Algorithmic Market Efficiency?

Machine Learning, Outperformance and Arbitrage Activity along the Cross-Section of Stock Returns

Constantin Schesch

July 2020

Supervisor: Prof. Augustin Landier

Master's Thesis

MSc in International Finance

Academic Year 2020-2021

JEL Codes: C55, C58, G17, G12, G14

Keywords: *Machine Learning, efficient markets hypothesis, arbitrage, short interest*

Abstract: The literature on market efficiency has historically focussed on how new information is reflected in market prices. In this work, we try to explore how, for a constant information set, newer and better algorithms are reflected in prices, an effect we call *algorithmic market efficiency*.

We build on an emerging body of work in the field of asset pricing that shows how Machine Learning algorithms substantially surpass classical methods like Ordinary Least Squares. Reproducing Gu et al. (2020), we use a large dataset of pricing factors, combined with an array of Machine Learning methods (notably trees and Neural Networks), to forecast stock returns out of sample. Non-linear methods achieve remarkably high predictive power, and portfolios built using these predictions robustly outperform the market.

To study whether this constitutes an arbitrage opportunity, or simply latent risk loadings, we analyse short interest along ML-predictions, and fail to find evidence of arbitrageurs exploiting more advanced algorithms to profit from ML-related pricing anomalies. We also do not find a decline in the profitability of a ML method after its publication. Finally, we explore market prescience, as measured by the relationship between short interest on a stock and its return, and find that it is generally low but not influenced by discoveries in this field.

All of this suggests, but falls well short of proving, that Machine Learning techniques do not constitute an arbitrage opportunity and that markets are algorithmically efficient.

Contents

"The machine does not isolate man from the great problems of nature, but plunges him more deeply into them."

Antoine de Saint-Exupéry, *Terre des Hommes* (1939)

1. Introduction

The financial literature on the predictability of the cross-section of asset returns was born from the body of work on market efficiency. Over the last decades it has progressed slowly, with more and more pricing factors being discovered but also more and more questions being asked about the validity of these anomalies. Only quite recently, some papers have shown how, for a given set of factors, more advanced forecasting methods taken from the field of Machine Learning (ML) can vastly outperform the traditional linear regression. Yet, the implicitly adopted view is often that ML methods simply *learn* asset risk premia : they would thus not constitute an arbitrage opportunity, which is why the implications of these recent discoveries on market efficiency have only rarely been discussed.

Canonical studies of the Efficient Market Hypothesis have focussed on how new information is incorporated into market prices. While important, we argue that this mechanism, which we call *informational market efficiency*, only covers one aspect of the phenomenon: it also matters how, for an unchanged information set, market prices reflect newer and better algorithms for analysis, forecasting and model selection. We style this latter effect *algorithmic market efficiency*.

Since this is an infinitely broad topic, we focus on a very restrictive aspect of the issue: whether the recent contributions applying Machine Learning techniques to classical asset pricing problems constitute an arbitrage opportunity. In particular, we set out to study whether there is a class of arbitrageurs exploiting Machine Learning techniques to outperform financial markets.

Our paper thus contains two main parts. The first is a replication of Gu, Kelly, and Xiu (2020), in which we use a large dataset of pricing factors published in the academic literature, covering 3.7 million observations in the United States over 1958-2016, and an array of relatively simple Machine Learning techniques to forecast stock returns out of sample. Like the above-cited authors, we find that while almost all methods have some predictive power, non-linear methods like Regression Trees and Neural Networks do particularly well. Additionally, portfolios built using their predictions significantly and robustly outperform the market.

The second part combines these ML forecasts with data on short interest to study the relationship between stock return predictions and open short interest: the main point is that we expect to see abnormal short interest around stocks that are forecasted to do poorly. Yet, with the notable exception of OLS for which this mechanism works as expected, the relationship between short interest and returns predicted by more advanced methods is usually negative: relative to others, stocks that are predicted to do poorly are usually shorted less.

We explore the time variation of this phenomenon, which shows some hard-to-explain trends but also that this pattern is broadly consistent since the 1990s, although it seems to have partially receded very recently. We also study related phenomena, namely the post-publication decline of ML anomalies and *market prescience* more generally, but again fail to find evidence of arbitrage opportunities.

Assuming our methods are correct, this suggests one of two things: either markets are profoundly algorithmically inefficient, or Machine Learning techniques do not reveal an arbitrage opportunity and do, in fact, learn to predict stock's rational risk premia. While absence of evidence should never be taken as evidence of absence, we view the latter as more plausible. However, it is clear that much more research into this fascinating question is needed.

Section 2 gives a brief overview of existing work on the topic, with a particular focus on three strands of the academic literature: (i) market efficiency and the predictability of the cross-section of stock returns, (ii) the contribution of Machine Learning to asset pricing and financial economics at large, (iii) studies of arbitrage activity and particularly of short interest. Section 3 proposes a brief formalization of algorithmic market efficiency, and discusses some further conceptual refinements.

Part 5 describes the structure of the Machine Learning algorithm, which is taken from Gu et al. (2020), while part 4 details the various data sources. Section 6 analyzes the predictive performance of the ML methods and section 7 shows how portfolios built around those performances robustly outperforms the markets, both of which confirm Gu et al. (2020)'s results.

In chapter 8, we analyze arbitrage activity, as measured by open short interest, around both ML-portfolios and ML-predictions. Section 9 then studies the (lack of) post-publication decline linked to the discovery of Machine Learning methods. Finally, part 10 studies *market prescience*, as measured by the direct between open short interest and a stock's actual return. Section 11 concludes by offering some thoughts on future research in the field of asset pricing and market efficiency.

2. Literature review

2.1. Literature on Market Efficiency and the Cross-Section of Stock Returns

Early empirical tests of the efficient markets hypothesis, as reviewed in Malkiel and Fama (1970), primarily focussed on weak-form efficiency, i.e. tried to test whether time series of financial prices could be used to make forecasts of future prices. However, as Malkiel and Fama note in the last section of their article focussing on semi-strong market efficiency, a much bigger challenge to the theory comes from forecasts of the cross-section of stock returns. This latter part mainly quotes Jensen (1968)'s work on the CAPM-alpha of mutual funds, which fails to find statistically significant evidence of positive alphas among even a subset of them. Both authors, reflecting a growing consensus within the economic profession at the time, thus conclude that there is no outside information reliably predicting the cross-section of stock returns, and that this further proves the efficient market hypothesis.

