procedure. We see that, for all trading rules and all DGP, our procedure decreases sharply the MSE compared to a sample splitting procedure, for all three DGPs and the 11 trading strategies. $MSE_l^{S^*}$ decreases in a range between 43.2% (for the simple filter rule in DGP2) to 93.9% (for the Channel Breakout rule in DGP3). Similarly, $MSE_l^{\bar{S}}$ decreases in a range between 37.4% (for the filter rule with constant holding period in the first DGP) and 80.8% (for the S&R rule with modified time delay). Overall, the decrease is stronger with the S_l^* estimator (the average decrease over all rules and all DGP is 80.7% for $MSE_l^{S^*}$ whereas it is 63.9% for $MSE_l^{\bar{S}}$). Therefore, it seems more interesting to use S_l^* instead of $\bar{S}_l$, to estimate the predictive ability. These results illustrate the superiority of our procedure over the sample splitting procedure. In particular, our bootstrap estimators of the predictive ability are way more efficient than

the one obtained by splitting the samples into two parts.

Figure 7 shows the out-of-sample mean excess returns of the rules for time horizons of varying lengths (from one day to 500 days), for DGP1. We see that the mean excess returns vary a lot for small time horizons, before converging to some values for horizons longer than 100 days. A similar effect is observed for the two other DGPs (they have been omitted for space considerations but are available upon demand to the authors). It illustrates that, for ARMA-GARCH-type models, the predictive abilities of the tested rules appear independent of the time horizon considered. Indeed, because the log returns are driven by a stationary process, the only thing that matters is the sample size, of both the training set and the validation set: the larger the training set, the more accurate we can capture the time dependencies in our estimation of the parameters; the larger the validation set, the more precisely we are able to estimate the expected value of the score function, for a given training sample size. Hence, using a rolling window with shorter subsamples of the training and validation sets should not improve the assessment of the true predictive ability. It would only increase the volatility of the estimator of the predictive ability. Hence, if the data follow exactly the specified GARCH process, none of the tested rules can beat the BH strategy, similarly to the conclusions of our empirical study.

Finally, we pick up at random one of the 250 samples. Then, we compare the Monte Carlo distribution of the predictive ability (obtained on the 250 samples) with the one obtained with the bootstrap procedure on this particular sample. Figure 8 shows this comparison for the eleven trading rules considered. We see that both distributions are pretty close to one another. It illustrates the fact that our bootstrap procedure can be a good substitute to the Monte Carlo simulation, impossible to perform in practice without supposing a distribution for the stochastic shocks.

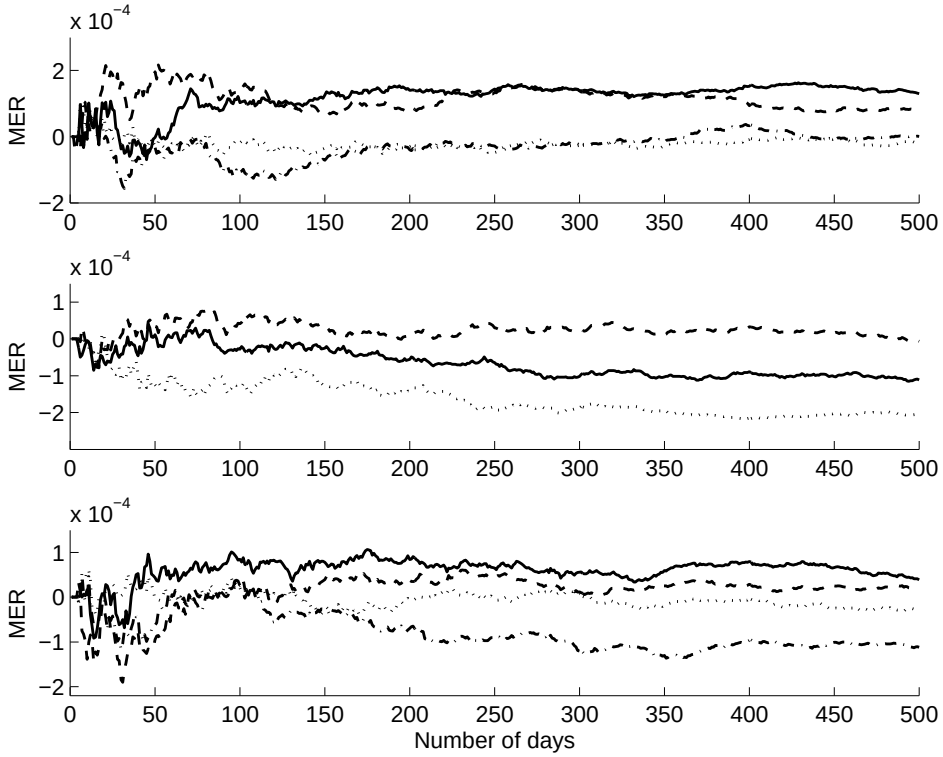

Figure 7: Out-of-sample mean excess return for the various strategies, obtained by Monte Carlo simulation for DGP1. On the x axis, time horizon (in days) used to compute the performance. On the y axis, daily mean excess returns. Top: filter rules (solid: FRS, dashed: FRM, dotted: FRC, dashed-dotted: FRN). Middle: moving average and CB rules (solid: MAB, dashed: MATD, dotted: CB). Bottom: S&R rules (solid: SRTD, dotted: SRMTD, dashed: SRB, dashed-dotted: SRMB).

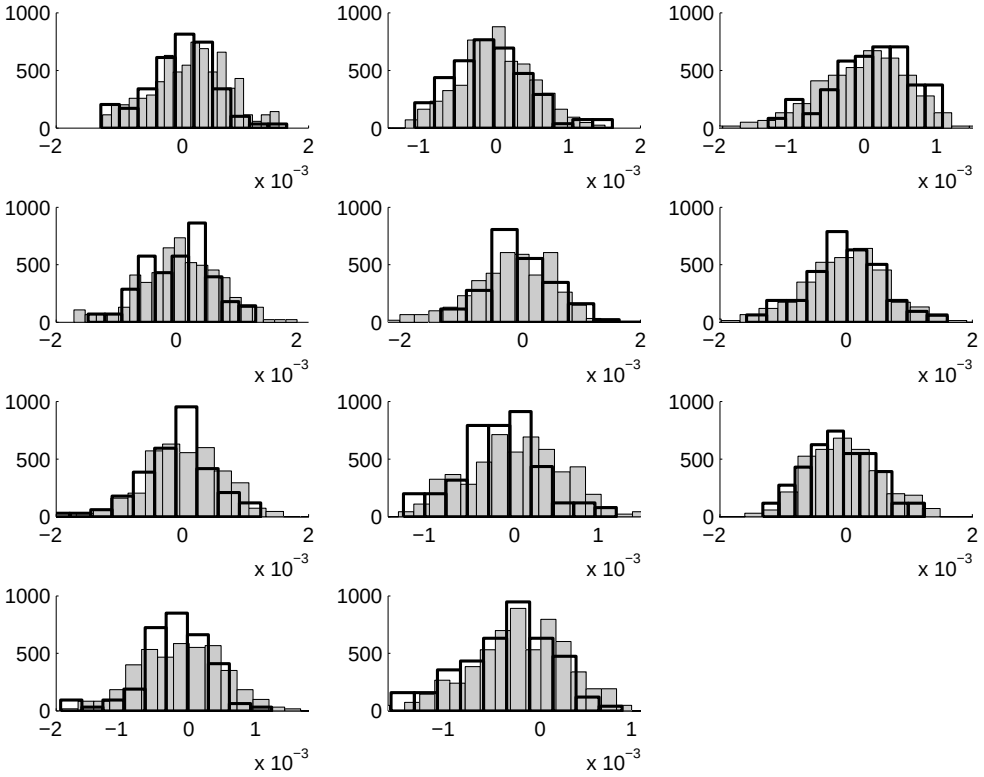

Figure 8: For DGP1, comparison between the true distribution of the out-of-sample predictive ability (in grey - obtained via MC simulation with 250 samples) and the bootstrap estimation of this distribution (transparent - i.e. the empirical distribution of $(1/V_b) \cdot \sum_{t=1}^{V_b} Q(\mathcal{R}^*_{b,t}, \bar{\theta}^*_{b,l})$), based on 100 resamples from a single sample of the MC simulation. On the horizontal axis: mean excess returns. The rules are displayed in the following order, from top left to bottom right: FRS, FRN, FRC, FRM, MAB, MATD, SRTD, SRB, SRMTD, SRMB and CB.

