Journal of Business Economics and Management
ISSN 1611-1699 / eISSN 2029-4433
Article in press
https://doi.org/10.3846/jbem.2021.15650

THE SEARCH FOR TIME-SERIES PREDICTABILITY-BASED ANOMALIES

Javier Humberto OSPINA-HOLGUÍN*, Ana Milena PADILLA-OSPINA

Faculty of Administrative Sciences, Universidad del Valle, Cali, Colombia

Received 09 April 2020; accepted 25 May 2021; first published online 29 November 2021

Abstract. This paper introduces a new algorithm for exploiting time-series predictability-based patterns to obtain an abnormal return, or alpha, with respect to a given benchmark asset pricing model. The algorithm proposes a deterministic daily market timing strategy that decides between being fully invested in a risky asset or in a risk-free asset, with the trading rule represented by a parametric perceptron. The optimal parameters are sought in-sample via differential evolution to directly maximize the alpha. Successively using two modern asset pricing models and two different portfolio weighting schemes, the algorithm was able to discover an undocumented anomaly in the United States stock market cross-section, both out-of-sample and using small transaction costs. The new algorithm represents a simple and flexible alternative to technical analysis and forecast-based trading rules, neither of which necessarily maximizes the alpha. This new algorithm was inspired by recent insights into representing reinforcement learning as evolutionary computation.

Keywords: stock market, investment algorithm, trading rules, alpha maximization, market timing, artificial intelligence, machine learning, reinforcement learning, evolutionary computation, perceptron.

JEL Classification: G11, G17.

Online supplementary material: Supporting information for this paper is available as online supplementary material at https://doi.org/10.3846/jbem.2021.15650

Introduction

This work introduces an algorithm designed to detect and profitably exploit the presence of time-series predictability-based anomalies. A market anomaly is a reliable and predictable pattern in the time-series or cross-section of asset returns that cannot be explained by a benchmark asset-pricing market model (Keim, 2008). An anomaly is typically demonstrated by rejecting the joint null hypothesis that the market is efficient and that asset returns behave according to a given benchmark asset-pricing model (Keim, 2008). In this context, the asset-

*Corresponding author. E-mail: *javier.ospina@correounivalle.edu.co* pricing model attempts to predict the "normal" asset returns based on the asset's exposure to a given set of risk factors.

When using the modern Fama and French (2015) asset-pricing model as a benchmark, the algorithm finds "abnormal returns" – or alphas – as high as 12% annually. Specifically, the proposed algorithm postulates a market timing strategy that each day stipulates, using past returns as input data, whether an investor is fully invested in a risky asset or in the standard risk-free asset (the 30-day T-bill). As a risky asset, each of the ten equal-weighted size deciles U.S. stock market portfolios representing the market cross section are used. Abnormal returns indicating an anomaly are found robustly in eight of the ten portfolios. Results are similar when using value-weighted size-decile portfolios. Furthermore, employing the Carhart (1997) asset-pricing model, which accounts for momentum, as a benchmark in a robustness check produces even higher abnormal returns, whether equal-weighted or value-weighted portfolios are used. Generally, alphas computed using both asset-pricing models and either weighting type tend to decline as firm size increases, disappearing for the largest company's size-decile portfolio.

This paper advances five original contributions. First, it develops a new algorithm that can automatically supply positive abnormal returns exploiting time-series predictability. Although many algorithms attempt to maximize profitability measures, such as total cumulative return or Sharpe ratio, a thorough literature search revealed only one work involving the development of an algorithm that can maximize alphas automatically: Brogaard and Zareei (2018). In comparison to that paper's, this study's algorithm is more flexible and considerably faster.

Second, this work's proposed algorithm is based on neither conventional technical analysis nor asset prices (or returns) forecasts. While conventional technical analysis appeals to various indicators to determine each buy and sell signal,[1] using such indicators is not necessarily scientifically justified and is instead typically based on traditional (folkloric) or otherwise subjective thinking. Forecasts, on the other hand, are most commonly built according to forecast error minimization. Such procedures do not necessarily maximize a risk-adjusted measure of profitability in the manner of alphas. This is especially evident when including transaction costs: even a perfect forecast may induce so many cumulative transaction costs that its insight into trading decisions may be suboptimal, or even useless, in terms of maximizing risk-adjusted profitability.

Instead, the proposed algorithm more closely follows reinforcement-learning literature: in its construction, the algorithm looks for an alpha maximizing trading rule, in which the rule is encoded in a perceptron function (cf. Gold, 2003). Beginning with the groundbreaking work by Allen and Karjalainen (1999), genetic algorithms and genetic programming have been widely used to construct trading rules (cf. Liu et al., 2020), including in hedge-fund and expensive-commercial-software contexts.[2] These trading rules are often represented using

[1] The terms "conventional technical analysis" and "technical analysis tradition" are used in this work to denote the standard methods of technical analysis. These methods are essentially identified in, for example, Murphy (1999), Plummer (2010), Kirkpatrick and Dahlquist (2011), Edwards, Magee, and Bassetti (2007), and Pring (2002), or by technical analysis certifications such as those awarded by the Chartered Market Technician® Program.

[2] See, for example, the Proteom Fund Ltd. (http://jonathankinlay.com/2018/09/developing-trading-strategies-with-genetic-programming) or Trading System Lab software, which featured an initial upfront cost of $60,000 (http://www.tradingsystemlab.com).

basic mathematical operations and logical connectors. Significantly, the present work implies the novel proposition of instead representing such trading rules as neural networks, which are both considerably simpler to program than genetic programming rules and substantially more general, given the approximation properties of neural networks (cf. Kidger & Lyons, 2020). Importantly, in this context, the trading rules are the neural networks themselves (the policies), departing from more common neural-networks-as-forecasting-tools applications. This new approach was inspired by the work of Salimans, Ho, Chen, Sidor, and Sutskever (2017), which interpreted reinforcement learning as a general evolutionary algorithm form, broadly simplifying reinforcement learning programming. Such simplification may ultimately enable advances not only in other financial fields (including risk management, portfolio allocation, and market microstructure) but also in related economics fields (including stochastic games, real-time bidding, consumption and income dynamics, and adaptive experimental design) (cf. Charpentier et al., 2021).[3]

Third, the proposed algorithm incorporates a market timing strategy that has already considered trading costs when it arrives at determining its trading signals – when to buy or sell the risky asset – meaning that the algorithm's alpha is already optimized for transaction costs, which contrasts with the usual development of trading rules (e.g., development based on forecasts or technical analysis), where rules might no longer be optimal after introducing transaction costs.

Fourth, this study's approach reveals a previously unreported anomaly in the U.S. stock market cross-section. The time-series predictability-based anomaly is robust to changes in the benchmark asset-pricing model, from the Fama and French (2015) to the Carhart model (1997), and portfolio construction, from equal-weighted to value-weighted. The first result is important because it demonstrates that the algorithm successfully exploits predictability patterns in the data, independent of the benchmark used. The second result is important because investors can often more easily reproduce value-weighted portfolios than equal-weighted portfolios, with the former approach often requiring less rebalancing.

Fifth, the anomaly documented in this work was identified using exclusively out-of-sample data. Searching for anomalies is frequently a trial-and-error procedure, which can produce false results given data-mining bias. All of the abnormal results this work reports were obtained using data not seen by the algorithm during its development. This practice alleviates any overfitting and data-mining concerns and strongly reinforces the conclusion that the anomaly found is real.

The rest of the paper is organized as follows: Section 1 provides a general overview of the literature associated with investment algorithms; Section 2 describes the algorithm for optimizing alpha directly and introduces the data used to test its efficacy; Section 3 provides the descriptive statistics of the (out-of-sample) data and the performance of the algorithm in terms of alphas for two types of size decile portfolios using two modern asset pricing models; the final section provides concluding remarks.

[3] An anonymous reviewer suggested emphasizing the potential connection to price formation and transmission mechanisms (e.g., Gričar & Bojnec, 2019; Hassouneh et al., 2015, 2017).