The discovery shortly thereafter of the value effect (Basu (1977)), through which firms with high book-to-market ratio outperform the market even after correcting for the CAPM-beta, and of the size effect (Banz (1981)), applying similarly to small-cap companies, quickly shook those certainties. Numerous other studies confirmed these "anomalies", and proposed even more pricing factors. Most notable among these are the Fama-French 3-factor asset pricing model (Fama and French (1993)), the Carhart 4-factor model (Carhart (1997)) and, more recently, the Fama-French 5-factor model (Fama and French (2015)): they have become the workhorse models of asset pricing, and we will use them later to evaluate the risk-adjusted performance of our portfolios.

In a second review on the topic, that similarly reflected the thinking of his time, Fama (1991) argues that tests of market efficiency are fundamentally linked to tests of asset pricing models, and that correctly interpreting the implications of return anomalies is an arduous task:

> *The relations between expected returns and book-to-market equity, size, E/P, and leverage are usually interpreted as embarrassments for the [CAPM] model, or the way it is tested, rather than as evidence of market inefficiency. The reason is that the expected-return effects persist. For example, small stocks have high expected returns long after they are classified as small. In truth, though, the existing tests can't tell whether the anomalies result from a deficient asset-pricing model or persistent mispricing of securities.*

The consensus view that return anomalies reflect entirely rational risk-pricings has broadly stayed in place since the mid-1990s. However, the parts identified by Eugene Fama as necessary to firmly establish this view, namely (i) micro-models for risk-factors, (ii) a critical review of testing methods and (iii) evidence of their post-publication persistence, have yielded mixed results in the academic literature.

The search for rational micro-foundations for the various empirically motivated risk factors has proven both fascinating and arduous. Starting from Ross (1976), Arbitrage Pricing Theory (APT) has tried to explain asset returns as linear combinations of various risk factors and each stock's corresponding factor loadings. This has proven an extremely flexible approach, although we will see later on that the linearity restriction might not account for all scenarios. Additionally, in the wake of Lucas (1978)'s early formulation of a Consumption Capital Asset Pricing Model (CCAPM), many researchers have tried to find consumer-focussed microfoundations for the features of the cross-section of stock returns. Campbell and Cochrane (1999) presents a notable contribution to this literature, in which the authors defend the thesis that all identified "anomalies" can be explained through such unobserved, but nonetheless very real, risk factors. However, the risk of theory dredging in this strand of the literature cannot be understressed.

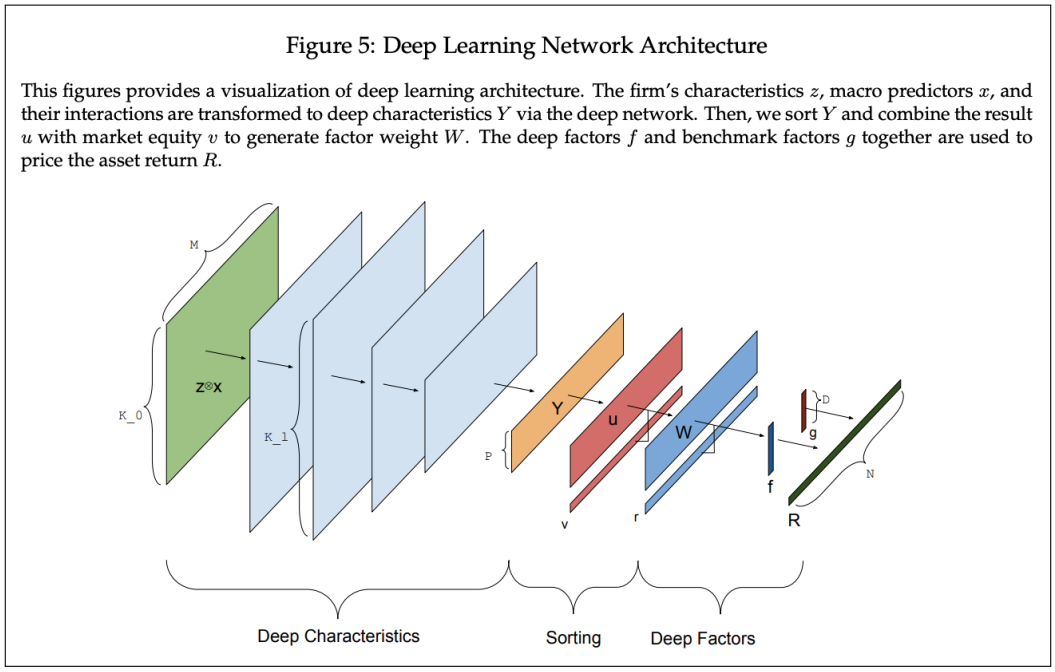
In their "supraview" of the topic, Green, Hand, and Zhang (2013) document 330 separate return-predicting signals publicly identified over 1970–2010. Recently, articles have tried to collect datasets containing all these signals and to assets which animals of this "factor zoo" possess true predictive power. For example, Green, Hand, and Zhang (2017) assemble a dataset of 94 characteristics and find that only 12 that are significant in Fama-Macbeth regressions: notwithstanding, this is the dataset we will be using in our analysis. The multiplication of these detected "anomalies" has also led authors to examine the effects of data-mining and p-hacking, as well as to recommend stricter hypothesis testing procedures (see e.g. Harvey, Liu, and Zhu (2016), who suggest a t-stat test of $t > 3$). In their bid to "tame" this zoo by introducing a recursive test for new factors, Feng, Giglio, and Xiu (2020) assemble a dataset containing 150 firm-level factors, while Hou, Xue, and Zhang (2020) go as far as 447 cross-sectional variables.

As foreseen by Eugene Fama, there is also a growing amount of research showing that anomalies at least partially dissipate after their publication. Dimson and Marsh (1999) provide one of the earliest examples, giving evidence of disappearance of the small size premium in the United Kingdom stock market after the publicization of the anomaly. In a much larger review, McLean and Pontiff (2016) study the post-publication validity of 97 variables, and show that portfolio returns are indeed significantly lower out-of-sample, but do not vanish entirely. However, it is hard to disentangle whether this effect is due to a correction of mispricing or to data-mining an p-hacking in econometric research. Additionally, Jacobs and Müller (2020) perform the same analysis for 241 anomalies in 39 stock markets, and find that the United States is "the only country with a reliable post-publication decline in long-short returns".