5 Conclusion

In this paper, we introduce a robust method to compute an estimator of the out-of-sample predictive ability of a trading rule. This measure is useful to compare different trading rules between each other, taking into account the parameters' selection. More precisely, we adapt a .632 bootstrap resampling procedure to the time series context. It enables to draw non-overlapping training sets and validation sets of data. We propose two ways to resample the data: either with a residuals-based resampling when some models can correctly fit the data, or with a block resampling when the modeling task is too difficult. With these

resampling techniques, we are able to keep intact the intrinsic time dependencies of the original data. Moreover, we provide a way to create bootstrap in- and out-of-sample time series of interest rate data, if a mean excess return criterion or a Sharpe ratio criterion are used as measures of the rules' performance.

Our method is an alternative that has several advantages over the simple sample splitting technique (with or without rolling window) traditionally used by researchers [see, e.g. Sullivan et al., 1999, Lukac et al., 1988, Allen and Karjalainen, 1999, Bajgrowicz and Scaillet, 2012, Kuang et al., 2014]. First, the obtained results are not dependent from some subjective cut-off points, because we use the whole dataset in the resampling procedure. Second, our bootstrap estimation is more robust because it does not rely on a single realization of a stochastic process. Instead, it offers a convenient way to generalize the splitting of the dataset into multiple subsamples exhibiting the same time dependencies and the same length. Moreover, thanks to the random resampling procedure, it incorporates new "scenarios" in our measure of the performance. These scenarios are useful to assess a trading rule on artificial but totally new data. Throughout a residuals-based resampling, we create new patterns of (stochastic) shocks and simultaneously keep the time dependencies intact. In this configuration, we can combine a large number of different pasts (our bootstrap training sets) with a large set of different futures (the bootstrap validation sets), although these sequences of events do not exist in our original time series.

We illustrate our methodology on three datasets (IBM, Microsoft and Dow Jones Industrial Average index daily stock returns), by comparing 11 trading rule specifications (that encompass 11,668 different parametrizations). We identify the best in-sample parametrizations and show that none of these parametrized rules seems able to beat a simple buy-and-hold strategy on out-of-sample data. Our results are in line with studies [Bajgrowicz and Scaillet, 2012, Sullivan et al., 1999, Brock et al., 1992, Kuang et al., 2014] that detected an in-sample superior performance of some trading rules but could not reproduce these results on out-of-sample data [Allen and Karjalainen, 1999, Bajgrowicz and Scaillet, 2012, Kuang et al., 2014]. These results appear to be robust to the resampling method used (block or residuals-based). However, these results say nothing about trading rules profitability in other markets, using other trading frequencies or more sophisticated rules.

One could argue, as in Bajgrowicz and Scaillet [2012], that shorter time horizons should be considered (e.g. a monthly basis) to better stick to practitioner's practices, instead of the very long time intervals that are used (i.e. 12 years of training sets and on average 6 years of validation sets). This is clearly one limitation of the empirical part of our study. Nevertheless, our procedure could be easily adapted to these requirements by using smaller initial dataset. However, as pointed out in Section 4 (and illustrated by Figure 7), if the returns are driven by some stationary processes, this exercise is pointless: reducing the sample size will only increase the variability of the results and will decrease the precision

of the estimated parameters.

Finally, a simulation study suggests that, under the assumption of a correctly specified model, our estimator of the out-of-sample performance is pretty good. On one side, our simulations show that our bootstrap measure of the performance is way better than the sample splitting measure, in term of MSE. On the other side, we observe that the bootstrap distribution of the out-of-sample performance is very close to the true distribution. These results illustrate that our bootstrap procedure is quite accurate and can be useful to rank trading strategies, as well as to measure their predictive ability.

Eventually, notice that we do not provide a valid statistical test to compare our bootstrap estimator of the prediction error across all rules. We believe that developing an extension of the Reality Check test might be a solution to this problem. However, it raises statistical questions that are beyond the scope of this paper. Nevertheless, such an interesting topic might be investigated in further researches.

Acknowledgements

J. Hambuckers acknowledges the support of the Belgian National Fund for Scientific Research (F.N.R.S) with a research fellow grant. The authors warmly thank the anonymous referee for her/his valuable suggestions.

References

F. Allen and R. Karjalainen. Using genetic algorithms to find technical trading rules. *Journal of Financial Economics*, 51(2):245–271, 1999.

X. Bai, J.R. Russel, and G.C. Tiao. Kurtosis of garch and stochastic volatility models with non-normal innovations. *Journal of Econometrics*, 114:349–360, 2003.

P. Bajgrowicz and O. Scaillet. Technical trading revisited: False discoveries, persistence tests, and transaction costs. *Journal of Financial Economics*, 106(3):473–491, 2012.

T.G. Bali and L. Wu. A comprehensive analysis of the short-term interest-rate dynamics. *Journal of Banking & Finance*, 30(4):1269–1290, 2006.

W. Brock, J. Lakonishok, and B. LeBaron. Simple technical trading rules and the stochastic properties of stock returns. *The Journal of Finance*, 47(5):1731–1764, 1992.

J.C. Cox, J.E. Ingersoll, and S.A. Ross. A theory of the term structure of interest rates. *Econometrica*, 53(2):385–407, 1985.

R. Davidson and J.G. MacKinnon. *Bootstrap Methods in Econometrics (Palgrave Handbook of Econometrics: Vol. 1 Econometric Theory)*. Palgrave Macmillan, First Edition, New York, 2006.

F. Diebold and C. Li. Forecasting the term structure of government bond yields. *Journal of Econometrics*, 130(2):337–364, 2006.

B. Efron. Bootstrap methods: another look at the jackknife. *Annals of Statistics*, 7(1): 1–26, 1979.

B. Efron. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331, 1983.

B. Efron and R. Tibshirani. *An introduction to the bootstrap*. New York, 1993.

B. Efron and R. Tibshirani. Improvements on cross-validation: The 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997.

R.F. Engle and G. Gonzalez-Rivera. Semiparametric arch models. *Journal of Business & Economic Statistics*, 9(4):345–359, 1991.

P. Falbo and C. Pelizzari. Stable classes of technical trading rules. *Applied Economics*, 43 (14):1769–1785, 2011.

J. Fang, B. Jacobsen, and Y. Qin. Predictability of the simple technical trading rules: An out-of-sample test. *Review of Financial Economics*, 23(1):30–45, 2014.

P. Hall and Q. Yao. Inference in arch and garch models with heavy-tailed errors. *Econometrica*, 71(1):285–317, 2003.

P.R. Hansen. A test for superior predictive ability. *Journal of Business & Economic Statistics*, 23(4):365–380, 2005.

P. Hsu, Y.-C. Hsu, and C.-M. Kuan. Testing the predictive ability of technical analysis using a new stepwise test without data snooping bias. *Journal of Empirical Finance*, 17 (3):471–484, 2010.

B. C. Kho. Time-varying risk premia, volatility, and technical trading rule profits: Evidence from foreign currency futures markets. *Journal of Financial Economics*, 41(2):249–290, 1996.

J.-P. Kreiss and E. Paparoditis. Bootstrap methods for dependent data: A review. *Journal of the Korean Statistical Society*, 40(4):357–378, 2011.

P. Kuang, M. Schröder, and Q. Wang. Illusory profitability of technical analysis in emerging foreign exchange markets. *International Journal of Forecasting*, 30(2):192–205, 2014.

H. R. Kunsch. The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics*, 17(3):1217–1241, 1989.

A. Lo and A. MacKinlay. Data-snooping biases in tests of financial asset pricing models. *The Review of Financial Studies*, 3(3):431–467, 1990.

L.P. Lukac, B.W. Brorsen, and S.H. Irwin. A test of futures market disequilibrium using twelve different technical trading systems. *Applied Economics*, 20(5):623–639, 1988.

C.-H. Park and S.H. Irwin. What do we know about the profitability of tehcnical analysis? *Journal of Economic Surveys*, 21(4):786–826, 2007.

D.N. Politis and J.P. Romano. The stationary bootstrap. *Journal of the American Statistical Association*, 89(428):1303–1313, 1994.

D.N. Politis and H. White. Automatic block-length selection for the dependent bootstrap. *Econometric Reviews*, 23(1):53–70, 2004.

J. P. Romano and M. Wolf. Stepwise multiple testing as formalized data snooping. *Econometrica*, 73(4):1237–1282, 2005.

L. Sarno, D.L. Thornton, and G. Valente. Federal funds rate prediction. *Journal of Money, Credit and Banking*, 37(3):449–471, 2005.

R. Sullivan, A. Timmermann, and H. White. Data-snooping, technical trading rule performance, and the bootstrap. *The Journal of Finance*, 54(5):1647–1691, 1999.