1. Literature review

The literature on investment algorithms (specifically, market-timing strategies based on past returns) is broad. It can be divided into four main branches: investment algorithms based on forecasts, investment algorithms based on conventional technical analysis, and investment algorithms based on reinforcement learning (or control theory in general, as in the case of dynamic programming), which includes the fourth branch, algorithms based on policy optimization.

Algorithms based on forecasts typically try to invest according to the best available forecast for the next day. Forecasts are based on different methodologies, from those based on conventional methodologies such as autoregressive moving averages models (Atsalakis & Valavanis, 2013), to those based on soft computing, such as neural networks or support-vector machines (Atsalakis & Valavanis, 2009; Henrique et al., 2019). The goal of such algorithms is to produce an optimal forecast and then to follow a simple investment strategy based on that forecast. An example of an investment strategy could be investing in a risky asset when the forecast for the risky asset's return is positive and investing in the riskless asset when the forecast for the risky asset's return is negative (Chan, 2017). Variants include investing in the risky asset only when a certain threshold for the risky asset's return is surpassed. A forecast is considered optimal or the best available because it minimizes an error measurement, such as the root-mean-squared error. However, an optimal forecast does not necessarily optimize risk-adjusted measures of total return, such as alpha, which the proposed algorithm does.

Algorithms based on conventional technical analysis, on the other hand, rely on investment indicators and rules often resembling folklore (Park & Irwin, 2007). While many traders follow these rules (cf. Lo & Hasanhodzic, 2010; Menkhoff, 2010), such rules do not generally have a substantial or scientific basis (Malkiel, 2007). Instead, they are heavily based on anecdotal experience or traditional beliefs. Although the empirical evidence in favor of technical analysis is controversial (Park & Irwin, 2007), some modern authors have tried to substantiate it (Han et al., 2013, 2016; Lo et al., 2000). Given the dubious basis, this paper distances itself from conventional technical analysis.

Algorithms based on reinforcement learning, or control theory, attempt to optimize a measurement of (risk-adjusted) total return (v.g., Bertoluzzo & Corazza, 2014; Cong et al., 2021; Kolm & Ritter, 2021; Mosavi et al., 2020; Pendharkar & Cusatis, 2018; Xufre Casqueiro & Rodrigues, 2006; Zhang et al., 2020). The investment process is seen in these kinds of algorithms as the generalization of a Markov decision-making process: an agent in a given state has to act in the market, (e.g. invest in a risky or riskless asset), and the market returns a reward and a new state of being for the agent (Charpentier et al., 2021; Fischer, 2018). Specialized deep-reinforcement-learning techniques have recently begun to be used.

Algorithms based on policy optimization can be seen as a variant of reinforcement learning wherein the policy is optimized directly. Most commonly, policy optimization does not require computing the policy encoded in the V or Q function (Sutton & Barto, 2018), but a parametrized policy that generates a whole sequence of actions and states can be optimized directly based on the total (possibly risk-adjusted) reward it generates for a whole episode. (A parametrized policy here is an investment rule described by a finite number of parameters that indicates how to invest given a state.) Often, evolutionary algorithms are used to perform this optimization. An example of this kind of algorithm is the work of Brogaard and Zareei (2018). Like the current work, those researchers attempted to optimize alpha, but using genetic programming to find optimal-technical-analysis rules instead of, as this paper does, attempting to use a perceptron function of past returns to find the optimal policy directly. (Sometimes the other types of algorithms are presumed to be "augmented" by technical indicators, such as in Brogaard and Zareei (2018).)

Reinforcement-learning literature is the newest and most regularly effective of the four different branches. Direct policy optimization is an innovation in this context. This work fills a gap in the literature by describing, for the first time, how alphas can be optimized directly in parametrized form without resorting to the traditional approach dictated by technical analysis and by introducing a more easily implemented and considerably faster new algorithm.

2. Data and methodology

To illustrate the identification of an anomaly based on predictability, the proposed algorithm is applied to the daily returns of the $(j=1,\ldots,10)$ size-decile portfolios of the United States market, with j ranging from the decile with the smallest firms to the decile with the largest firms. The decile portfolios were constructed based on sorting NYSE, AMEX, and NASDAQ stocks into ten groups (deciles) according to market equity (size). Upon assigning stocks to portfolios, daily returns (including dividends) were calculated using equal weighting or value weighting. The portfolios were rebalanced at the end of each June using June's market equity and NYSE breakpoints. The portfolios for the period between July 1, 1963, and April 30, 2019, were obtained from Kenneth French's website[4], with returns measured in terms of percentage (computed using current dollar prices). Half the sample was used for training (7026 data points), and half the sample was used for testing the algorithm out-of-sample.

Let us begin by expressing the market timing strategy in terms of the linear parametric functional form that guides the investment process. In a market timing strategy, the investment rule used is of one particular form: it indicates when to leave or when to enter the market, that is, when to hold the risky or the riskless asset. In other words, the market timing strategy's naked return R_{jt}^{M} at time t according to the investment rule on the underlying j-th portfolio is:

$$R_{jt}^{M}=\begin{cases} R_{jt}, & \text{if} \quad \mathrm{Pos}_{t-1}=1 \quad \text{and} \quad \mathrm{Pos}_{t-2}=1 \\ R_{jt}-\tau, & \text{if} \quad \mathrm{Pos}_{t-1}=1 \quad \text{and} \quad \mathrm{Pos}_{t-2}=0 \\ r_{ft}, & \text{if} \quad \mathrm{Pos}_{t-1}=0 \quad \text{and} \quad \mathrm{Pos}_{t-2}=0 \\ r_{ft}-\tau, & \text{if} \quad \mathrm{Pos}_{t-1}=0 \quad \text{and} \quad \mathrm{Pos}_{t-2}=1 \end{cases}, \tag{1}$$

where Pos_{t-1} is the position the investment strategy set the previous day (i.e., at $t-1$), Pos_{t-2} is the position set at $t-2$, R_{jt} is the daily return of the j-th risky portfolio at time t, r_{ft} is the risk-free rate at time t, and τ is the transaction cost per trade. The position

[4] https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

Pos_{t-1} (or Pos_{t-2}) takes the value of 1 if the investment rule dictates buying or holding the risky portfolio, obtaining return R_{jt} for the portfolio, and it takes the value of 0 if the investment rule is invested off-the-market in the risk-free asset. The 30-day T-bill is used as a risk-free asset, assuming that a transaction cost is charged for trading the risky portfolio but that there are no costs for trading the 30-day T-bill (e.g., Balduzzi & Lynch, 1999; Han et al., 2013; Lynch & Balduzzi, 2000).

The measure of profitability or performance of the strategy is the Fama and French (2015) alpha, although another asset pricing model is used as a robustness check. The return used to measure the alpha is based on the zero-cost arbitrage portfolio, which is long in the market timing portfolio resulting from applying the investment rule and short in the underlying portfolio. That is, the excess return of the investment rule IR_{jt} is computed relative to the buy and hold positions on the risky portfolio; therefore, it is defined as:

$$\mathrm{IR}_{jt} \equiv R_{jt}^{M} - R_{jt}\,. \tag{2}$$

According to the above, the portfolio performance measure IR_{jt} is the alpha α_j from the time series regression of the five-factor Fama and French model (2015). That is, the alpha from the regression

$$\begin{aligned}\mathrm{IR}_{jt} = \alpha_j + \beta_{j,\mathrm{MKT}} r_{\mathrm{MKT},t} + \beta_{j,\mathrm{SMB}} r_{\mathrm{SMB},t} + \beta_{j,\mathrm{HML}} r_{\mathrm{HML},t} +\\ \beta_{j,\mathrm{RMW}} r_{\mathrm{RMW},t} + \beta_{j,\mathrm{CMA}} r_{\mathrm{CMA},t} + \varepsilon_{jt},\end{aligned} \tag{3}$$

where $r_{\mathrm{MKT},t}$ is the daily return in excess of the market, $r_{\mathrm{SMB},t}$ is the daily return of the small-minus-big (SMB) factor related to size, $r_{\mathrm{HML},t}$ is the daily return of the high-minus-low (HML) factor related to growth, $r_{\mathrm{RMW},t}$ is the daily return of the robust-minus-weak (RMW) factor related to operating profitability, and $r_{\mathrm{CMA},t}$ is the daily return of the conservative-minus-aggressive (CMA) factor related to investment aggressiveness. The Fama and French (2015) daily factors and the daily risk-free rate are taken from Kenneth French's website. See Fama and French (2015) for a complete description of the factor returns. The risk-free rate corresponds to the one-month Treasury bill rate (from Ibbotson Associates).