2.2. Literature on Machine Learning in Finance

The debate on the predictability of stock returns and its implications for market efficiency is thus far from settled. In our view, a new area, namely that of Machine Learning (ML) applications in Finance, can shed additional light on the topic. While the conceptual challenges are fundamentally the same, it offers a new area of application which could bring in in some new evidence. In line with the boom in ML-related research in Economics in general (see e.g. Currie, Kleven, and Zwiers (2020), who show that it is mentioned in more than 25% of NBER articles published in 2018), these techniques have started to feature prominently in the Finance literature. Readers can refer themselves to Weigand (2019) for a more exhaustive review.

Figure 1: Example of a complex, purpose-built ML architecture, taken from Feng et al. (2018)



The most developed strand of the Machine Learning literature in finance, which is also the one that bears the closest relationship to our present work, focusses on the contribution of ML methods to the prediction of the cross-section of stock returns using fundamental, stock-level information. The most general and widely-cited article adopting this approach is Gu, Kelly, and Xiu (2020), which this memoir largely tries to replicate: authors predict stock returns using a large dataset of 94 stock-level factors (taken from Green, Hand, and Zhang (2017)) and other industry and macroeconomic covariates for all stocks listed in the NYSE, AMEX, and NASDAQ from March 1957 to December 2016.

Gu et al. find significant out-of-sample explanatory power of Machine Learning methods, in particular of non-linear methods like Regression Trees, random forests and Neural Networks. This explanatory power further translates into significant alpha, using top/bottom decile long/short portfolio. Additionally, authors provide a useful review of the (fairly elementary) Machine Learning methods they use, and conduct additional analyses on the explanatory power of various factors and on the importance of non-linear interactions.

Many similar articles follow an analogous strategy to explain the cross-section of stock returns. On the author hand, recently authors have started building much more complex, purpose-built models to predict returns. Feng, Polson, and Xu (2018) look into sorting algorithms that minimize cross-sectional errors, and Chen, Pelger, and Zhu (2019) combine classical stock-level information with rich macroeconomic predictions taken from Long-Short-Term-Memory (LSTM) Networks.

Other articles have tried to adapt these techniques to other contexts, like Wu, Chen, Xu, He, and Tindall (2019) who predict returns in the Chinese stock markets and find robust outperformance, or to other asset classes. Among the latter, we could cite Rojas and Herman (2018) for foreign exchange markets, Dimitriadou, Gogas, Papadimitriou, and Plakandaras (2019) for oil markets, Bianchi, Büchner, and Tamoni (2019) for bond risk premia, Barr, Ellis, Kassab, Redfearn, Srinivasan, and Voris (2017) for the Home Price Index, as well as Culkin and Das (2017) for financial options.

Beyond articles that combine risk factors, i.e. information that has already been extensively used in the academic litterature, recent paper have tried to use ML to introduce new types and sources of information. Sentiment analysis based on text data has typified this approach: see for example Ke,

Kelly, and Xiu (2019) for a recent and extremely promising contribution, and also Gentzkow, Kelly, and Taddy (2019) for a comprehensive review of works on this theme. Like in the case of classical factors, this has also inspired increasing fears of data-picking. Arnott, Harvey, and Markowitz (2019) proposes a clear and systematic backtesting protocol in the age of Machine Learning, built around the key insight: "Acknowledge out of sample is not really out of sample".

Similarly, researchers have tried to study whether ML can contribute to technical analysis, by building models that try to predict future prices using only past prices as information. An early approach to this problem was Allen and Karjalainen (1999), where authors used genetical algorithms to select optimal technical trading rules but showed that, once trading costs are accounted for, buy-and-hold strategies do not consistently outperform the market. In a recent udpate to this approach, Brogaard and Zareei (2018) shows that Machine Learning can improve performance and even generate significant out-of-sample alpha, but that it has been decaying over time. Fischer and Krauss (2018) use Long-Short-Term-Memory Networks and find predictive accuracy and excess returns which seem to be mainly driven by reversal effects.

Hsu, Lessmann, Sung, Ma, and Johnson (2016) similarly use technical information to predict stock returns, but do so across thirty-two countries: they find that the performance of ML methods varies from market to market, notably through its depth and maturity. Krauss, Do, and Huck (2017) perform arbitrage on the S&P500 using neural Neural Networks and tree-based methods, and find robust albeit declining returns that, in their view, "pose a severe challenge to the semi-strong form of market efficiency". Indeed, this strand of the literature is undeniably the one that has studied the link between Machine Learning and market efficiency the most.

Indeed, in most of the literature on the cross-section of stock returns, the implicitly adopted view is that Machine Learning allows to identify latent risk factors. However, while the body of work on asset pricing factors was principally focussed on its relationship with market efficiency, for now little attention seems to been given to the relationship between the effectiveness of Machine Learning methods for stock return prediction and the efficiency of financial markets.

2.3. Literature on Arbitrage Activity and Short Interest

In our attempt to study the market efficiency implications of Machine Learning, we will also try to investigate the arbitrage activity associated with these techniques. There is a very extensive body of theoretical work on arbitrage activity, short sale constraints and their effect of equilibrium asset prices. Gromb and Vayanos (2010) provide a recent survey of these works, and additionally nest the various costs of arbitrage in a simple, tractable model. While the theory on arbitrage is well-developed, studying it empirically has proven a much bigger challenge.

The most straight-forward approach is developped in Hanson and Sunderam (2014), where authors use measures of short interest to infer the amount of equity capital allocated to equity arbitrage strategies, namely value and momentum. Authors further show how this contributed to the decline of their profitability, but argue that such arbitrage may not completely eliminate excess returns. We follow the authors in inferring arbitrage activity from open short interest in our own analysis, because it is the most robust method in our view, although more creative methodologies have been devised and are worth mentioning.