N. Taylor. The rise and fall of technical trading rule success. *Journal of Banking & Finance*, 40(0):286–302, 2014.

H. White. A reality check for data snooping. *Econometrica*, 68(5):1097–1126, 2000.

Tables

Data	$\hat{\omega}^{(m)}$	$\hat{\alpha}_1^{(m)}$	$\hat{\alpha}_2^{(m)}$	$\hat{\alpha}_3^{(m)}$	$\hat{\beta}_1^{(m)}$	$\hat{\beta}_2^{(m)}$	$\hat{\beta}_3^{(m)}$	$\hat{\omega}$	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\beta}$
IBM	0.0003	-	-	-	-	-	-	$9.2e^{-6}$	0.114	-	0.838
MSFT	0.0005	-0.732	-0.352	-	0.655	0.265	-	$3.9e^{-6}$	0.041	-	0.945
DJIA	0.0002	-0.166	-0.592	0.348	0.051	0.543	-0.3957	$4.6e^{-7}$	$1e^{-4}$	0.115	0.859

Table 1: Estimated ARMA and GARCH parameters for IBM, MSFT and DJIA time series, obtained quasi-maximum likelihood procedures.

Trading rule	Mean excess return (S_l^0)	Parameters
Filter rule simple	$5.7649e^{-4}$	$x = 0.005$, $y = 0.15$
Filter rule neutral	$4.0558e^{-4}$	$x = 0.2$, $y = 0.2$
Filter rule constant	$5.6298e^{-4}$	$c = 5$, $x = 0.04$, $y = 0.15$
Filter rule modified	$6.1827e^{-4}$	$e = 10$, $x = 0.005$, $y = 0.025$
MA time delay	$4.4277e^{-4}$	$n_{small} = 30$, $n_{large} = 40$, $d = 5$
MA filter band	$4.3592e^{-4}$	$n_{small} = 25$, $n_{large} = 40$, $c = 25$, $b = 0.001$
S&R time delay	$2.2057e^{-4}$	$c = 10$, $n = 2$, $d = 4$
S&R filter band	$4.7524e^{-4}$	$c = 50$, $n = 5$, $b = 0.001$
S&R modified time delay	$2.525e^{-4}$	$c = 25$, $e = 2$, $d = 5$
S&R modified filter band	$4.9223e^{-4}$	$c = 50$, $e = 20$, $b = 0.005$
Channel breakout	$-1.776e^{-5}$	$n = 10$, $x = 0.005$, $c = 50$, $b = 0.01$

Table 2: Optimal parameters for the mean excess return criterion, for IBM time series.

Trading rule	Sharpe ratio (S_l^0)	Parameters
Filter rule simple	0.0404	$x = 0.005$, $y = 0.15$
Filter rule neutral	0.0292	$x = 0.2$, $y = 0.2$
Filter rule constant	0.0396	$c = 20$, $x = 0.025$, $y = 0.75$
Filter rule modified	0.0436	$e = 10$, $x = 0.005$, $y = 0.025$
MA time delay	0.0323	$n_{small} = 30$, $n_{large} = 40$, $d = 5$
MA filter band	0.0319	$n_{small} = 25$, $n_{large} = 40$, $c = 25$, $b = 0.001$
S&R time delay	0.0159	$c = 10$, $n = 2$, $d = 4$
S&R filter band	0.0357	$c = 50$, $n = 5$, $b = 0.001$
S&R modified time delay	0.0185	$c = 25$, $e = 2$, $d = 5$
S&R modified filter band	0.0364	$c = 50$, $e = 20$, $b = 0.005$
Channel breakout	-0.0013	$n = 10$, $x = 0.005$, $c = 50$, $b = 0.01$

Table 3: Optimal parameters for the Sharpe ratio criterion, for IBM time series.

Trading rule	Mean excess return (S_l^0)	Parameters
Filter rule simple	$2.6816e^{-4}$	$x = 0.005$, $y = 0.15$
Filter rule neutral	$1.8285e^{-4}$	$x = 0.2$, $y = 0.2$
Filter rule constant	$6.4456e^{-4}$	$c = 20$, $x = 0.025$, $y = 0.025$
Filter rule modified	$4.9557e^{-4}$	$e = 15$, $x = 0.005$, $y = 0.15$
MA time delay	$4.4964e^{-4}$	$n_{small} = 20$, $n_{large} = 25$, $d = 5$
MA filter band	$4.7021e^{-4}$	$n_{small} = 15$, $n_{large} = 30$, $c = 10$, $b = 0.001$
S&R time delay	$3.3983e^{-4}$	$c = 50$, $n = 15$, $d = 2$
S&R filter band	$2.546e^{-4}$	$c = 50$, $n = 20$, $b = 0.015$
S&R modified time delay	$3.449e^{-4}$	$c = 50$, $e = 10$, $d = 2$
S&R modified filter band	$3.0982e^{-4}$	$c = 50$, $e = 10$, $b = 0.015$
Channel breakout	$2.9315e^{-4}$	$n = 5$, $x = 0.02$, $c = 50$, $b = 0.01$

Table 4: Optimal parameters for the mean excess returns criterion, for MSFT time series.

Trading rule	Sharpe ratio (S_l^0)	Parameters
Filter rule simple	0.0156	$x = 0.005$, $y = 0.15$
Filter rule neutral	0.0107	$x = 0.2$, $y = 0.2$
Filter rule constant	0.0378	$c = 20$, $x = 0.025$, $y = 0.025$
Filter rule modified	0.0293	$e = 15$, $x = 0.005$, $y = 0.15$
MA time delay	0.0264	$n_{small} = 20$, $n_{large} = 25$, $d = 5$
MA filter band	0.0276	$n_{small} = 15$, $n_{large} = 30$, $c = 10$, $b = 0.001$
S&R time delay	0.0201	$c = 50$, $n = 15$, $d = 2$
S&R filter band	0.0161	$c = 50$, $n = 20$, $b = 0.015$
S&R modified time delay	0.0205	$c = 50$, $e = 10$, $d = 2$
S&R modified filter band	0.0183	$c = 50$, $e = 10$, $b = 0.015$
Channel breakout	0.0175	$n = 5$, $x = 0.02$, $c = 50$, $b = 0.01$

Table 5: Optimal parameters for the Sharpe ratio criterion, for MSFT time series.

Trading rule	Mean excess return (S_l^0)	Parameters
Filter rule simple	$1.477e^{-4}$	$x = 0.09$, $y = 0.1$
Filter rule neutral	$1.535e^{-4}$	$x = 0.09$, $y = 0.1$
Filter rule constant	$1.55e^{-4}$	$x = 0.035$, $y = 0.04$, $c = 50$
Filter rule modified	$1.743e^{-4}$	$x = 0.035$, $y = 0.04$, $e = 5$
MA time delay	$1.192e^{-4}$	$n_{small} = 40$, $n_{large} = 150$, $d = 5$
MA filter band	$1.215e^{-4}$	$n_{small} = 10$, $n_{large} = 100$, $c = 50$, $b = 0.001$
S&R time delay	$1.164e^{-4}$	$c = 50$, $n = 20$, $d = 2$
S&R filter band	$8.09e^{-5}$	$c = 50$, $n = 5$, $b = 0.001$
S&R modified time delay	$1.13e^{-4}$	$c = 20$, $e = 5$, $d = 2$
S&R modified filter band	$3.94e^{-5}$	$c = 5$, $e = 200$, $b = 0.001$
Channel breakout	$4.55e^{-5}$	$n = 25$, $x = 0.005$, $c = 25$, $b = 0.04$

Table 6: Optimal parameters for the MER criterion, for DJIA time series.

Trading rule	Sharpe ratio (S_l^0)	Parameters
Filter rule simple	0.0297	$x = 0.09$, $y = 0.1$
Filter rule neutral	0.0309	$x = 0.09$, $y = 0.1$
Filter rule constant	0.0309	$x = 0.035$, $y = 0.04$, $c = 50$
Filter rule modified	0.039	$x = 0.035$, $y = 0.04$, $e = 5$
MA time delay	0.0238	$n_{small} = 40$, $n_{large} = 150$, $d = 5$
MA filter band	0.0244	$n_{small} = 10$, $n_{large} = 100$, $c = 50$, $b = 0.001$
S&R time delay	0.0234	$c = 50$, $n = 20$, $d = 2$
S&R filter band	0.0164	$c = 50$, $n = 5$, $b = 0.001$
S&R modified time delay	0.0227	$c = 20$, $e = 5$, $d = 2$
S&R modified filter band	0.0079	$c = 5$, $e = 200$, $b = 0.001$
Channel breakout	0.0137	$n = 25$, $x = 0.005$, $c = 25$, $b = 0.04$

Table 7: Optimal parameters for the Sharpe ratio criterion, for DJIA time series.