Before continuing, it is important to stress the main implication of the recorded alpha being measured using an algorithmic arbitrage market timing investment (long in the algorithm, short in the underlying decile portfolio). The implication is that, by the definition in Eq. (2), this work focuses on the degree to which the market timing strategy outperforms the size-decile portfolio after discounting the particular size-decile portfolio performance. Thus, if the market timing strategy naively advised to always buy and hold the size decile portfolio, then the daily returns of the algorithmic arbitrage market timing investment and its alpha would be zero every time (by Eq. (2)), as would the final cumulative return.

The proposal of the new investment strategy is to increase the likelihood of finding a market timing algorithm that automatically generates a high risk-adjusted return in terms of the alpha. On the one hand, rules from the technical analysis tradition are not used because their scientific status is highly doubtful. On the other hand, prediction rules ultimately only minimize the sum of the distances between the predicted values and the respective real values, thus not necessarily promoting maximum profitability. Instead, a market timing strategy that could succeed by the nature of its construction is sought after; that is, a strategy that could automatically maximize alpha.

Based on the previous framework, the equations that specify the position of a general investment strategy in terms of a parametric functional form f can be written as follows:

$$\mathrm{Pos}_{t-1} = H\left(f(\mathbf{a},\mathbf{R}^{*}_{t-1})\right) \text{ and } \mathrm{Pos}_{t-2} = H\left(f(\mathbf{a},\mathbf{R}^{*}_{t-2})\right), \tag{4}$$

where H is the Heaviside function

$$H(x) = \begin{cases} 0, \, x < 0 \\ 1, \, x \geq 0 \end{cases}, \tag{5}$$

f is a general parametric function, $\mathbf{a} = \left(a_0 \quad a_1 \quad \dots \quad a_I\right)$ is a vector of parameters to be specified, $\mathbf{R}^{*}_{t-1} = \left(R_{jt-1}, R_{jt-2}, \dots, R_{jt-I}\right)$ is a vector with the information of the past returns from $t-I$ until $t-1$, and $\mathbf{R}^{*}_{t-2} = \left(R_{jt-2}, R_{jt-3}, \dots, R_{jt-I-1}\right)$ is a vector with the information of the past returns from $t-I-1$ until $t-2$. When $f(\mathbf{a},\mathbf{R}^{*}_{t-1})$ and $f(\mathbf{a},\mathbf{R}^{*}_{t-2})$ are known, they use the information available on past returns to establish when to buy or hold the risky portfolio, and when to sell it and hold the risk-free asset instead. Notice how, unlike standard predictive investment strategies, neither $f(\mathbf{a},\mathbf{R}^{*}_{t-1})$ nor $f(\mathbf{a},\mathbf{R}^{*}_{t-2})$ is a prediction in itself but rather a rule about the investment position to be taken (just as a moving average rule establishes this position without making any prediction about what is the future return).

To make the problem more manageable, several simplifying assumptions can be made. For example, a linear function f could be used, one of the simplest possible cases, i.e., a function $f(\mathbf{a},\mathbf{R}^{*}_{t-k}) = a_0 + \sum_{i=k}^{I+(k-1)} a_i R_{jt-i}$. This form, used in (4), recalls the perceptron function employed in reinforcement learning for trading by Gold (2003). Specifically, the present work uses $I = 10$ past returns for the optimal control investment rule. Following this section, this simple parametric rule's ability to cover a broad spectrum of possible investment strategies will be appreciated.

In this framework, to build an optimal investment rule, values of the vector $\mathbf{a}$ parameters such that $f(\mathbf{a},\mathbf{R}^{*}_{t-1})$ and $f(\mathbf{a},\mathbf{R}^{*}_{t-2})$ maximize alpha are needed. While this framework enables optimization of other performance measures, including cumulative returns and the Sharpe ratio, the present study is concerned with the Fama and French alpha. To optimize the Fama and French (2015) alpha specifically, it is possible to search the vector $\mathbf{a}$ of parameters which chooses the investment positions Pos_{t-1} and Pos_{t-2} between $t-2=1$ and $t-2=T_{\text{training}}-2$ in the training data such that the alpha α_j is maximum in the time series regression from the Fama and French model, i.e., the regression in (3). Since from (2), (1) and (4), consecutively:

$$\mathrm{IR}_{jt}=R_{jt}^{M}-R_{jt}=\begin{cases} R_{jt}-R_{jt}, & \text{if}\,\mathrm{Pos}_{t-1}=1 \text{ and } \mathrm{Pos}_{t-2}=1\\ R_{jt}-R_{jt}-\tau, & \text{if}\,\mathrm{Pos}_{t-1}=1 \text{ and } \mathrm{Pos}_{t-2}=0\\ r_{ft}-R_{jt}, & \text{if}\,\mathrm{Pos}_{t-1}=0 \text{ and } \mathrm{Pos}_{t-2}=0\\ r_{ft}-R_{jt}-\tau, & \text{if}\,\mathrm{Pos}_{t-1}=0 \text{ and } \mathrm{Pos}_{t-2}=1 \end{cases}$$
$$=\begin{cases} 0, & \text{if}\, f(\mathbf{a},\mathbf{R}_{t-1}^{*})>0 \text{ and } f(\mathbf{a},\mathbf{R}_{t-2}^{*})>0\\ -\tau, & \text{if}\, f(\mathbf{a},\mathbf{R}_{t-1}^{*})>0 \text{ and } f(\mathbf{a},\mathbf{R}_{t-2}^{*})\le 0\\ r_{ft}-R_{jt}, & \text{if}\, f(\mathbf{a},\mathbf{R}_{t-1}^{*})\le 0 \text{ and } f(\mathbf{a},\mathbf{R}_{t-2}^{*})\le 0\\ r_{ft}-R_{jt}-\tau, & \text{if}\, f(\mathbf{a},\mathbf{R}_{t-1}^{*})\le 0 \text{ and } f(\mathbf{a},\mathbf{R}_{t-2}^{*})>0 \end{cases}, \quad (6)$$

the term IR_{jt} in (3) can be expressed in terms of the rules $f(\mathbf{a},\mathbf{R}_{t-1}^{*})$ and $f(\mathbf{a},\mathbf{R}_{t-2}^{*})$, and it is even possible to estimate the alpha in this regression by ordinary least squares as

$$\hat{\alpha}_j=\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{IR}_j)\right)_1, \quad (7)$$

that is, as the first element of the vector $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{IR}_j)$, where $\mathbf{X}$ is the matrix of regressors of (3), $\mathbf{IR}_j$ is the vector of regressands expressed in terms of the rules $f(\mathbf{a},\mathbf{R}_{t-1}^{*})$ and $f(\mathbf{a},\mathbf{R}_{t-2}^{*})$, and the operations $\mathbf{A}^T$ and $\mathbf{A}^{-1}$ denote the transpose and inverse of matrix $\mathbf{A}$, respectively. Consequently, the components of $\mathbf{a}$ can be chosen that maximize the estimated alpha $\hat{\alpha}_j=\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{IR}_j)\right)_1$ in order to determine the investment rule $f(\mathbf{a},\mathbf{R}_{t-1}^{*})$.