For example, Pontiff (2006) argues that "idiosyncratic risk is the single largest cost faced by arbitrageurs", and thus suggests using idiosyncratic risk as an (inverse) proxy for arbitrage activity. McLean (2010) follow this approach and examine whether the performance of momentum and reversal strategies is the the result of idiosyncratic risks limiting arbitrage. They indeed find a positive relationship between idiosyncratic risk and reversal returns, but find no significant relationship for momentum, which suggests that at least some of these are not exploitable mispricings. Stambaugh,

Yu, and Yuan (2015) also follow this approach.

More exotic approaches that are worth mentioning are Chen, Da, and Huang (2019), who use net arbitrage trading (NAT), defined as the difference between quarterly abnormal hedge fund holdings and abnormal short interest. They show that, among anomalies, abnormal returns are realized only among stocks experiencing large NAT. Lou and Polk (2012) propose to use "comomentum", the high-frequency abnormal return correlation among stocks on which a typical momentum strategy would speculate. They similarly report evidence of arbitrage activity on anomaly portfolios.

Finally, a more recent but nonetheless fascinating body of work tries to study the effects of arbitrage activity on the correction of pricing anomalies. For example, Chordia, Subrahmanyam, and Tong (2014) find that, in recent years, increased hedge fund assets under management, short interest and aggregate share turnover have led to a decline in anomaly-based trading profits in recent years. Jacobs (2015) however detect no relationship between anomaly strategy profits and time-varying market-level arbitrage constraints, although in their view this suggests that the base level of arbitrage restrictions, and not its variations, play a role in explaining market inefficiencies.

In this field as well, results are therefore far from clear, and much work remains to be done. We hope that, here again, the case of Machine Learning can offer some humble light to a tenebrous, complex and fascinating topic.

3. Algorithmic market efficiency

3.1. Classical, or informational, market efficiency

Classical definitions of market efficiency are usually given with respect to an information set available at time t. For example, Jensen (1968) states:

> *A market is efficient with respect to information set I_t if it is impossible to make economic profits by trading on the basis of information set I_t.*

In this context, *economic profit* is a key expression because it emphasizes the importance of appropriately discounting future revenue flows. Mathematically, this means that, given an information set I_t, any financial asset's price at time t, denoted P_t, will be equal to the conditional expectation of its future price P_{t+1} plus intermediary dividend flows D_{t+1}, discounted at the appropriate discount rate Q_t:

$$P_t = \mathbb{E}(Q_t(P_{t+1} + D_{t+1})|I_t) \tag{1}$$

It should be emphasized that the discount factor Q_t is a random variable, which is why it is usually termed the stochastic discount factor (SDF) in the finance literature. We can rewrite this expression in terms of returns to obtain a more tractable form. Denoting its excess return $R_{t+1} = \frac{P_{t+1}-D_{t+1}-P_t}{P_t} - rf_t$ where rf_t is the risk-free rate between t and $t+1$, the EMH simply states:

$$\mathbb{E}_t(R_{t+1}Q_{t+1}|I_t) = 0 \tag{2}$$

As noted e.g. in Timmermann and Granger (2004), this introduces a major joint hypothesis testing problem to all empirical tests of the efficient market hypothesis: any information or method that predicts an asset's return, i.e. that appears to violate (2), could simply be correlated to the stochastic discount factor Q_{t+1}. We set this problem aside for now, and denote $r_{t+1} = R_{t+1}Q_{t+1}$ the risk-adjusted return of an asset : market efficiency then equivalently means $\mathbb{E}_t(r_{t+1}|I_t) = 0$.

Because conditional expectations are mathematically complex objects that strike the human mind with deceptive familiarity, equation (2) has a seemingly very intuitive interpretation. *Instinctively*, if we had $\mathbb{E}_t(r_{t+1}|I_t) \neq 0$, say $\mathbb{E}_t(r_{t+1}|I_t) = \alpha > 0$, we would *obviously* expect some investor to buy the asset and win a risk-free excess return of α. This type of arbitrage would then drive the price up, so that the expected return conditional on I_t would again be 0 : this is the classical arbitrage argument for market efficiency.

However, in more rigidly mathematical terms, equation (2) only defines a function as being null. Indeed, denoting $\mathfrak{I}$ the space from which the information set I_t is drawn and $\mathcal{I} \subset \mathcal{P}(\mathfrak{I})$ the associated $\sigma-$algebra, the conditional expectation $\mathbb{E}_t(r_{t+1}|I_t)$ is only defined as some $\mathcal{I}$-measurable function $g \in \mathcal{M}(\mathfrak{I}, \mathcal{I}, \mathbb{R})$ such that, for any $\mathcal{I}-$measurable function $f \in \mathcal{M}(\mathfrak{I}, \mathcal{I}, \mathbb{R})$, we have

$$\mathbb{E}\big(g(I_t)f(I_t)\big) = \mathbb{E}\big(r_{t+1}f(I_t)\big) \tag{3}$$

and we then say that $\mathbb{E}_t(r_{t+1}|I_t) = g(I_t)$. Since the definition of the conditional expectation is not constructive, (2) makes a statement about a very large class of functions. Conversely, if EMH is violated, the set of functions for which (2) nonetheless holds might actually be very large.

3.2. Algorithmic market efficiency

Turning back to the example of our hypothetical arbitrageur, it seems clear that this argument for market efficiency at time t will only hold if he knows this conditional expectation function exactly, i.e. if he knows $g(I_t)$. This immediately introduces a two-fold problem: (i) the space $\mathcal{M}(\mathfrak{I}, \mathcal{I}, \mathbb{R})$ of $\mathcal{I}$-measurable functions from $\mathfrak{I}$ to $\mathbb{R}$ is vast, and might very well include functions that have not been discovered at time t and will thus not be available to the arbitrageur; (ii) even if the optimal function g has already been discovered at time t, he will still need to search the space $\mathcal{M}(\mathfrak{I}, \mathcal{I}, \mathbb{R})$ to find it, which is a complex and error-prone process.

Arbitrageurs can therefore only be as good as the arbitrage-detection methods of their time, and it is clear that any arbitrage argument for market efficiency should include these two elements in its formulation. Luckily, (i) trying to invent complex measurable functions to map a large feature space onto an explained variable, and (ii) finding creative ways of searching the gargantuan set of such functions to find optimal ones according to a specified risk metric is precisely the endeavour of Machine Learning.