Trading rule	MER	$S_l^{0,*}$	S_l^*	$\bar{S}_l$	$\bar{S}_l^+$	SR	$S_l^{0,*}$	S_l^*	$\bar{S}_l$	$\bar{S}_l^+$
Filter rule simple		$5.556e^{-4}$	$1.539e^{-4}$	$3.094e^{-4}$	$1.539e^{-4}$		0.0406	0.011	0.022	0.011
Filter rule neutral		$4.333e^{-4}$	$-0.343e^{-4}$	$1.276e^{-4}$	$-0.343e^{-4}$		0.0394	−0.004	0.008	−0.004
Filter rule constant		$7.059e^{-4}$	$0.693e^{-4}$	$2.510e^{-4}$	$0.693e^{-4}$		0.0513	0.005	0.018	0.005
Filter rule modified		$7.016e^{-4}$	$0.555e^{-4}$	$2.626e^{-4}$	$0.555e^{-4}$		0.0513	0.005	0.019	0.005
MA filter band		$5.733e^{-4}$	$-0.722e^{-4}$	$1.173e^{-4}$	$-0.722e^{-4}$		0.0416	−0.005	0.009	−0.005
MA time delay		$5.133e^{-4}$	$-0.338e^{-4}$	$1.390e^{-4}$	$-0.338e^{-4}$		0.0373	−0.002	0.011	−0.002
S&R time delay		$5.265e^{-4}$	$-0.711e^{-4}$	$0.362e^{-4}$	$-0.711e^{-4}$		0.0385	−0.006	0.002	−0.006
S&R filter band		$4.023e^{-4}$	$-1.285e^{-4}$	$0.937e^{-4}$	$-1.285e^{-4}$		0.0331	−0.008	0.008	−0.008
S&R modified time delay		$5.125e^{-4}$	$-1.016e^{-4}$	$0.287e^{-4}$	$-1.016e^{-4}$		0.0375	−0.007	0.002	−0.007
S&R modified filter band		$5.265e^{-4}$	$-0.711e^{-4}$	$1.362e^{-4}$	$-0.711e^{-4}$		0.0385	−0.006	0.010	−0.006
Channel breakout		$1.907e^{-4}$	$-3.463e^{-4}$	$-2.254e^{-4}$	$-3.463e^{-4}$		0.0154	−0.027	−0.018	−0.027

Table 8: Values of the mean excess return (MER) and the Sharpe ratio (SR), for the IBM daily time series, obtained using a residuals-based resampling method. $S_l^{0,*}$ stands for $\sum_{b=1}^{B}\sum_{t=h+1}^{n} Q(R_{b,t}^*, \bar{\theta}_{b,l}^*)/(B(n-h))$ and is the bootstrap in-sample performance. See above for the definition of S_l^*, $\bar{S}_l$ and $\bar{S}_l^+$.

Trading rule	MER	$S_l^{0,*}$	S_l^*	$\bar{S}_l$	$\bar{S}_l^+$	SR	$S_l^{0,*}$	S_l^*	$\bar{S}_l$	$\bar{S}_l^+$
Filter rule simple		$4.689e^{-4}$	$-0.710e^{-4}$	$0.538e^{-4}$	$0.538e^{-4}$		0.030	−0.004	0.003	0.003
Filter rule neutral		$3.429e^{-4}$	$-1.167e^{-4}$	$-0.064e^{-4}$	$-1.167e^{-4}$		0.030	−0.014	−0.005	−0.014
Filter rule constant		$6.877e^{-4}$	$-0.727e^{-4}$	$1.912e^{-4}$	$-0.727e^{-4}$		0.043	−0.004	0.011	−0.004
Filter rule modified		$6.454e^{-4}$	$-1.317e^{-4}$	$0.992e^{-4}$	$-1.317e^{-4}$		0.041	−0.002	0.010	−0.002
MA filter band		$5.840e^{-4}$	$-2.108e^{-4}$	$0.322e^{-4}$	$-2.108e^{-4}$		0.037	−0.013	0.002	−0.013
MA time delay		$5.230e^{-4}$	$-1.734e^{-4}$	$0.635e^{-4}$	$-1.734e^{-4}$		0.033	−0.010	0.004	−0.010
S&R time delay		$4.586e^{-4}$	$-0.907e^{-4}$	$0.677e^{-4}$	$-0.907e^{-4}$		0.032	−0.004	0.005	−0.004
S&R filter band		$3.730e^{-4}$	$-1.509e^{-4}$	$-0.017e^{-4}$	$-1.338e^{-4}$		0.027	−0.012	−0.001	−0.010
S&R modified time delay		$4.308e^{-4}$	$-1.431e^{-4}$	$0.365e^{-4}$	$-1.431e^{-4}$		0.030	−0.009	0.002	−0.009
S&R modified filter band		$4.586e^{-4}$	$-0.907e^{-4}$	$0.567e^{-4}$	$-0.358e^{-4}$		0.032	−0.014	−0.002	−0.010
Channel breakout		$1.589e^{-4}$	$-2.985e^{-4}$	$-0.808e^{-4}$	$-2.985e^{-4}$		0.011	−0.020	−0.006	−0.020

Table 9: Values of the mean excess return (MER) and the Sharpe ratio (SR), for the MSFT daily time series, obtained using a residuals-based resampling method.

Trading rule	MER	$S_l^{0,*}$	S_l^*	$\bar{S}_l$	$\bar{S}_l^+$	SR	$S_l^{0,*}$	S_l^*	$\bar{S}_l$	$\bar{S}_l^+$
Filter rule simple		$1.101e^{-4}$	$0.334e^{-4}$	$0.755e^{-4}$	$0.334e^{-4}$		0.028	0.008	0.015	0.008
Filter rule neutral		$0.887e^{-4}$	$-0.131e^{-4}$	$0.482e^{-4}$	$-0.131e^{-4}$		0.030	−0.007	0.007	−0.007
Filter rule constant		$1.375e^{-4}$	$0.120e^{-4}$	$0.646e^{-4}$	$0.120e^{-4}$		0.033	0.004	0.014	0.004
Filter rule modified		$1.350e^{-4}$	$-0.074e^{-4}$	$0.595e^{-4}$	$-0.074e^{-4}$		0.033	−0.001	0.012	−0.001
MA time delay		$1.168e^{-4}$	$-0.414e^{-4}$	$0.177e^{-4}$	$-0.414e^{-4}$		0.028	−0.009	0.004	−0.009
MA filter band		$1.145e^{-4}$	$-0.393e^{-4}$	$0.199e^{-4}$	$-0.393e^{-4}$		0.027	−0.008	0.005	−0.008
S&R time delay		$1.004e^{-4}$	$-0.421e^{-4}$	$0.162e^{-4}$	$-0.421e^{-4}$		0.024	−0.010	0.003	−0.010
S&R filter band		$0.544e^{-4}$	$-0.423e^{-4}$	$0.030e^{-4}$	$-0.423e^{-4}$		0.016	−0.012	−0.002	−0.012
S&R modified time delay		$0.962e^{-4}$	$-0.430e^{-4}$	$0.144e^{-4}$	$-0.430e^{-4}$		0.023	−0.010	0.002	−0.010
S&R modified filter band		$1.004e^{-4}$	$-0.421e^{-4}$	$-0.121e^{-4}$	$-0.421e^{-4}$		0.024	−0.010	0.003	−0.010
Channel breakout		$0.221e^{-4}$	$-0.404e^{-4}$	$-0.088e^{-4}$	$-0.404e^{-4}$		0.014	−0.015	−0.005	−0.015

Table 10: Values of the mean excess return (MER) and the Sharpe ratio (SR), for the DJIA daily time series, obtained with a residuals-based resampling method.

Stat.	MER	S_l^0	$S_l^{0,*}$	$\bar{S}_l^*$	SR	S_l^0	$S_l^{0,*}$	$\bar{S}_l^*$
IBM (res)		$4.2530e^{-4}$	$4.1278e^{-4}$	$5.1952e^{-4}$		0.0297	0.0270	0.0288
IBM (block)		$4.2530e^{-4}$	$4.3403e^{-4}$	$4.0675e^{-4}$		0.0297	0.0305	0.0285
MSFT (res)		$3.1733e^{-4}$	$3.1990e^{-4}$	$3.2073e^{-4}$		0.0184	0.0204	0.0197
MSFT (block)		$3.1733e^{-4}$	$3.4084e^{-4}$	$3.0792e^{-4}$		0.0184	0.02	0.0182
DJIA (res)		$7.7108e^{-5}$	$7.8265e^{-5}$	$8.2332e^{-5}$		0.0151	0.0191	0.0190
DJIA (block)		$7.7108e^{-5}$	$7.9296e^{-5}$	$6.9151e^{-5}$		0.0151	0.0160	0.0146

Table 11: Mean excess return (MER) and Sharpe ratio (SR) for a buy-and-hold strategy. First column: performance obtained on the initial datasets. Second and third column: average mean excess return obtained on the bootstrap training sample and the bootstrap validation sets. These quantities are computed for both a residuals-based procedure (*res*) and a block bootstrap procedure (*block*).