However, in practice, there are two major obstacles to achieving the desired maximization. The first one is that $\hat{\alpha}_j=\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{IR}_j)\right)_1$ is an enormous algebraic expression, specifically of $T_{\text{training}}-2$ addends, which may involve the optimization of an expression of thousands of terms. But the second obstacle is more serious. Each of these terms contains expressions in terms of Heaviside functions. These expressions are not only highly nonlinear but also do not change locally almost everywhere. They are also discontinuous and nondifferentiable, since the derivative of $H(x)$ is the Dirac delta function $\delta(x)$ which is a "generalized function" (i.e., a distribution) having the property of being zero everywhere, infinity in 0, and whose integral over the reals is 1.

To overcome the obstacles that the optimization involves, a heuristic method of optimization was used that, unlike classical optimization methods such as quasi-Newton methods or gradient descent, does not require that the function is continuous, changes locally or is even differentiable. The method chosen is inspired by the theory of evolution and is called differential evolution (DE) (Rocca et al., 2011; Storn & Price, 1997).

The method starts from an initial population of m input vectors $\{\mathbf{a}_1,\mathbf{a}_2,\ldots,\mathbf{a}_m\}$. This population is composed of vectors $\mathbf{a}$ randomly chosen as potential candidates to maximize the estimated alpha $\hat{\alpha}_j=\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{IR}_j)\right)_1$. As the number of components of each candidate vector is $I+1=11$, m is chosen much higher than $I+1$. Later, each element of this initial population "evolves" to generate a new candidate to solve the optimization problem by two consecutive mechanisms. First, by mutation from three vectors of the initial population $\mathbf{a}_d$, $\mathbf{a}_e$ and $\mathbf{a}_f$ one solution candidate is obtained as $\mathbf{a}'=\mathbf{a}_f+s(\mathbf{a}_d-\mathbf{a}_e)$, where s

is a scale factor less than one. Second, by crossing (interbreeding), a new candidate $\mathbf{a}''$ is obtained from candidate $\mathbf{a}'$ and another point $\mathbf{a}_g$ of the initial population. The new candidate takes the i -th coordinate of $\mathbf{a}'$ and replaces it with the i -th coordinate of $\mathbf{a}_g$ with probability ρ or leaves if unchanged with probability $1-\rho$. If $\hat{\alpha}_j(\mathbf{a}'') > \hat{\alpha}_j(\mathbf{a}_g)$, $\mathbf{a}''$ replaces $\mathbf{a}_g$ in the initial population. If $\hat{\alpha}_j(\mathbf{a}'') \le \hat{\alpha}_j(\mathbf{a}_g)$, $\mathbf{a}_g$ is not replaced. The end result is also a new population with m vectors.

The process is repeated iteratively, and the optimization stops when the difference between the optimal candidate of $\hat{\alpha}_j$ is within a numerically insignificant specified distance of the optimum in the previous population, as are the distances between the vectors corresponding to these optimum values, (i.e., when the optimum converges based on predetermined tolerances). Given that, for the particular type of problem addressed here, there is no single solution vector – instead, the solution is a region – the fulfilling tolerances used were relatively large (0.001 in each case).

The final result of the optimization is an optimization parameter vector $\mathbf{a}^\bullet$ which chooses the positions taken by the investment strategy:

$$\text{Pos}_{t-1} = \begin{cases} 1, \text{ if } f(\mathbf{a}^\bullet, \mathbf{R}^*_{t-1}) > 0 \\ 0, \text{ if } f(\mathbf{a}^\bullet, \mathbf{R}^*_{t-1}) \le 0 \end{cases} \text{ and } \text{Pos}_{t-2} = \begin{cases} 1, \text{ if } f(\mathbf{a}^\bullet, \mathbf{R}^*_{t-2}) > 0 \\ 0, \text{ if } f(\mathbf{a}^\bullet, \mathbf{R}^*_{t-2}) \le 0 \end{cases} \quad (8)$$

in a manner that optimizes the estimated Fama and French alpha in the training data (i.e., within the sample). These optimal parameters are also expected to work out-of-sample if there is no overfitting during the optimization.

Since the parametric form of the function chosen to establish the position is linear, i.e., $f(\mathbf{a}, \mathbf{R}^*_{t-1}) = a_0 + \sum_{i=1}^{I} a_i R_{jt-i}$ (and $f(\mathbf{a}, \mathbf{R}^*_{t-2}) = a_0 + \sum_{i=2}^{I+1} a_i R_{jt-i}$), and besides it depends on $I = 10$ past returns, one wonders how general this parametric form is. Although it would have been possible to choose a more general functional form, for example, a nonlinear function with more parameters, the greater the number of parameters used and the complexity of the formula, the greater the possibility of overfitting. In other words, the possibility of finding an investment rule that generates an optimal alpha in the training data but only a mediocre alpha in the test data (outside the sample). Indeed, the functional form used was chosen to be simple and have few parameters for this reason. Despite this, such a functional form turns out to be quite flexible, as is explained below.

First, it is trivial to see that any inequality of the form $g(\mathbf{a}^1, \mathbf{R}^*_{t-1}) > C$ or of the form $g(\mathbf{a}^1, \mathbf{R}^*_{t-1}) < C$, with $g(\mathbf{a}^1, \mathbf{R}^*_{t-1}) = a_0^1 + \sum_{i=1}^{I} a_i^1 R_{jt-i}$, is a special case of $f(\mathbf{a}, \mathbf{R}^*_{t-1}) > 0$. Similarly, any inequality of the form $g_1(\mathbf{a}^1, \mathbf{R}^*_{t-1}) > g_2(\mathbf{a}^2, \mathbf{R}^*_{t-1})$ or $g_1(\mathbf{a}^1, \mathbf{R}^*_{t-1}) < g_2(\mathbf{a}^2, \mathbf{R}^*_{t-1})$, with $g_k(\mathbf{a}^k, \mathbf{R}^*_{t-1}) = a_{0k} + \sum_{i=1}^{I} a_{ik} R_{jt-i}$, is also a special case of $f(\mathbf{a}, \mathbf{R}^*_{t-1}) > 0$. Accordingly, neither the direction of the inequality, nor the presence or absence of a constant, nor the fact that the right side of the inequality is zero limit the generality of this type of investment strategy.

Secondly, Zakamulin (2016) has shown that a wide variety of technical analysis indicators can be written in terms of a linear combination of returns. Among others, momentum indicators and general moving average indicators including simple moving averages, simple moving average crossings, and linear, exponential and reverse exponential moving averages. In line with this, all these indicators, if they maximize the estimated alpha, have been verified by optimizing the selected investment strategy. So, according to the above, the parametric form of the chosen function, although parsimonious, is also quite general.

Thirdly, the parametric form of the function used is nothing more than a perceptron; that is, an artificial neuron using the Heaviside function as the activation function. The learning here is achieved through differential evolution. As a linear algorithm, perceptrons are very efficient if the training set is linearly separable, and this choice of function can be seen as the first approximation of a solution to the problem of finding an optimal investment algorithm.

3. Results and discussion

Table 1 shows the descriptive statistics of the testing sample from May 22, 1991, to April 30, 2019, for equal-weighted (Panel A) and value-weighted (Panel B) size decile portfolios, respectively. Of the 20 portfolios, only the equal-weighted size decile portfolio with the smallest firms has a mean significantly different from zero $(p = 0.02)$. This portfolio is also the only one with a Sharpe ratio over one, a risk-adjusted measure that otherwise ranges from 0.47 to 0.57. Each portfolio is negatively skewed, except for the equal-weighted size-decile portfolio with the largest firms.