To introduce Machine Learning, or more generally any forecasting method, into our definition of market efficiency, we thus define the set of algorithms for predicting excess returns available at time t, which we denote $\mathcal{A}_t \subset \mathcal{M}(\mathfrak{I}, \mathcal{I}, \mathbb{R})$. We call the elements of $\mathcal{A}_t$ algorithms because they involve complex model selection, tuning and training procedures, but mathematically they are simply $\mathcal{I}$-measurable functions from $\mathfrak{I}$, the set from which I_t is drawn, to $\mathbb{R}$, the set of values that the expected excess return $\mathbb{E}_t(r_{t+1}|I_t)$ can take.

Drawing on Timmermann and Granger (2004), we can then say that a financial market is efficient with respect to the set of algorithms $\mathcal{A}_t$, or simply *algorithmically efficient*, if:

$$\forall a \in \mathcal{A}_t \quad \mathbb{E}_t\big(r_{t+1}\ a(I_t)\big) = 0 \tag{4}$$

In words, this means there is no algorithm a_t within the set available at time $\mathcal{A}_t$ that can, using the available information set I_t, generate information about the expected excess returns of stocks market. It seems intuitive that *informational market efficiency* implies *algorithmic market efficiency*, which a closer inspection of (2) and (3) immediately confirms:

$$\forall a \in \mathcal{A}_t \subset \mathcal{M}(\mathfrak{I}, \mathcal{I}, \mathbb{R}) \qquad \mathbb{E}_t\big(r_{t+1}\ a(I_t)\big) = \mathbb{E}_t\big(\mathbb{E}_t(r_{t+1}|I_t)\ a(I_t)\big) = \mathbb{E}_t\big(0 \times a(I_t)\big) = 0 \tag{5}$$

and thus (2) $\Rightarrow$ (4).

We can perhaps shed some more light on $\mathcal{A}_t$ by detailing its elements. In analogy to our empirical estimation procedure, we turn to a setting with n stocks, for which we observe returns from 1 to t, $\mathbf{r}$ in vector notation, and p characteristics from 0 to t, written $\mathbf{z}$. The information set thus writes $I_t = \{\mathbf{z}_t, \mathbf{z}_{t-1}, \ldots, \mathbf{z}_0, \mathbf{r}_{t-1}, \ldots, \mathbf{r}_0\}$. We then focus on algorithms $a = \{s, g\}$ that combine a *prediction function* $g(\mathbf{z}_t) = \mathbf{r}_{t+1}$ from characteristics into future returns, and a *selection function* $s(\mathbf{z}_{t-1}, \ldots, \mathbf{z}_0, \mathbf{r}_{t-1}, \ldots, \mathbf{r}_0) = g$ from past characteristics and returns to prediction functions.

In the case of a standard linear regression (OLS), a would thus write:

$$a_{OLS}(I_t) = \theta \mathbf{z}_t \quad \text{with} \quad \theta \in \underset{\theta}{\operatorname{argmin}} \sum_{\tau=0}^{t-1} \sum_{k=0}^{n} (r_{n,\tau} - \theta \mathbf{z}_{n,\tau})^2 \tag{6}$$

More complex prediction algorithms can then involve more complex prediction functions g, ranging from a fairly straightforward Lasso function to Neural Networks that can essentially not be understood by the human mind. Additionally, they will involve increasingly complex parameter and hyperparameter selection methods s, with rolling windows, cross validation, ad hoc loss functions etc.

3.3. Semi-strong, strong and super-strong algorithmic efficiency

Following Malkiel and Fama (1970), the efficient market hypothesis is usually classified into three different formulations: (i) the *weak form*, which says that prices reflect all information from past prices; (ii) the *semi-strong form*, according to which prices reflect all public information, including prices but also fundamental stock characteristics; (iii) the *strong form*, stating that prices reflect all public and private information.

In practice most research on informational market efficiency has focused on the semi-strong formulation. We might however try to establish similar refinements for algorithmic market efficiency by coming back to the definition of $\mathcal{A}_t$. For now, we have only characterized it somewhat vaguely as the set of algorithms for predicting excess returns available at time t.

One could first think that it only includes all publicly available forecasting methods, for example from publications in the academic literature, because arbitrageurs do not have the time and resources to discover radically new algorithms by themselves. We call this view the *semi-strong form* of algorithmic market efficiency.

However, one might wonder whether there is not an effect similar to the "reverse file drawer bias" identified by Timmermann and Granger (2004): "A researcher who genuinely believes he or she has identified a method for predicting the market has little incentive to publish the method in an academic journal and would presumably be tempted to sell it to an investment bank." Similarly, a researcher, or even the research department of a bank or hedge, might be tempted not to publicize a newly discovered forecasting method if it allows them to profit from currently mispriced assets. We dub the opposing view, according to which prices reflect information as analyzed by all public and privately discovered algorithms, the *strong form* of algorithmic market efficiency.

Finally, the most optimistic view simply states that (4) applies $\forall a \in \mathcal{M}(U, \Sigma, \mathbb{R}^n)$, i.e. that all algorithms, even those that have not yet been discovered are reflected in prices. We call this view the *super-strong form*. Since it will at first seem obviously impossible, we can try to give a very simple argument for why it might nonetheless be plausible: financial markets incorporate information through analysis channels that are not simple computational algorithms but rather complex social processes (ranging from IFRS standards to the opinion section of the Financial Times). It is quite likely that these processes are best described by a class of mathematical functions that has not yet been discovered and that, even more radically, is too complex to ever be discovered explicitly. It could thus be that markets are more efficient than our algorithms will ever allow us to understand.

Figure 2: Different Formulations of Informational and Algorithmic Market Efficiency

Market Efficiency Matrix		Algorithmic Market Efficiency		
		Semi-strong: public algorithms	**Strong: public and private algorithms**	**Super-strong: discovered and undiscovered algorithms**
Informational Market Efficiency	**Weak: price information**	Prices reflect all information from past prices, as analyzed by all publicly available algorithms.	Prices reflect all information from past prices, as analyzed by all public and privately discovered algorithms.	Prices reflect all information from past prices, as analyzed by all discovered and undiscovered algorithms.
	Semi-strong: public information	Prices reflect all public information, as analyzed by all publicly available algorithms.	Prices reflect all public information, as analyzed by all public and privately discovered algorithms.	Prices reflect all public information, as analyzed by all discovered and undiscovered algorithms.
	Strong: public and private information	Prices reflect all public and private information, as analyzed by all publicly available algorithms.	Prices reflect all public and private information, as analyzed by all public and privately discovered algorithms.	Prices reflect all public and private information, as analyzed by all discovered and undiscovered algorithms.