DGP 1

	S^{MC}	MSE^{SS}	MSE^{S^*}	$MSE^{\bar{S}}$	$RMSE^{S^*}$	$RMSE^{\bar{S}}$
FRS	$1.31e^{-4}$	$3.57e^{-7}$	$1.16e^{-7}$	$1.57e^{-7}$	0.324	0.441
FRN	$-1.77e^{-5}$	$2.86e^{-7}$	$4.58e^{-8}$	$1.28e^{-7}$	0.160	0.446
FRC	$1.59e^{-6}$	$4.43e^{-7}$	$8.30e^{-8}$	$2.77e^{-7}$	0.187	0.626
FRM	$7.78e^{-5}$	$4.28e^{-7}$	$5.69e^{-8}$	$1.77e^{-7}$	0.133	0.414
MAB	$-1.54e^{-4}$	$5.28e^{-7}$	$5.56e^{-8}$	$2.87e^{-7}$	0.105	0.543
MATD	$-9.83e^{-6}$	$5.36e^{-7}$	$4.54e^{-8}$	$1.70e^{-7}$	0.085	0.317
SRTD	$2.89e^{-5}$	$4.20e^{-7}$	$3.75e^{-8}$	$1.30e^{-7}$	0.089	0.310
SRB	$1.06e^{-5}$	$3.69e^{-7}$	$2.72e^{-8}$	$7.09e^{-8}$	0.074	0.192
SRMTD	$-2.30e^{-5}$	$3.93e^{-7}$	$3.30e^{-8}$	$1.54e^{-7}$	0.084	0.391
SRMB	$-1.20e^{-4}$	$4.30e^{-7}$	$4.33e^{-8}$	$1.84e^{-7}$	0.101	0.428
CB	$-2.19e^{-4}$	$2.94e^{-7}$	$5.02e^{-8}$	$1.21e^{-7}$	0.171	0.413

DGP 2

	S^{MC}	MSE^{SS}	MSE^{S^*}	$MSE^{\bar{S}}$	$RMSE^{S^*}$	$RMSE^{\bar{S}}$
FRS	$1.17e^{-4}$	$6.20e^{-8}$	$3.52e^{-8}$	$3.20e^{-8}$	0.568	0.516
FRN	$3.81e^{-5}$	$6.07e^{-8}$	$1.19e^{-8}$	$1.99e^{-8}$	0.195	0.328
FRC	$9.04e^{-5}$	$7.74e^{-8}$	$2.46e^{-8}$	$3.27e^{-8}$	0.318	0.422
FRM	$7.17e^{-5}$	$6.37e^{-8}$	$2.21e^{-8}$	$3.00e^{-8}$	0.346	0.470
MAB	$1.93e^{-5}$	$1.05e^{-7}$	$1.70e^{-8}$	$3.52e^{-8}$	0.162	0.335
MATD	$4.84e^{-5}$	$9.73e^{-8}$	$1.83e^{-8}$	$3.34e^{-8}$	0.188	0.344
SRTD	$2.73e^{-5}$	$6.93e^{-8}$	$1.26e^{-8}$	$2.45e^{-8}$	0.181	0.353
SRB	$-3.09e^{-5}$	$6.39e^{-8}$	$6.97e^{-9}$	$1.67e^{-8}$	0.109	0.261
SRMTD	$1.16e^{-5}$	$7.84e^{-8}$	$1.09e^{-8}$	$2.68e^{-8}$	0.139	0.342
SRMB	$-2.13e^{-5}$	$8.90e^{-8}$	$1.42e^{-8}$	$2.93e^{-8}$	0.160	0.329
CB	$-1.22e^{-4}$	$4.08e^{-8}$	$2.95e^{-9}$	$1.11e^{-8}$	0.072	0.273

DGP 3

	S^{MC}	MSE^{SS}	MSE^{S^*}	$MSE^{\bar{S}}$	$RMSE^{S^*}$	$RMSE^{\bar{S}}$
FRS	$8.52e^{-5}$	$3.06e^{-8}$	$1.54e^{-8}$	$1.09e^{-8}$	0.503	0.356
FRN	$1.07e^{-5}$	$2.31e^{-8}$	$5.38e^{-9}$	$7.06e^{-9}$	0.233	0.306
FRC	$6.37e^{-5}$	$3.42e^{-8}$	$1.08e^{-8}$	$1.06e^{-8}$	0.316	0.311
FRM	$5.23e^{-5}$	$2.99e^{-8}$	$8.60e^{-9}$	$8.79e^{-9}$	0.288	0.294
MAB	$1.09e^{-5}$	$4.17e^{-8}$	$7.54e^{-9}$	$1.22e^{-8}$	0.181	0.292
MATD	$1.61e^{-5}$	$4.05e^{-8}$	$8.41e^{-9}$	$1.46e^{-8}$	0.208	0.361
SRTD	$-1.44e^{-5}$	$3.37e^{-8}$	$5.48e^{-9}$	$1.26e^{-8}$	0.163	0.374
SRB	$-2.92e^{-5}$	$2.63e^{-8}$	$3.33e^{-9}$	$5.60e^{-9}$	0.127	0.213
SRMTD	$-1.63e^{-5}$	$3.27e^{-8}$	$5.24e^{-9}$	$1.27e^{-8}$	0.161	0.389
SRMB	$-2.08e^{-5}$	$3.92e^{-8}$	$6.96e^{-9}$	$1.08e^{-8}$	0.178	0.275
CB	$-8.75e^{-5}$	$1.93e^{-8}$	$1.17e^{-9}$	$4.99e^{-9}$	0.061	0.259

Table 12: Mean squared error between the Monte Carlo estimator of the predictive ability S^{MC} and the estimators obtained either with the sample splitting procedure (MSE^{SS}) or the residuals-based bootstrap approach (MSE^{S^*} and $MSE^{\bar{S}}$), for the 11 trading strategies.

Appendix

A Trading rule descriptions and parameters

Abbreviation 1	Abbreviation 2	Full name of the rule
FRS	Filter rule simple	Filter rule, simple extrema
FRN	Filter rule neutral	Filter rule, simple extrema, neutral position
FRC	Filter rule constant	Filter rule, simple extrema, constant holding
FRM	Filter rule modified	Filter rule, modified extrema
MATD	MA time delay	moving average, time delay
MAB	MA filter band	moving average, filter band, constant holding period
SRTD	S&R time delay	S&R, simple extrema, time delay, constant holding period
SRB	S&R filter band	S&R, simple extrema, filter band, constant holding period
SRMTD	S&R modified time delay	S&R, modified extrema, time delay, constant holding period
SRMB	S&R modified filter band	S&R, modified extrema, filter band, constant holding period
CB	Channel breakout	Channel breakout, constant holding period, filter band, simple extrema

Table A.1: Correspondence table between rules abbreviations and their full descriptions.

A.1 Filter rules

A filter rule generates buy and sell signals when it detects a sufficient large price reversing, compared to an upper (respectively lower) threshold. The filter rule is supposed to smooth the price shift detection to ignore too small price variations, considered as noise. When we record a sufficient decrease, compared to the upper threshold, we sell the asset because it indicates a downward trend. Similarly, when we register a sufficient increase compared to the lower threshold, we buy the asset to benefit from the upward trend. We use two different ways to compute these threshold along time. The first way consists in taking the most recent price that is higher (resp. lower) than its e preceding closing prices. The second one consists in taking the highest (respectively lowest) price registered while holding a long (respectively short) position in the asset. Every time the position is changed, upper and lower thresholds are updated. Other features are also included: a minimal holding period (i.e. when entering a position, we are obliged to keep it at least c time periods) or the possibility of a neutral position (i.e. the possibility to not have any position in the asset).

x= change in security price (percentage) required to initiate a long position,
y= change in security price (percentage) required to initiate a short position,
e=parameter entering the second definition of extrema (number of previous days that should be higher (lower) than the subsequent high (low) used to initiate a position),
c=minimum holding period,
$x = \{0.005, 0.01, 0.015, 0.02, 0.025,$
$0.03, 0.035, 0.04, 0.045, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.12, 0.14, 0.16, 0.18, 0.2,$
$0.25, 0.3, 0.4, 0.5\}$,

$y = \{0.005, 0.01, 0.015, 0.02, 0.025, 0.03, 0.04, 0.05, 0.075, 0.1, 0.15, 0.2\}$,
$e = \{1, 2, 3, 4, 5, 10, 15, 20\}$,
$c = \{5, 10, 20, 25, 50\}$,