The algorithm was designed and tested using the Wolfram and Julia languages, as well as Microsoft Excel. The final results (presented in Table 2) were computed in the Wolfram and R languages on a 6th Generation Intel Core i7-6700HQ. Training and testing each investment rule for a given decile portfolio required approximately 20 seconds. Panel A of Table 2 shows the out-of-sample Fama and French (2015) alphas that the algorithm achieved for the ten equal-weighted size-decile portfolios, while Panel B shows the same for the value-weighted size decile portfolios. To demonstrate that abnormal returns can survive transaction costs, a transaction cost of 1 basis point per trade was used for calculation as per Eq. (1). Given this cost was recognized by Balduzzi and Lynch (1999) as the lower limit for transaction costs in the 1990s, it is highly probable that big institutional investors can now trade below that cost. (In any case, Appendix A in the Supplementary Material develops a sensibility analysis of alpha in terms of transaction costs for further transparency.) The results clearly show alphas of significant economic importance in almost every portfolio, except for those with the largest firms. This could be because the excess returns of the arbitrage portfolios are less volatile, and therefore less suitable for a market-timing strategy. The alphas found in the equal-weighted size-decile portfolios grew from 0.09% to 12.12% annually as firm size decreased (Table 2, Panel A), while they grew from −3.73% to 12.03% annually as firm size decreased for the value-weighted portfolio (Table 2, Panel B).

However, trends were not monotonic. Using equal-weighted size decile portfolios (Table 2, Panel A), the algorithm did not return significant alphas for size deciles 9 and 10, those featuring the largest firms. For size deciles 2–8, alphas ranged from 5.49% annually for decile 3 to 8.70% annually for decile 7. The alpha calculated increased over 12% annually for size decile 1, which featured the smallest firms. For the value-weighted size decile portfolios

(Table 2, Panel B), again, the trend was not monotonic. The alpha returned by the algorithm for size decile 9 was not significant, while the alpha for size decile 10 was negative. Between portfolio size decile 7 and 8, the alpha dropped from 7.40% annually for decile 7 to 4.06% annually for decile 8. Between portfolios with size decile 2 and 6, alphas ranged from 5.13% annually for size deciles 3 to 7.19% annually for size decile 6. Alphas grew to 12.03% annually for size decile 1 portfolio, which featured the smallest firms.

The betas for market and size factors were almost invariably negative for both the equal-weighted (Table 2, Panel A) and value-weighted portfolios (Table 2, Panel B), indicating that the investment strategy represented a useful hedge against those factors. The betas for the other factors were most commonly insignificant, indicating that exposure to those factors was irrelevant to increasing returns.

An automatic anomaly search with the Carhart model (1997) was conducted as a robustness check:

$$\mathrm{IR}_{jt} = \alpha_j + \beta_{j,\mathrm{MKT}} r_{\mathrm{MKT},t} + \beta_{j,\mathrm{SMB}} r_{\mathrm{SMB},t} + \beta_{j,\mathrm{HML}} r_{\mathrm{HML},t} + \beta_{j,\mathrm{MOM}} r_{\mathrm{MOM},t} + \varepsilon_{jt}, \tag{9}$$

where $r_{\mathrm{MKT},t}$ is the daily return in excess of the market, $r_{\mathrm{SMB},t}$ is the daily return of the small-minus-big (SMB) factor related to size, $r_{\mathrm{HML},t}$ is the daily return of the high-minus-low (HML) factor related to growth, and $r_{\mathrm{MOM},t}$ is the momentum factor. Panel C of Table 2 shows the out-of-sample results for equal-weighted size-decile portfolios, while Panel D shows the same for value-weighted size-decile portfolios under Carhart's model.

The results show that the algorithm is not only capable of reproducing the predictability anomaly using those factors, but is capable of obtaining even (economically) higher alphas for eight of the ten portfolios when considering equal-weighted size decile portfolios (Table 2, Panel C) and for ten of the ten portfolios when considering value-weighted size decile portfolios (Table 2, Panel D).

Table 1. Panel A. Summary statistics for the daily returns of equal-weighted size decile portfolios

Decile	Naked returns (in annual percentage):									
	R_j									
	Smallest	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	Largest
Mean	31.44**	14.91	14.60	13.00	13.57	13.72	13.46	13.25	13.09	11.73
St. Deviation	13.49	21.84	21.53	21.24	21.39	19.89	19.73	19.86	19.19	18.82
Skewness	–0.57	–0.10	–0.19	–0.14	–0.15	–0.14	–0.14	–0.05	–0.05	0.12
Sharpe Ratio	2.15	0.57	0.56	0.50	0.52	0.57	0.56	0.54	0.55	0.49
Observations	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027

Note: Summary statistics for the daily returns of equal-weighted size decile portfolios for the testing sample between May 22, 1991, and April 30, 2019. *, **, and *** indicate a significant mean at the 0.1, 0.05, and 0.01 levels, respectively. "St. Deviation" stands for "Standard Deviation".

Panel B. Summary statistics for the daily returns of value-weighted size decile portfolios

Decile	Naked returns (in annual percentage):									
	R_j									
	Smallest	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	Largest
Mean	11.88	12.75	13.23	12.10	12.82	12.83	12.99	12.98	12.46	10.64
St. Deviation	16.40	21.36	21.00	20.54	20.55	18.94	18.76	18.87	18.05	17.51
Skewness	−0.64	−0.23	−0.29	−0.23	−0.26	−0.26	−0.31	−0.22	−0.17	−0.03
Sharpe Ratio	0.57	0.48	0.51	0.47	0.50	0.55	0.56	0.56	0.55	0.47
Observations	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027

Note: Summary statistics for the daily returns of value-weighted size decile portfolios for the testing sample between May 22, 1991, and April 30, 2019. *, **, and *** indicate a significant mean at the 0.1, 0.05, and 0.01 levels, respectively. “St. Deviation” stands for “Standard Deviation”.

Table 2. Panel A. Fama and French (2015) algorithmic alphas for each equal-weighted size decile portfolio

Decile	Dependent variable:									
	IR_j									
	Smallest	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	Largest
α_j	*12.12****	*6.71****	*5.49***	*6.80****	*6.42****	*7.25****	*8.70****	*5.53****	2.04	0.09
	[0.00]	[0.00]	[0.02]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.28]	[0.93]
$\beta_{j,\mathrm{MKT}}$	−0.30***	−0.60***	−0.61***	−0.61***	−0.61***	−0.60***	−0.60***	−0.59***	−0.61***	−0.58***
	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]
$\beta_{j,\mathrm{SMB}}$	−0.25***	−0.55***	−0.50***	−0.43***	−0.34***	−0.22***	−0.15***	−0.06*	0.04	0.15***
	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.06]	[0.10]	[0.00]
$\beta_{j,\mathrm{HML}}$	−0.04	−0.16***	−0.08*	−0.06	−0.05	0.00	−0.04	−0.01	−0.02	0.00
	[0.11]	[0.00]	[0.06]	[0.17]	[0.25]	[0.93]	[0.34]	[0.84]	[0.69]	[0.99]
$\beta_{j,\mathrm{RMW}}$	0.10***	0.00	0.02	0.03	0.08**	0.05	0.05	0.11**	0.04	0.01
	[0.00]	[0.94]	[0.54]	[0.32]	[0.05]	[0.30]	[0.16]	[0.01]	[0.45]	[0.75]
$\beta_{j,\mathrm{CMA}}$	0.00	0.00	0.02	0.07	0.07	0.07	0.05	0.00	0.00	0.05
	[0.99]	[0.92]	[0.78]	[0.21]	[0.18]	[0.15]	[0.36]	[0.99]	[0.94]	[0.34]
Obs.	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027
Adj. R^2	0.41	0.53	0.54	0.55	0.54	0.55	0.54	0.53	0.55	0.55

Note: A Fama and French (2015) time series regression on the algorithmic arbitrage market timing investment – long in the algorithm and short in the underlying decile portfolio – was conducted for each equal-weighted size decile portfolio for the testing sample between May 22, 1991, and April 30, 2019. For each decile, the algorithm was trained on data from July 1, 1963, to June 6, 1991. A transaction cost of 1 bp was assumed for training and testing. The 30-day T-bill is used as the risk-free asset, and one of the ten NYSE/AMEX/NASDAQ equal-weighted market-cap size-decile portfolios is used as the risky asset. The betas correspond to each of the Fama and French (2015) factors: MKT represents the excess market return factor, SMB represents the small-minus-big factor related to size, HML represents the high-minus-low factor related to growth, RMW represents the robust-minus-weak factor related to operating profitability, and CMA represents the conservative-minus-aggressive factor related to investment aggressiveness. Alphas are annualized and presented as percentages. Bootstrap *p*-values based on 5,000 bootstrap replications are in brackets (see Appendix B in the Supplementary Materials for details). *, **, and *** indicate significance at the 0.1, 0.05, and 0.01 levels, respectively. Obs. and Adj. R^2 represent the number of observations and the adjusted coefficient of determination.