3.4. Testing algorithmic EMH : our Machine Learning Framework

Informational market efficiency and algorithmic market efficiency, as stated in (2) and (4), are impossible to demonstrate empirically, but quite straightforward to reject. It would, in theory, be enough to show that we can indeed find functions that (even very imperfectly) predict the cross-section of stock returns r_{t+1}.

However, we are faced with a problem that has plagued the litterature in this field from the outset, and that we had set aside by examining stochastically discounted returns $r_{t+1} = Q_{t+1}R_{t+1}$. Indeed, we can only observe the raw returns R_{t+1}, because the stochastic discount factor is by nature inobservable. In fact, as many articles have show, simple correlation between stock returns and the SDF will imply a forecastable cross-section of stock returns, without violating EMH in the slightest. Formally, we can rewrite (2) as :

$$\mathbb{E}_t(R_{t+1}|I_t) = \frac{\text{Cov}(R_{t+1}, Q_{t+1}|I_t)}{\mathbb{E}_t(Q_{t+1}|I_t)} \tag{7}$$

We then rewrite this SDF loading as a function $\lambda(z_t)$ of observables plus an error term η_{t+1} with zero mean:

$$\frac{\text{Cov}(R_{t+1}, Q_{t+1}|I_t)}{\mathbb{E}_t(Q_{t+1}|I_t)} = \lambda(z_t) + \eta_{t+1} \tag{8}$$

The expected return on a stock can then be written as the sum of this SDF loading plus an arbitrage signal $\theta(z_t)$ and a second error term e_{t+1}:

$$\mathbb{E}_t(R_{t+1}|I_t) = \lambda(z_t) + \eta_{t+1} + \theta(z_t) + e_{t+1} = 0 \tag{9}$$

This equation is always true because, if EMH is true, it simply implies that we have $\theta(z_t)+e_{t+1}=0$, i.e. the arbitrage signal is null. A test of EMH is thus simply a test of $\theta(z_t)+e_{t+1}=0$.

This motivates our Machine Learning framework, which uses a very simple additive prediction error model following Gu et al. (2020):

$$r_{i,t+1} = \mathbb{E}_t(r_{i,t+1}|I_t) + \epsilon_{i,t+1} = g(\mathbf{z}_t) \tag{10}$$

where we try to estimate, or *learn*, $\mathbb{E}_t(r_{i,t+1}|I_t)$ using the *model* $g(\mathbf{z}_t)=\mathbf{r}_{t+1}$ and a *selection method* $s(\mathbf{z}_{t-1},\dots,\mathbf{z}_0,\mathbf{r}_{t-1},\dots,\mathbf{r}_0)=g$. Since we can only observe $g=\lambda+\theta$, we run again into the fundamental joint testing problem of asset pricing, because we obviously do not know the correct asset pricing model λ.

Although it does not entirely allow us to avoid this inescapable quandary, the previous discussion on algorithmic market efficiency helps us to somewhat refine tests of EMH. Indeed, in keeping with the notations above, informational EMH simply states that:

$$\mathbb{E}_t(R_{t+1}|I_t) - \lambda(z_t) + \eta_{t+1} = 0 \qquad \Rightarrow \qquad \mathbb{E}_t(R_{t+1} - \lambda(z_t)|I_t) = 0 \tag{11}$$

Markets are then efficient with respect to the set of algorithms $\mathcal{A}$ if:

$$\forall a \in \mathcal{A} \quad \mathbb{E}_t\big[a(I_t)\ (R_{t+1} - \lambda(z_t))\big] = 0 \tag{12}$$

In words, this means that the "projection" of the true arbitrage signal θ onto $\mathcal{A}$ is null. If we denote $\mathcal{A}^{pub}$ the set of public knowledge algorithm and $\mathcal{A}^{priv}$ that of private algorithms, we can write the more refined versions of AEMH as:

$$\begin{aligned}
&\text{Semi-strong AEMH:} && \forall a \in \mathcal{A}^{pub} \quad \mathbb{E}_t\big[a(I_t)\ (R_{t+1} - \lambda(z_t))\big] = 0 \\
&&& \text{and } \exists a \in \mathcal{A}^{priv} \quad \mathbb{E}_t\big[a(I_t)\ (R_{t+1} - \lambda(z_t))\big] \neq 0 \\
&\text{Strong AEMH:} && \forall a \in \mathcal{A}^{pub} \cup \mathcal{A}^{priv} \quad \mathbb{E}_t\big[a(I_t)\ (R_{t+1} - \lambda(z_t))\big] = 0 \\
&&& \text{and } \exists a \in \mathcal{M} \setminus (\mathcal{A}^{pub} \cup \mathcal{A}^{priv}) \quad \mathbb{E}_t\big[a(I_t)\ (R_{t+1} - \lambda(z_t))\big] \neq 0 \\
&\text{Super-strong AEMH:} && \exists a \in \mathcal{M} \quad \mathbb{E}_t\big[a(I_t)\ (R_{t+1} - \lambda(z_t))\big] \neq 0
\end{aligned} \tag{13}$$

Combining this tripartition with the test of $\theta=0$ shows that one of four scenarios has to be true:

(i) Machine Learning is not an arbitrage opportunity and has never been because markets are super-strongly algorithmically efficient

(ii) Machine Learning is not an arbitrage opportunity but has been in the past because market are only strongly algorithmically efficient

(iii) Machine Learning is an arbitrage opportunity and is currently exploited because markets are semi-strongly algorithmically efficient

(iv) Machine Learning is an arbitrage opportunity and has never been exploited because markets are profoundly algorithmically inefficient

Put very briefly, the modest aim of this work is to convince the reader that we are either in case (i) or in case (iv), and then to cajole him into believing, as we do, that (i) is much more plausible.

4. Data and Data Sources

For the Machine Learning part, as well as for further analyses and notably that of arbitrage activity, we use quite a few different data sets. They are synthesized in Table 1, and we give a brief description of data sources and initial manipulations in this section.