A.2 Moving averages rules

It exists numerous specifications of trading rules based on moving averages but the most famous are the moving average cross-over rules. We focus on that type of rule, where buy and sell signals are generated by crossovers of a slow moving average by a fast moving average. A fast moving average is based on fewer days than a slow moving average. The moving average with a window size of size n for a particular time t is simply the arithmetic mean of the closing prices over the previous n days, including the current day. We include two refinements of this rule: a band filter, that imposes to the slow moving average to be b % higher (lower) than the fast moving average, to generate a signal; and a time delay, that imposes to the signal to persist d days before effectively entering a new position (in the case of a buy signal, the slow moving average must be above the slow one d days before effectively buying).

n=number of days in a moving average,
b=band filter size (%),
d=number of days for the time delay filter,
c=minimum holding period,
$n = \{2, 5, 10, 15, 20, 25, 30, 40, 50, 75, 100, 125, 150, 200, 250\}$,
$b = \{0.001, 0.005, 0.01, 0.015, 0.02, 0.03, 0.04, 0.05\}$,
$d = \{2, 3, 4, 5\}$,
$c = \{5, 10, 20, 25, 50\}$,

A.3 Support and resistance rules

Similarly, the support and resistance (S&R) rule generates buy and sell signals when the closing price is higher (lower) than the highest (lowest) price over the previous n days. An alternative definition of high and low, as in the filter rule, can be used. Similar to the moving average rules, we impose a band filter or a time delay.

n=number of days in the support and resistance range,
e=parameter entering the second definition of extrema (number of previous days that should be higher (lower) than the subsequent high (low) used to initiate a position),
b=band filter size (%),
d=number of days for the time delay filter,

c=minimum holding period,
$n = \{2, 5, 10, 15, 20, 25, 50, 100, 150, 200, 250\}$,
$e = \{2, 3, 4, 5, 10, 20, 25, 50, 100, 200\}$,
$b = \{0.001, 0.005, 0.01, 0.015, 0.02, 0.03, 0.04, 0.05\}$,
$d = \{2, 3, 4, 5\}$,
$c = \{5, 10, 25, 50\}$,

A.4 Channel breakouts rule

A channel breakouts rule is based on the difference between local minimum and local maximum prices. A channel is said to occur when the highest price over the previous n days (excluding the price at time t) is within x percent of the lowest price over the same n previous days. A breakout occurs when the price at time t is higher or lower than these high or low prices, generating buy or sell signals.

n=number of days of the channel,
x=difference between high and low prices (%),
b=band filter size (%),
c=minimum holding period,
$n = \{5, 10, 15, 20, 25, 50, 100, 150, 200, 250\}$,
$x = \{0.005, 0.01, 0.02, 0.03, 0.05, 0.075, 0.1, 0.15\}$,
$b = \{0.001, 0.005, 0.01, 0.015, 0.02, 0.03, 0.04, 0.05\}$,
$c = \{5, 10, 25, 50\}$,

B Complementary data analysis

For the IBM time series, a graphical analysis revealed that no sample autocorrelation function (SACF) seems significantly different from zero. LM tests up to lag 20 could not reject the null hypothesis of no serial autocorrelation. For the MSFT and DJIA time series, we tried different ARMA models with various combinations of p and q, for $p = 0, \cdots, 3$ and $q = 0, \cdots, 3$. We used the AIC criterion, as well as p-values of LM tests to select the combination that removed best the time dependencies. We performed the LM tests with 8 and 11 lags. Also, we check for a possible unit-root in the mean process by using augmented Dickey-Fuller tests. We conclude that the best model for MSFT is an ARMA(2,2)-GARCH(1,1) model whereas the best model for DJIA is an ARMA(3,3)-GARCH(2,1) model. For the DJIA, even if a small time dependencies in the mean remains, we prefer a parsimonious model rather than a more complicated one. Further details regarding the performed tests are available upon demand to the authors.

C Interest rate model

In our application, we use Diebold and Li [2006] dynamic Nelson-Siegel model. We describe in short the estimation method and the hypotheses of this model.

The Nelson-Siegel model gives us, at a given time, the entire yield curve, i.e. yield values as a function of the maturity. Diebold and Li [2006] suppose that this relationship evolves along time and can be described by the following equation:

$$r_t^f(\tau^r) = \beta_{1,t}^r + \beta_{2,t}^r \left(\frac{1 - \mathrm{e}^{-\lambda^r \tau^r}}{\lambda^r \tau^r} \right) + \beta_{3,t}^r \left(\frac{1 - \mathrm{e}^{-\lambda^r \tau^r}}{\lambda^r \tau^r} - \mathrm{e}^{-\lambda^r \tau^r} \right), \tag{27}$$

where τ^r is the maturity, λ^r, β_1^r, β_2^r and β_3^r are parameters. λ^r is fixed along time and chosen equal to the value that maximizes $\left(\frac{1-\mathrm{e}^{-\lambda^r \tau^r}}{\lambda^r \tau^r} - \mathrm{e}^{-\lambda^r \tau^r} \right)$ for τ^r equal to 30 months. They also argue that the parameters $\beta_{i,t}^r$, $i = 1, \cdots, 3$ can be interpreted as factors corresponding to the level, the slope and the curvature of the yield curve. They propose to model these factors using AR(1) models. More formally, they suppose that:

$$\beta_{i,t}^r = \alpha_i^r \beta_{i,t-1}^r + \epsilon_{i,t},\ i = 1, \cdots, 3, \tag{28}$$

where α_i^r is the AR parameter of factor i and $\epsilon_{i,t}$ are i.i.d. mean-zero stochastic innovations of the i^{th} factor at time t. Diebold and Li [2006] propose the following procedure to obtain estimators of $\beta_{i,t}^r$ and α_i^r, $i = 1, \cdots, 3$ and $t = 1, \cdots, n$:

1. for $t = 1, \cdots, n$, get OLS estimators of $\beta_{i,t}^r$, $i = 1, \cdots, 3$,
2. fit AR(1) models on the obtained series of $\hat{\beta}_{i,t}^r$, $i = 1, \cdots, 3$.

In our application, we are interested in the yield for a single maturity. However, we need several yields (at least 3) at each time t, to obtain OLS estimator of $\beta_{i,t}^r$, $i = 1, 2, 3$. Therefore, we add 10 other daily time series with different maturities (1, 3, 6, 12, 24, 36, 60,84, 120 and 240 months), which are treasury rates time series extracted from the web site of the US Federal reserve (http://www.federalreserve.gov). Figure C.1 shows these interest rates. Figure C.2 shows the estimated latent factors using this method and these datasets. Figure C.3 shows the residuals of this model. We observe that, for the daily maturity, the data during the last financial crisis (2008-2010) seem not well filtered. This is a limitation of this approach.

Using augmented Dickey-Fuller tests at various lags, we cannot reject the hypothesis of a unit root in the processes driving the latent factors. A random walk could be a suitable model, but we prefer to use an AR(1) model, with its parameters estimated using classical maximum likelihood techniques. Table C.1 shows the estimated parameters. Despite the

presence of heteroscedasticity, we prefer to use a parsimonious model to stay in line with Diebold and Li [2006] approach. In our simulation framework, we resample the empirical residuals to create new series of stochastic innovations. To prevent the yields to become negative, we also apply the following constraint on the slope coefficients:

$$\beta_{2,t}^r \geq \frac{-\beta_{1,t}^r - v_2 \cdot \beta_{3,t}^r}{v_1} \tag{29}$$

where $v_1 = \frac{1-\mathrm{e}^{-\lambda^r \tau^r}}{\lambda^r \tau^r}$ and $v_2 = \frac{1-\mathrm{e}^{-\lambda^r \tau^r}}{\lambda^r \tau^r} - \mathrm{e}^{-\lambda^r \tau^r}$. When $\beta_{2,t}^r$ does not fulfil this condition, we simply put its value equal to $\frac{-\beta_{1,t}^r - v_2 \cdot \beta_{3,t}^r}{v_1}$.

λ^r is chosen equal to 0.0598 throughout our application and is simply obtained by solving:

$$\frac{\tau^r \, \mathrm{e}^{-\lambda^r \tau^r}}{\lambda^r \tau^r} - \tau^r \frac{(1 - \mathrm{e}^{-\lambda^r \tau^r})}{(\lambda^r)^2 (\tau^r)^2} + \tau^r \, \mathrm{e}^{-\lambda^r \tau^r} = 0$$

for $\tau^r = 30$ (in months).

Factor	constant	AR(1) parameter	ADF p-value
$\beta_{1,t}^r$	0.0136	0.9969	0.3177
$\beta_{2,t}^r$	-0.0036	0.9988	0.6650
$\beta_{3,t}^r$	-0.0143	0.9961	0.2981

Table C.1: Estimated parameters of AR(1) models for the various latent factors. The last column gives the p-values of the Augmented Dickey-Fuller (ADF) tests with 8 lags (approximately the logarithm of the size of our time series).