Panel B. Fama and French (2015) algorithmic alphas for each value-weighted size decile portfolio

Decile	Dependent variable:									
	IR_j									
	Smallest	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	Largest
α_j	*12.03****	*5.85***	*5.13***	*5.67****	*6.19****	*7.19****	*7.40****	*4.06***	1.60	-3.73^{**}
	[0.00]	[0.01]	[0.02]	[0.01]	[0.00]	[0.00]	[0.00]	[0.03]	[0.33]	[0.03]
$\beta_{j,MKT}$	-0.41^{***}	-0.60^{***}	-0.58^{***}	-0.59^{***}	-0.58^{***}	-0.58^{***}	-0.59^{***}	-0.60^{***}	-0.58^{***}	-0.52^{***}
	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]
$\beta_{j,SMB}$	-0.42^{***}	-0.57^{***}	-0.49^{***}	-0.44^{***}	-0.34^{***}	-0.23^{***}	-0.16^{***}	-0.08^{***}	−0.01	0.14^{***}
	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.63]	[0.00]
$\beta_{j,HML}$	-0.08^{**}	-0.09^{**}	−0.06	−0.04	−0.04	0.03	0.02	0.03	0.03	0.02
	[0.02]	[0.04]	[0.12]	[0.36]	[0.36]	[0.36]	[0.39]	[0.29]	[0.35]	[0.57]
$\beta_{j,RMW}$	0.10^{***}	0.01	0.04	0.02	0.04	0.06^{*}	0.05^{*}	0.06^{*}	−0.02	−0.04
	[0.01]	[0.68]	[0.15]	[0.47]	[0.18]	[0.10]	[0.09]	[0.09]	[0.47]	[0.16]
$\beta_{j,CMA}$	0.02	−0.06	0.02	0.03	0.03	0.01	−0.05	−0.06	-0.08^{**}	0.00
	[0.62]	[0.21]	[0.64]	[0.55]	[0.49]	[0.89]	[0.17]	[0.11]	[0.04]	[0.99]
Obs.	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027
Adj. R^2	0.49	0.54	0.54	0.55	0.53	0.56	0.55	0.55	0.55	0.52

Note: A Fama and French (2015) time series regression on the algorithmic arbitrage market timing investment – long in the algorithm and short in the underlying decile portfolio – was conducted for each value-weighted size decile portfolio for the testing sample between May 22, 1991, and April 30, 2019. For each decile, the algorithm was trained on data from July 1, 1963, to June 6, 1991. A transaction cost of 1 bp was assumed for training and testing. The 30-day T-bill is used as the risk-free asset, and one of the ten NYSE/AMEX/NASDAQ value-weighted market-cap size-decile portfolios is used as the risky asset. The betas correspond to each of the Fama and French (2015) factors: MKT represents the excess market return factor, SMB represents the small-minus-big factor related to size, HML represents the high-minus-low factor related to growth, RMW represents the robust-minus-weak factor related to operating profitability, and CMA represents the conservative-minus-aggressive factor related to investment aggressiveness. Alphas are annualized and presented as percentages. Bootstrap *p*-values based on 5,000 bootstrap replications are in brackets (see Appendix B in the Supplementary Materials for details). *, **, and *** indicate significance at the 0.1, 0.05, and 0.01 levels, respectively. Obs. and Adj. R^2 represent the number of observations and the adjusted coefficient of determination.

Panel C. Carhart (1997) algorithmic alphas for each equal-weighted size decile portfolio

Decile	Dependent variable:									
	IR_j									
	Smallest	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	Largest
α_j	*12.58****	*5.75***	*5.37***	*6.88****	*7.69****	*7.60****	*9.47****	*6.68****	2.51	0.46
	[0.00]	[0.01]	[0.02]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.17]	[0.77]
$\beta_{j,\mathrm{MKT}}$	−0.32***	−0.61***	−0.61***	−0.60***	−0.63***	−0.62***	−0.61***	−0.63***	−0.60***	−0.59***
	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]
$\beta_{j,\mathrm{SMB}}$	−0.26***	−0.55***	−0.50***	−0.41***	−0.34***	−0.25***	−0.16***	−0.09**	0.03	0.14***
	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.01]	[0.13]	[0.00]
$\beta_{j,\mathrm{HML}}$	−0.06***	−0.15***	−0.08***	−0.03	0.01	0.04	0.01	0.04	−0.01	0.06*
	[0.01]	[0.00]	[0.01]	[0.31]	[0.67]	[0.24]	[0.58]	[0.23]	[0.82]	[0.09]
$\beta_{j,\mathrm{MOM}}$	−0.03	0.01	0.03	0.02	0.04*	0.03	0.03*	0.05**	0.06***	0.05**
	[0.16]	[0.62]	[0.15]	[0.30]	[0.09]	[0.19]	[0.10]	[0.02]	[0.01]	[0.01]
Obs.	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027
Adj. R^2	0.40	0.53	0.54	0.53	0.55	0.55	0.55	0.55	0.55	0.55

Note: A Carhart (1997) time series regression on the algorithmic arbitrage market timing investment – long in the algorithm and short in the underlying decile portfolio – was conducted for each equal-weighted size decile portfolio for the testing sample between May 22, 1991, and April 30, 2019. For each decile, the algorithm was trained on data from July 1, 1963, to June 6, 1991. A transaction cost of 1 bp was assumed for training and testing. The 30-day T-bill is used as the risk-free asset, and one of the ten NYSE/AMEX/NASDAQ equal-weighted market-cap size-decile portfolios is used as the risky asset. The betas correspond to each of the Carhart (1997) factors: MKT represents the excess market return factor, SMB represents the small-minus-big factor related to size, HML represents the high-minus-low factor related to growth, and MOM represents the momentum factor. Alphas are annualized and presented as percentages. Bootstrap *p*-values based on 5,000 bootstrap replications are in brackets (see Appendix B in the Supplementary Materials for details). *, **, and *** indicate significance at the 0.1, 0.05, and 0.01 levels, respectively. Obs. and Adj. R^2 represent the number of observations and the adjusted coefficient of determination

Panel D. Carhart (1997) algorithmic alphas for each value-weighted size decile portfolio

Decile	Dependent variable:									
	IR_j									
	Smallest	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	Largest
α_j	*13.03****	*8.20****	*6.28****	*6.17****	*7.06****	*7.83****	*8.25****	*5.17***	2.12	−3.67**
	[0.00]	[0.00]	[0.01]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.18]	[0.02]
$\beta_{j,\mathrm{MKT}}$	−0.43***	−0.61***	−0.60***	−0.60***	−0.60***	−0.60***	−0.58***	−0.60***	−0.58***	−0.54***
	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]
$\beta_{j,\mathrm{SMB}}$	−0.42***	−0.57***	−0.51***	−0.42***	−0.35***	−0.24***	−0.17***	−0.09***	−0.01	0.14***
	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.00]	[0.79]	[0.00]
$\beta_{j,\mathrm{HML}}$	−0.09***	−0.15***	−0.07**	−0.06**	−0.04	0.01	−0.02	−0.02	−0.02	0.03
	[0.00]	[0.00]	[0.03]	[0.04]	[0.14]	[0.72]	[0.41]	[0.43]	[0.37]	[0.27]