The core dataset of our analysis is the 94-variable set of factors published by Green et al. (2017) and adjusted by Gu et al. (2020), which we retrieved from Dacheng Xiu's website in September 2019. It contains 3,760,208 for 94-factors that have been published in the academic litterature as potential predictors of the cross-section of asset returns. The dataset covers all stocks listed in the NYSE, AMEX and NASDAQ, ranging from March 1957 (the start date of the S&P 500) to December 2016.

Although Green and coauthors find that only a small subset of them have true predictive power, we follow Gu et al. (2020) in keeping the whole dataset, as Machine Learning methods are specifically designed to take advantage of all available information while limiting the risk of overfitting. The factors contained in this dataset are sometimes very classical, e.g. CAPM-beta, firm size, book-to-market ratio and various measures of momentum, but others are much more exotic, e.g. changes in tax expense, convertible debt indicators or financial statement scores.

Among the 94 characteristics, 61 are updated annually, 13 quarterly and 20 monthly. Additionally, the observations have been adjusted by Gu and coauthors so as to only include publicly available information: monthly characteristics are delayed by 1 month, quarterly ones by 4 months, and annual factors by at least 6 months. The shift has already been done in the initial dataset, so factors can be used directly to predict returns in "real time". This avoids forward-looking bias in the Machine Learning component.

We match Gu et al.'s database, in which observations are identified by *date* and *Permno*, with CRSP returns, using those same identifiers. We also add the five Fama-French aggregate factors, namely Small-minus-big firm size, High-minus-low book-to-market ratio, Robust-minus-weak operating profitability and Conservative-minus-aggressive investment. We further include the Momentum aggregate factor from the Carhart 4-factor model. All of these have been downloaded from Ken French's website in June 2020. Finally, we add a large dataset of aggregate macro factors taken from Amit Goyal's website. These notably include information on growth, inflation, dividend yields, risk-free interest rates as well as risky corporate debt yields.

Matching is generally not a problem, but we drop all observations for which returns are missing. Since our algorithm progresses on an annual basis, we clip the observations for some months from 1957 because the dataset does not contain the full year. Additionally, we rescale all factors so that their mean and standard deviation across the full dataset are zero and one respectively. The final data set for the Machine Learning component contains 3,728,811 observations for 120 variables. Each row is uniquely identified by a firm's *Permno* and the *date* corresponding to each month's last trading day. The dataset thus ranges from January 31st, 1958, to December 30th, 2016. It contains 29,830 unique *Permno*s corresponding to equally many publicly traded firms on various exchanges in the United States of America.

Table 1: Data Sources

Data Source	Retrieved on	Observations	Variables of interest
Factors from Green et al. (2017)	Sept. 2019	3,760,208	94 factors (see Annex)
CRSP Database	Oct. 2019	4,514,430	*ret, mktcap, shrout*
Fama-French factors	June 2019	1,119	*smb, hml, rmw, cma, mom*
Amit Goyal's macro predictors	Nov. 2019	1,776	16 macro factors
Compustat Supplemental Short Interest File	Feb. 2020	2,221,113	*shortintadj*
CRSP / Comp. Merged Security Monthly	Feb. 2020	6,130,813	*cshtrm*

The latter parts of this work, focussing on arbitrage around Machine Learning Portfolios and Predictions, relies on two additional data sources: the Compustat Supplemental Short Interest File and the CRSP / Compustat Merged Security Monthly Dataset. We discuss merging and interpolation issues emerging from these two sets in section 8.2.

5. Structure of the algorithm

The algorithm to investigate machine-learning-related market anomalies essentially consists of two interrelated parts: (i) a proper Machine Learning component and (ii) a loop component applying the latter over all years. We then analyse the performance of the algorithm using yet again two distinct parts, the first one focussing on within- and out-of-sample predictive performance, and the other converting predictions into tradeable portfolios and verifying whether they outperform the market.

5.1. Machine Learning Component

The Machine Learning component uses different Machine Learning methods to predict future returns using a dataset of past returns. In our context, a *method* means both a *statistical or algorithmic technique* (ranging from OLS to Neural Networks) combined with a *cross-validation* strategy to tune the latter's hyperparameters.

We test 16 different Machine Learning techniques : Ordinary Least Squares (OLS), Lasso regression, Ridge regression, Elastic Net regression, Principal Component Regression (PCR), Principal Least Squares (PLS), the Regression tree, the Gradient Boosted Regression Tree, the Random Forest as well as 1-, 2-, 3-, 5- and 10- Layer Neural Networks. These are not particularly complex techniques, in fact they belong to the standard canon of ML techniques taught even at the undergraduate level: since the ambition of our work is to study algorithmic sources of information that are likely to have been used in practice, we actually aim to use very simple and straightforward methods.

We will not review each method individually, and readers are referred to either the original publications mentioned in Table 6, to Gu et al. (2020) which contains an excellent, synthetic description of these methods, or simply to any graduate-level Machine Learning textbook. However, to make our methodology somewhat clearer, we focus on two interesting sets of methods: linear regression and its penalized versions, as well as Neural Networks.

First, the training data set is split again into two new, smaller, training and testing datasets. Then different hyperparameters (see above), e.g. in the case of 3-layer Neural Networks architectures like (16,8,4), (32,16,8) and (64,32,16), are trained on the small training dataset. Their out-of-sample performance is then tested on the small test dataset, and the model best-performing hyperparameter is selected and retrained on the full, initial train dataset. trainng for 3-layer networks, are trained on the training data set.

In fact, the algorithm uses k-fold cross-validation (with three or two folds, depending on the computational complexity of a technique) and then retrains the technique on the full training dataset. For computationally unintensive algorithms, e.g. Ridge regression, this allows for exhaustive testing of the whole parameter space, which allows for fairly high confidence in the obtained hyperparameters. For more intensive ones, we have to rely on a combination of cross-validation with manual tuning, whereby intuition, trial-and-error and good fortune are used to restrict the space of tested hyperparameters. This introduces very fundamental reporting bias concerns, especially when tested values are not fully accounted for. Finally, for the most intensive methods, i.e. 10-Layer Neural Networks, we have to rely on entirely manual hyperparameter tuning, which brings the least confidence in the obtained result.