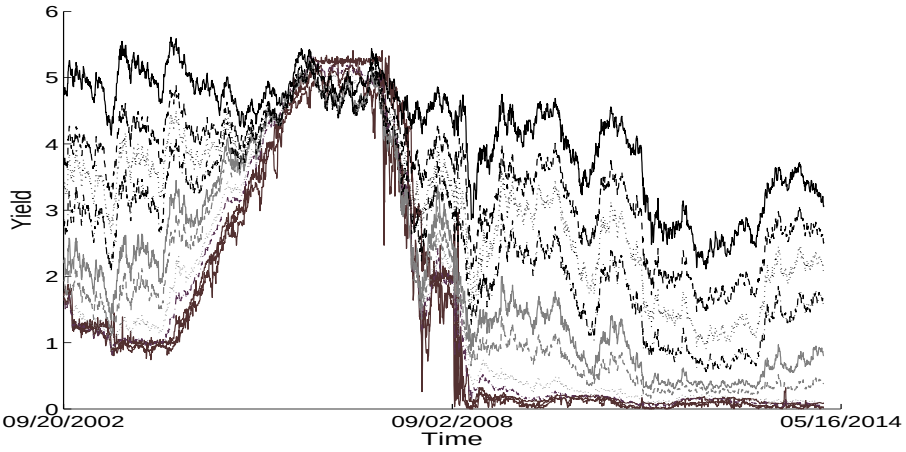

Figure C.1: Treasury rates for various maturities (1 day, 1 month, 3 months, 6 months, 1 year, 2 years, 3 years, 5 years, 7 years, 10 years and 20 years) for the period 09/20/2002-05/16/2014.

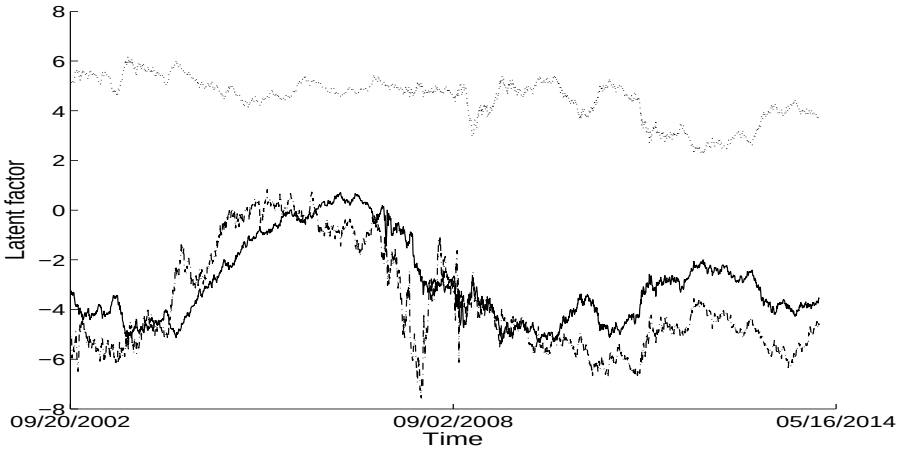

Figure C.2: Time series of estimated latent factors for level ($\hat{\beta}^r_{1,t}$, dotted), slope ($\hat{\beta}^r_{2,t}$, solid) and curvature ($\hat{\beta}^r_{3,t}$, dash-dotted) parameters, $t = 1, \cdots, n$.

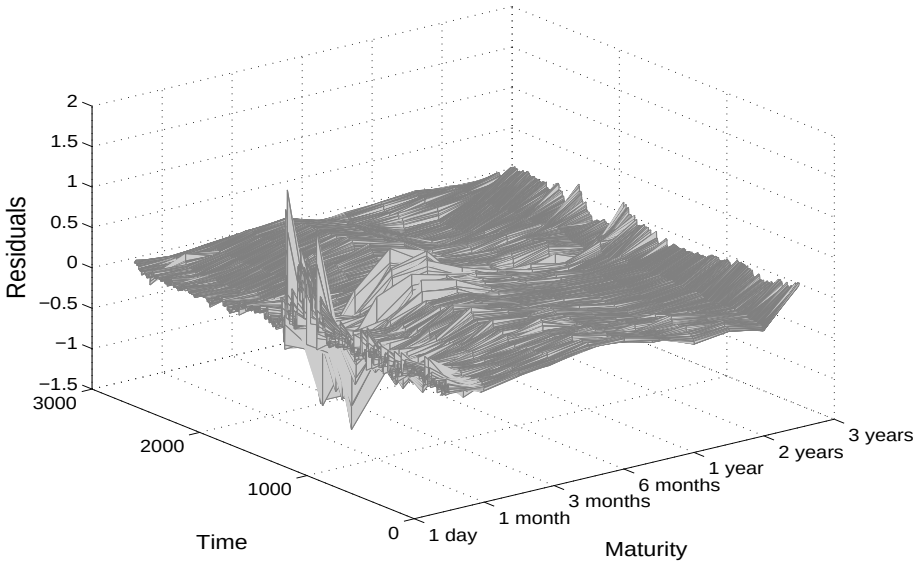

Figure C.3: Response surface of the residuals for the interest rates model. In abscissa, the maturity. In ordinate, time.

D Additional results

First, some words regarding the .632 weights. As explained in Section 2.1, the factors .632 and .368 arise because S_l^* is too pessimistic (for a prediction error, $S_l^* - S_l^0$ will be too large, by a factor of approximately 1/.632). Indeed, in practice, it is likely that in-sample and future data are very "close" (i.e. similar) from each other. If we use S_l^* as an estimator of the predictive ability, we only consider the situations where the predicted points are far from their training set. At the contrary, if we only use S_l^0, we only consider the situation where the predicted points are also in the training set. Multiplying by .632 correct roughly for this bias (see Efron and Tibshirani [1997] for a more theoretical discussion on the subject). Nevertheless, when the considered rule overfits a lot the data, S_l^0 is artificially too low (still for a prediction error), driving $\bar{S}_l$ to be too low too. To offset this effect, Efron and Tibshirani [1997] proposed to use $w(\Lambda)$ instead of .368.

As explained in Section 2, Λ must be viewed as the expected error rate when predictors and predicted variables are independent, i.e. as if we were predicting using an independent random process. In the case of a binary classification problem, Efron and Tibshirani [1997] gives us the following expression for Λ:

$$\Lambda = p_1(1 - q_1) + (1 - p_1)q_1 \tag{30}$$

where p_1 is the observed proportion of the predicted variable taking value 1 and q_1 the proportion of the predictions taking value 1. This is similar to computing the expected value of the score function (here, a misclassification rate) when we predict with a Bernoulli process of parameter q_1, the particular realization of another binomial process of parameter p_1. Transposing this idea to our problem, Λ is an estimator of the expected mean excess return when the predictions (of future ups, downs or neutral market variations) are made with a three-dimension categorical random process with parameters q_1, q_2 and $1 - q_1 - q_2$ (n is the number of predictions on the initial data, q_1 is the proportion of predicted up and q_1 the prediction of predicted down, both in the initial sample). To compute this quantity, we generate randomly 5000 series of $n = 2933$ trading positions (1 if we are long, -1 if we are short, 0 if we are out of the market), that follow this categorical process. With these positions at hands, we can compute a mean excess return for each scenario, and take the average.

	IBM			MSFT			DJIA		
Trading rule	Λ	w_{res}	w_{block}	Λ	w_{res}	w_{block}	Λ	w_{res}	w_{block}
Filter rule simple	$4.61e^{-4}$	1	1	$3.55e^{-4}$	0.632	0.632	$6.706e^{-5}$	1	1
Filter rule neutral	$4.04e^{-4}$	1	1	$2.13e^{-4}$	1	1	$6.59e^{-5}$	1	1
Filter rule constant	$4.63e^{-4}$	1	1	$4.76e^{-5}$	1	1	$5.336e^{-5}$	1	1
Filter rule modified	$9.9e^{-5}$	1	1	$3.1e^{-4}$	1	1	$5.852e^{-5}$	1	1
MA time delay	$8.71e^{-5}$	1	1	$5.28e^{-5}$	1	1	$2.275e^{-5}$	1	1
MA filter band	$9.5e^{-5}$	1	1	$2.59e^{-5}$	1	1	$3.058e^{-5}$	1	1
S&R time delay	$2.7e^{-6}$	1	1	$1.47e^{-5}$	1	1	$2.279e^{-5}$	1	1
S&R filter band	$1.54e^{-5}$	1	1	$-1.51e^{-4}$	1	1	$6.5e^{-6}$	1	1
S&R modified time delay	$3.6e^{-6}$	0.97	1	$-7.9e^{-6}$	1	1	$2.214e^{-5}$	1	1
S&R modified filter band	$2.11e^{-5}$	1	1	$-2.03e^{-4}$	1	1	$4.288e^{-5}$	1	1
Channel breakout	$-2.1e^{-4}$	1	1	$-1.74e^{-4}$	1	1	$-6.88e^{-6}$	1	1

Table D.1: Intermediate parameters for the computation of $\bar{S}_l^+$, for the the three considered time series.