End of Panel D

Decile	Dependent variable:									
	IR_j									
	Smallest	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	Largest
$\beta_{j,\text{MOM}}$	-0.07^{***}	-0.03^{*}	-0.04^{**}	-0.03^{*}	-0.05^{***}	-0.05^{***}	-0.05^{**}	-0.04^{**}	-0.03^{**}	-0.01
	[0.00]	[0.06]	[0.02]	[0.09]	[0.01]	[0.00]	[0.00]	[0.02]	[0.03]	[0.66]
Obs.	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027	7,027
Adj. R^2	0.48	0.55	0.55	0.54	0.53	0.55	0.54	0.55	0.55	0.54

Note: A Carhart (1997) time series regression on the algorithmic arbitrage market timing investment – long in the algorithm and short in the underlying decile portfolio – was conducted for each value-weighted size decile portfolio for the testing sample between May 22, 1991, and April 30, 2019. For each decile, the algorithm was trained on data from July 1, 1963, to June 6, 1991. A transaction cost of 1 bp was assumed for training and testing. The 30-day T-bill is used as the risk-free asset, and one of the ten NYSE/AMEX/NASDAQ value-weighted market-cap size-decile portfolios is used as the risky asset. The betas correspond to each of the Carhart (1997) factors: MKT represents the excess market return factor, SMB represents the small-minus-big factor related to size, HML represents the high-minus-low factor related to growth, and MOM represents the momentum factor. Alphas are annualized and presented as percentages. Bootstrap *p*-values based on 5,000 bootstrap replications are in brackets (see Appendix B in the Supplementary Materials for details). *, **, and *** indicate significance at the 0.1, 0.05, and 0.01 levels, respectively. Obs. and Adj. R^2 represent the number of observations and the adjusted coefficient of determination.

General Note: The alphas in italics are significantly positive (abnormal) in the respective asset pricing model.

As firm size decreased, the alphas found for the equal-weighted size decile portfolios grew from 0.46% to 12.58% annually (Table 2, Panel C); for value-weighted portfolios, they grew from −3.67% to 13.03% (Table 2, Panel D). For the equal-weighted size decile portfolios (Table 2, Panel C), the alpha calculated was insignificant for the two size decile portfolios featuring the largest firms. The alpha dropped monotonically between size decile portfolios 7 and 10, from 9.47% annually for 7 to 0.46% annually for 10. Between portfolios with size decile 2 and 6, the alpha ranged from 5.37% annually for size decile 3 to 7.69% annually for size decile 5. The alpha calculated reached over 12.5% annually for the portfolio featuring the smallest firms.

For the value-weighted portfolios (Table 2, Panel D), the alpha was negative or not significantly different from zero for the two portfolios with the largest firms. From size deciles 2 and 8, the algorithm calculated alphas ranged from 5.17% annually for size decile 8 to 8.20% annually for size decile 2. The algorithm calculated an alpha of over 13% annually for the portfolio with the smallest firms.

As with the Fama and French (2015) model, negative exposure (beta) to the market and size factors were found to be predominant, for both the equal-weighted (Table 2, Panel C) and value-weighted portfolios (Table 2, Panel D). This, in turn, suggests that the algorithm constitutes a valuable hedge against exposure to these factors. The algorithm is also a valuable hedge against exposure to the book-to-market HML factor in three of the equal-weighted portfolios (Table 2, Panel C) and four of the value-weighted portfolios (Table 2, Panel D). In the case of the value-weighted portfolios, the algorithm also returned negative momentum betas for nine of the ten portfolios (Table 2, Panel D). In turn, it returned positive momentum betas in five of the ten equal-weighted portfolios; and statistically indistinguishable from zero betas in the other five cases (Table 2, Panel C).

In summary, the algorithm not only delivers statistically and economically significant positive risk-adjusted returns (alphas) most of the time using both the Fama and French (2015) and Carhart (1997) models but is also a valuable hedge against market and size factors. In the case of Carhart's model, the algorithm also provides a valuable hedge against the momentum factor when value-weighted portfolios are considered.

In simpler terms, the algorithm – applied to different portfolios ordered by size – was able to generate an investment that greatly outperformed the buy and hold strategy out-of-sample, even after both transaction costs and adjusting returns according to the algorithm's exposure to some of the most widely employed risk factors. Furthermore, exposure to these risk factors was generally low or negative, implying that when risk due to these factors is high (for example, when the market is trending downward), the algorithm performs even better. It should be noted that these risk-adjusted returns were obtained directly from the algorithm's design and, thus, should be generalizable to other portfolios containing time-series patterns. Additionally, the algorithm is simpler and much faster than alternatives, such as those using genetic programming (cf. Brogaard & Zareei, 2018). Together, these results strengthen the notion that using this algorithm represents a competitive choice for building market-timing strategies.

Conclusions

This paper has introduced a novel algorithm for automatically discovering anomalies based on the time-series predictability of asset returns. The proposed algorithm delivers a market timing strategy that decides whether to invest in or continue holding a risky asset or invest in or continue holding a riskless asset instead. This decision is made every day using a parametric perceptron function of past returns to represent the trading rule. Then, the algorithm directly optimizes the trading rule parameters for a maximum alpha using in-sample differential evolution. In contrast to forecast-based trading rules, which minimize forecast error, or technical analysis trading rules, which often reflect traditional or subjective interpretations, the proposed algorithm can automatically accommodate any exploitable time-series patterns of returns for optimal risk-adjusted returns. Given its design, the algorithm can also incorporate transaction costs in the trading rule representation – also differing from conventional technical analysis or prediction-based strategies – and deliver optimal alphas in the presence of transaction costs.

To demonstrate its capabilities, the algorithm was applied to size-decile portfolios representing a cross-section of the U.S. stock market, identifying an unreported out-of-sample anomaly using two of the most popular modern asset-pricing models and using different weighting schemes to construct the portfolios. The trading strategies obtained reasonably good (even as high as 12% annually) risk-adjusted performance in terms of the alpha for almost all of the examined portfolios.

One of the algorithm's current limitations concerns the complexities of establishing a market inefficiency. When testing efficiency, it is well-known that the rejection of efficiency can be caused by the market being truly inefficient or by the wrong asset-pricing model having been used to define normal returns. That is, reported abnormal returns might be the product of true exploitable predictability or the use of an inappropriate set of risk factors. Fortunately, if a new set of risk factors is required, and if exploitable patterns remain, the algorithm is sufficiently flexible to identify abnormal returns using the new set.

A second limitation of the algorithm is its use of a perceptron instead of a more general neural network architecture. Given the attempt to build a minimal working example, using a perceptron facilitated avoiding overfitting. Nonetheless, we are currently working towards generalizing the algorithm to other architectures, which may ultimately produce even better results. Despite this limitation, the present study provides a new avenue for research by offering the possibility of an algorithm that can both calculate investment strategies that optimize the alpha and automatically search for out-of-sample anomalies.

A direct line of future research might involve generalizing the linear trading rule that determines purchase-and-sales orders [$f(\mathbf{a}^{\bullet},\mathbf{R}_{t-1}^{*})$] to even more general nonlinear alternatives to assess whether such alternatives tend towards overfitting. Furthermore, the input variables used to construct the trading strategies could easily be extended beyond previous returns to incorporate other market variables. The algorithm could also be used to optimize other risk-adjusted measures, such as the Sharpe ratio or even total returns. Additionally, the algorithm is not restricted to the data selected for this study. Instead, its design makes it generalizable to other portfolios featuring exploitable trading patterns.

Funding

This work was supported by Universidad del Valle under Grant "Convocatoria Interna para Presentación de Proyectos de Investigación y Creación Artística en las Ciencias, las Artes, las Humanidades, las Tecnologías y la Innovación – Convocatoria 120 – 2019".

Disclosure statement

The authors declare that they have no conflict of interest.

References

Allen, F., & Karjalainen, R. (1999). Using genetic algorithms to find technical trading rules. *Journal of Financial Economics*, *51*(2), 245–271. https://doi.org/10.1016/S0304-405X(98)00052-X

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques – Part II: Soft computing methods. *Expert Systems with Applications*, *36*(3), 5932–5941. https://doi.org/10.1016/j.eswa.2008.07.006

Atsalakis, G. S., & Valavanis, K. P. (2013). Surveying stock market forecasting techniques – Part I: Conventional method. In C. Zopounidis (Ed.), *Computation optimization in economics and finance research compendium* (pp. 49–104). Nova Science Publishers.

Balduzzi, P., & Lynch, A. W. (1999). Transaction costs and predictability: Some utility cost calculations. *Journal of Financial Economics*, *52*(1), 47–78. https://doi.org/10.1016/S0304-405X(99)00004-5

Bertoluzzo, F., & Corazza, M. (2014). Reinforcement learning for automated financial trading: Basics and applications. In S. Bassis, A. Esposito, & F. Morabito (Eds.), *Recent advances of neural network models and applications* (pp. 197–213). Springer. https://doi.org/10.1007/978-3-319-04129-2_20

Brogaard, J., & Zareei, A. (2018). Machine learning and the stock market. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3233119

Carhart, M. M. (1997). On persistence in mutual fund performance. *The Journal of Finance*, *52*(1), 57–82. https://doi.org/10.1111/j.1540-6261.1997.tb03808.x

Chan, E. P. (2017). *Machine trading: Deploying computer algorithms to conquer the markets*. John Wiley & Sons, Inc. https://doi.org/10.1002/9781119244066

Charpentier, A., Élie, R., & Remlinger, C. (2021) Reinforcement learning in economics and finance. *Computational Economics*, 1–38. https://doi.org/10.1007/s10614-021-10119-4

Cong, L., Tang, K., Wang, J., & Zhang, Y. (2021). AlphaPortfolio: Direct construction through deep reinforcement learning and interpretable AI. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3554486

Edwards, R. D., Magee, J., & Bassetti, W. H. C. (2007). *Technical analysis of stock trends* (9th ed.). CRC Press.

Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*, *116*(1), 1–22. https://doi.org/10.1016/j.jfineco.2014.10.010

Fischer, T. G. (2018, December). *Reinforcement learning in financial markets – a survey* (FAU Discussion Papers in Economics No. 12/2018). Erlangen. http://hdl.handle.net/10419/183139

Gold, C. (2003). FX trading via recurrent reinforcement learning. In *2003 IEEE International Conference on Computational Intelligence for Financial Engineering* (pp. 363–370). Hong Kong, China. https://doi.org/10.1109/CIFER.2003.1196283

Gričar, S., & Bojnec, Š. (2019). Prices of short-stay accommodation: Time series of a eurozone country. *International Journal of Contemporary Hospitality Management*, *31*(12), 4500–4519. https://doi.org/10.1108/IJCHM-01-2019-0091

Han, Y., Yang, K., & Zhou, G. (2013). A new anomaly: The cross-sectional profitability of technical analysis. *Journal of Financial and Quantitative Analysis*, *48*(5), 1433–1461. https://doi.org/10.1017/S0022109013000586

Han, Y., Zhou, G., & Zhu, Y. (2016). A trend factor: Any economic gains from using information over investment horizons? *Journal of Financial Economics*, *122*(2), 352–375. https://doi.org/10.1016/j.jfineco.2016.01.029

Hassouneh, I., Serra, T., & Bojnec, Š. (2015). Nonlinearities in the Slovenian apple price transmission. *British Food Journal*, *117*(1), 461–478. https://doi.org/10.1108/BFJ-03-2014-0109

Hassouneh, I., Serra, T., Bojnec, Š., & Gil, J. M. (2017). Modelling price transmission and volatility spillover in the Slovenian wheat market. *Applied Economics*, *49*(41), 4116–4126. https://doi.org/10.1080/00036846.2016.1276273

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, *124*, 226–251. https://doi.org/10.1016/j.eswa.2019.01.012

Keim, D. B. (2008). Financial market anomalies. In S. N. Durlauf & L. E. Blume (Eds.), *The new Palgrave dictionary of economics* (2nd ed.). Palgrave Macmillan.

Kidger, P., & Lyons, T. (2020). *Universal approximation with deep narrow networks*. http://arxiv.org/abs/1905.08539v2

Kirkpatrick, C., & Dahlquist, J. (2011). *Technical analysis: The complete resource for financial market technicians*. FT Press.

Kolm, P. N., & Ritter, G. (2021). Modern perspectives on reinforcement learning in finance. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3449401

Liu, Y., Zhou, G., & Zhu, Y. (2020). Maximizing the Sharpe ratio: A genetic programming approach. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3726609

Lo, A. W., & Hasanhodzic, J. (2010). *The evolution of technical analysis: Financial prediction from Babylonian tablets to Bloomberg terminals*. John Wiley & Sons.

Lo, A. W., Mamaysky, H., & Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The Journal of Finance*, *55*(4), 1705–1765. https://doi.org/10.1111/0022-1082.00265

Lynch, A. W., & Balduzzi, P. (2000). Predictability and transaction costs: The impact on rebalancing rules and behavior. *The Journal of Finance*, *55*(5), 2285–2309. https://doi.org/10.1111/0022-1082.00287

Malkiel, B. G. (2007). *A random walk down Wall Street: The time-tested strategy for successful investing* (revised and updated). W. W. Norton & Company.

Menkhoff, L. (2010). The use of technical analysis by fund managers: International evidence. *Journal of Banking & Finance*, *34*(11), 2573–2586. https://doi.org/10.1016/j.jbankfin.2010.04.014

Mosavi, A., Faghan, Y., Ghamisi, P., Duan, P., Ardabili, S. F., Salwana, E., & Band, S. S. (2020). Comprehensive review of deep reinforcement learning methods and applications in economics. *Mathematics*, *8*(10), 1640. https://doi.org/10.3390/math8101640

Murphy, J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. New York Institute of Finance.

Park, C.-H., & Irwin, S. H. (2007). What do we know about the profitability of technical analysis? *Journal of Economic Surveys*, *21*(4), 786–826. https://doi.org/10.1111/j.1467-6419.2007.00519.x

Pendharkar, P. C., & Cusatis, P. (2018). Trading financial indices with reinforcement learning agents. *Expert Systems with Applications*, *103*, 1–13. https://doi.org/10.1016/j.eswa.2018.02.032

Plummer, T. (2010). *Forecasting financial markets the psychology of successful investing*. Kogan Page.

Pring, M. (2002). *Technical analysis explained: The successful investor's guide to spotting investment trends and turning points* (4th ed.). McGraw-Hill.

Rocca, P., Oliveri, G., & Massa, A. (2011). Differential evolution as applied to electromagnetics. *IEEE Antennas and Propagation Magazine*, *53*(1), 38–49. https://doi.org/10.1109/MAP.2011.5773566

Salimans, T., Ho, J., Chen, X., Sidor, S., & Sutskever, I. (2017). *Evolution strategies as a scalable alternative to reinforcement learning*. http://arxiv.org/abs/1703.03864

Storn, R., & Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359. https://doi.org/10.1023/A:1008202821328

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT press. http://incompleteideas.net/book/the-book.html

Xufre Casqueiro, P., & Rodrigues, A. J. L. (2006). Neuro-Dynamic trading methods. *European Journal of Operational Research*, *175*(3), 1400–1412. https://doi.org/10.1016/j.ejor.2005.02.015

Zakamulin, V. (2016). Market timing with moving averages: Anatomy and performance of trading rules. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2585056

Zhang, Z., Zohren, S., & Roberts, S. (2020). Deep reinforcement learning for trading. *The Journal of Financial Data Science*, *2*(2), 25–40. https://doi.org/10.3905/jfds.2020.1.030