Values of J_{res} (for residuals-based bootstrap) and J_{block} (for block bootstrap) are obtained with equation (17). When the computed value is above 1, we simply take J equal to 1: $J' = \min\{J, 1\}$. Indeed, it indicates that the difference between in-sample and out-of-sample performance is higher than the performance based on a random strategy, suggesting that the overfitting is pretty high. At the contrary, if J is lower than 0, we take $J' = 0$. This is also what is proposed in Efron and Tibshirani [1997] to ensure that J is between 0 et 1. Value of w_{res} and w_{block} for the Sharpe ratio are identical. They can be obtained upon demand to the authors (they have been omitted for space considerations).

Below, we present the results obtained with the block bootstrap procedure. These

results are similar to the ones obtained with the mean excess return criterion. These results suggest that the supposed models in the residuals-based resampling procedure are correct.

Trading rule	MER	$S_l^{0,*}$	S_l^*	$\bar{S}_l$	$\bar{S}_l^+$	SR	$S_l^{0,*}$	S_l^*	$\bar{S}_l$	$\bar{S}_l^+$
Filter rule simple		$5.610e^{-4}$	$1.649e^{-4}$	$3.164e^{-4}$	$1.649e^{-4}$		0.0395	0.012	0.022	0.012
Filter rule neutral		$4.253e^{-4}$	$-0.399e^{-4}$	$1.240e^{-4}$	$-0.399e^{-4}$		0.037	−0.008	0.006	−0.008
Filter rule constant		$7.129e^{-4}$	$0.670e^{-4}$	$2.495e^{-4}$	$0.670e^{-4}$		0.050	0.005	0.018	0.005
Filter rule modified		$7.123e^{-4}$	$0.446e^{-4}$	$2.557e^{-4}$	$0.446e^{-4}$		0.050	0.003	0.018	0.003
MA filter band		$5.734e^{-4}$	$-1.380e^{-4}$	$0.757e^{-4}$	$-1.380e^{-4}$		0.040	−0.010	0.006	−0.010
MA time delay		$5.089e^{-4}$	$-1.141e^{-4}$	$0.883e^{-4}$	$-1.141e^{-4}$		0.036	−0.008	0.007	−0.008
S&R time delay		$4.637e^{-4}$	$-0.554e^{-4}$	$0.461e^{-4}$	$-0.554e^{-4}$		0.037	−0.004	0.003	−0.004
S&R filter band		$3.632e^{-4}$	$-1.299e^{-4}$	$0.928e^{-4}$	$-1.299e^{-4}$		0.030	−0.012	0.006	−0.012
S&R modified time delay		$4.436e^{-4}$	$-0.253e^{-4}$	$0.769e^{-4}$	$-0.253e^{-4}$		0.035	−0.003	0.005	−0.003
S&R modified filter band		$4.637e^{-4}$	$-0.554e^{-4}$	$1.461e^{-4}$	$-0.554e^{-4}$		0.037	−0.007	0.009	−0.007
Channel breakout		$0.889e^{-4}$	$-3.119e^{-4}$	$-2.037e^{-4}$	$-3.119e^{-4}$		0.007	−0.025	−0.016	−0.025

Table D.2: Values of the mean excess return (MER) and the Sharpe ratio (SR), for the IBM daily time series, obtained using a block bootstrap method. Resampled interest rates are correlated with the bootstrap returns. The average block length is 10.

Trading rule	MER	$S_l^{0,*}$	S_l^*	$\bar{S}_l$	$\bar{S}_l^+$	SR	$S_l^{0,*}$	S_l^*	$\bar{S}_l$	$\bar{S}_l^+$
Filter rule simple		$4.078e^{-4}$	$-1.600e^{-4}$	$-0.024e^{-4}$	$-0.024e^{-4}$		0.024	−0.009	0.000	0.000
Filter rule neutral		$2.891e^{-4}$	$-1.510e^{-4}$	$-0.281e^{-4}$	$-1.510e^{-4}$		0.025	−0.011	−0.003	−0.011
Filter rule constant		$6.961e^{-4}$	$-0.655e^{-4}$	$1.958e^{-4}$	$-0.655e^{-4}$		0.041	−0.004	0.012	−0.004
Filter rule modified		$6.466e^{-4}$	$-1.456e^{-4}$	$0.903e^{-4}$	$-1.456e^{-4}$		0.038	−0.008	0.005	−0.008
MA filter band		$5.976e^{-4}$	$-2.181e^{-4}$	$0.276e^{-4}$	$-2.181e^{-4}$		0.035	−0.013	0.001	−0.013
MA time delay		$5.241e^{-4}$	$-1.645e^{-4}$	$0.691e^{-4}$	$-1.645e^{-4}$		0.031	−0.010	0.004	−0.010
S&R time delay		$4.600e^{-4}$	$-1.028e^{-4}$	$0.601e^{-4}$	$-1.028e^{-4}$		0.030	−0.006	0.004	−0.006
S&R filter band		$3.782e^{-4}$	$-2.074e^{-4}$	$-0.374e^{-4}$	$-2.074e^{-4}$		0.024	−0.013	−0.002	−0.013
S&R modified time delay		$4.252e^{-4}$	$-0.982e^{-4}$	$0.648e^{-4}$	$-0.982e^{-4}$		0.027	−0.007	0.003	−0.007
S&R modified filter band		$4.600e^{-4}$	$-1.028e^{-4}$	$0.490e^{-4}$	$-0.502e^{-4}$		0.030	−0.013	−0.001	−0.009
Channel breakout		$1.589e^{-4}$	$-2.826e^{-4}$	$-0.707e^{-4}$	$-2.826e^{-4}$		0.010	−0.018	−0.005	−0.018

Table D.3: Values of the mean excess return (MER) and Sharpe ratio (SR), for the MSFT daily time series, obtained using a block bootstrap method. Resampled interest rates are correlated with the bootstrap returns. The average block length is 10.

Trading rule	MER	$S_l^{0,*}$	S_l^*	$\bar{S}_l$	$\bar{S}_l^+$	SR	$S_l^{0,*}$	S_l^*	$\bar{S}_l$	$\bar{S}_l^+$
Filter rule simple		$1.197e^{-4}$	$-0.119e^{-4}$	$0.468e^{-4}$	$-0.119e^{-4}$		0.0252	−0.002	0.010	−0.002
Filter rule neutral		$0.953e^{-4}$	$-0.235e^{-4}$	$0.416e^{-4}$	$-0.235e^{-4}$		0.028	−0.010	0.005	−0.010
Filter rule constant		$1.594e^{-4}$	$-0.158e^{-4}$	$0.470e^{-4}$	$-0.158e^{-4}$		0.032	−0.003	0.009	−0.003
Filter rule modified		$1.578e^{-4}$	$-0.252e^{-4}$	$0.482e^{-4}$	$-0.252e^{-4}$		0.032	−0.004	0.012	−0.004
MA filter band		$1.432e^{-4}$	$-0.602e^{-4}$	$0.058e^{-4}$	$-0.602e^{-4}$		0.028	−0.012	0.001	−0.012
MA time delay		$1.331e^{-4}$	$-0.571e^{-4}$	$0.086e^{-4}$	$-0.571e^{-4}$		0.026	−0.011	0.002	−0.011
S&R time delay		$1.260e^{-4}$	$-0.645e^{-4}$	$0.021e^{-4}$	$-0.645e^{-4}$		0.025	−0.014	0.000	−0.014
S&R filter band		$0.712e^{-4}$	$-0.577e^{-4}$	$-0.067e^{-4}$	$-0.577e^{-4}$		0.017	−0.013	−0.002	−0.013
S&R modified time delay		$1.211e^{-4}$	$-0.640e^{-4}$	$0.011e^{-4}$	$-0.640e^{-4}$		0.024	−0.013	0.000	−0.013
S&R modified filter band		$1.260e^{-4}$	$-0.645e^{-4}$	$-0.263e^{-4}$	$-0.645e^{-4}$		0.025	−0.014	−0.006	−0.014
Channel breakout		$0.323e^{-4}$	$-0.611e^{-4}$	$-0.219e^{-4}$	$-0.611e^{-4}$		0.010	−0.015	−0.004	−0.015

Table D.4: Values of the mean excess return (MER) and Sharpe ratio (SR), for the DJIA daily time series, obtained using a block bootstrap method. Resampled interest rates are correlated with the bootstrap returns. The average block length is 10.