All Machine Learning algorithms start with a set of data (the "in-sample" sample) on which we will train their parameters and tune their hyperparameters. In the case of Ordinary Least Squares, there

are no hyperparameters to be tuned, so we directly estimate the parameter using the full training set. Keeping the notation of section 3, it directly writes:

$$\widehat{\mathbf{r}}_{t+1} = \mathbf{z}_t \, \widehat{\theta} \quad \text{with} \quad \widehat{\theta} \in \underset{\theta}{\operatorname{argmin}} \sum_{\tau=0}^{t-1} \sum_{k=0}^{n} (r_{n,\tau+1} - \theta \mathbf{z}_{n,\tau})^2 = \underset{\theta}{\operatorname{argmin}} \, ||\mathbf{R} - \mathbf{Z} \, \theta||_2^2$$

where upper-case letters denote matrices assembling the vector observations across time and $|| \cdot ||_2$ denotes the euclidean (ℓ_2) norm. Of course, the minimization problem to the right has a famous closed-form solution, $\widehat{\theta} = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{R}$. Without hyperparameters, the ML procedure is thus remarkably simple, als OLS is the only method we use that does not have any.

Indeed, if the sample is small and the number of parameters is low (which admittedly is not the case here), this introduces a risk of overfitting in-sample that will translate into poor out-of-sample performance. To overcome this issue, improved versions of OLS that constrain the size of the estimator have been devised. The optimal strength of this regularization naturally depends on the problem at hand, and will thus need to be tuned through a hyperparameter. In particular, we focus on two famous regularization methods:

$$\text{Ridge } (\ell_2\text{-regularization}): \quad \widehat{\mathbf{r}}_{t+1} = \mathbf{z}_t \, \widehat{\theta} \quad \text{with} \quad \widehat{\theta} \in \underset{\theta}{\operatorname{argmin}} \, ||\mathbf{R} - \mathbf{Z} \, \theta||_2^2 + \lambda_{ridge} ||\theta||_2^2$$

$$\text{Lasso } (\ell_1\text{-regularization}): \quad \widehat{\mathbf{r}}_{t+1} = \mathbf{z}_t \, \widehat{\theta} \quad \text{with} \quad \widehat{\theta} \in \underset{\theta}{\operatorname{argmin}} \, ||\mathbf{R} - \mathbf{Z} \, \theta||_2^2 + \lambda_{lasso} ||\theta||_1$$

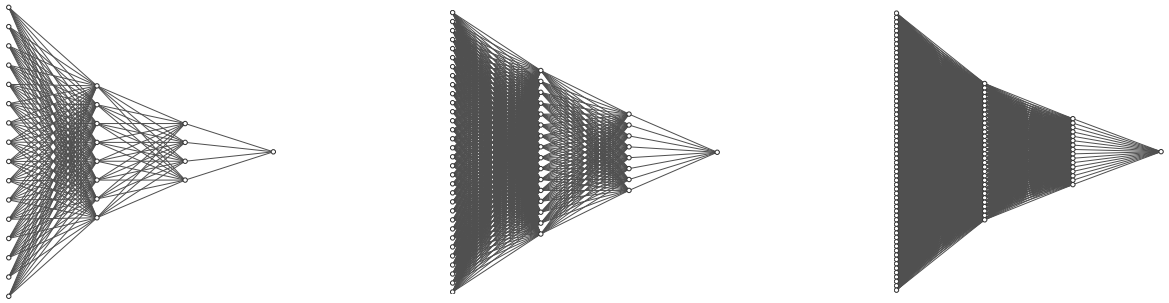
To tune the value of the hyperparameters, we resort to 3-fold cross-validation. For each tested value of the hyperparameter, the training set is split three times into a smaller training dataset (2/3 of the size of the overall training set) and a smaller testing dataset (1/3 of the size of the overall training set). The model parameter (θ) is then trained on each training dataset and its R^2 score is computed on the corresponding testing dataset. R^2 scores are averaged across all three folds, and the tested hyperparameters with the highest average score is selected. Finally, the model parameter is trained on the full, initial training dataset using the optimal hyperparameter value.

In the case of Ridge and Lasso, we test λ_{lasso} and λ_{ridge} on 100 values log-uniformly spaced between 10^{-10} and 10^{10}. In general, very low values of λ_{lasso} and λ_{ridge} are selected in the cross-validation phase, which means that optimal regularization is very weak: this is probably because we usually have many observations in comparison to the number of variables, and thus regularized versions do not perform better than the simple Ordinary Least Squares estimator. In a sense, Lasso and Ridge, as well as their combination Enet, are thus not very useful Machine Learning techniques in our setting, but they have helped us explain how we proceed.

For Ridge and Lasso, we could test a very large hyperparameter space because the computation of the estimates is fast since they are quite simple. For most other methods however, the estimation of one model is quite lengthy. Additionally, many models (notably Regression Trees and its derivatives) have many, many customization options, which makes the number of possible hyperparameter combinations very very large: an exhaustive search of the hyperparameter space through cross-validation is thus prohibitive in terms of processing power. We thus had to resort to a combination of manual tuning, typically for hyper-parameters that produced universally sub-par results, and of cross-validation for a restricted set of hyperparameters, whose optimal values change over time.

This was typically the case for Neural Networks, which are quite advanced model that take important amounts of time to train. Although normally the depth of the neural network, i.e. the number of hidden layers between the input and output layer, is treated as a hyperparemeter of its own, we used networks of different depths and only compare their performance ex-post. However, there are other hyperparameters left to be tuned, notably the activation function, the learning rate function and the

Figure 3: Tested Architectures for 3-Layer Neural Networks



(16, 8, 4) (32, 16, 4) (64, 32, 16)

number of nodes in each layer. Manual experimentation has shown that, in our setting, logistic activation functions with exponentially decreasing learning rates usually perform best. We test different values for the exponent in the learning rate functions ($t \in \{0.1, 0.5, 0.9\}$) as the optimal value changes from time to time.

The most important hyperparameter left to be tuned is the number of nodes in each hidden layer. In the case of 3-Layer Neural Networks, we thus test specifications with $(16, 8, 4)$, $(32, 16, 4)$ or $(64, 32, 16)$ structures, where the sequence indicates the number of nodes in each successive hidden layer. Figure 3 shows stylized representations of the different specifications being tested. Here again, the optimal architecture selected varies through time. Gu et al. (2020) have studied the question of time-varying model complexity in more depth, but have not given it much further thought in this work.

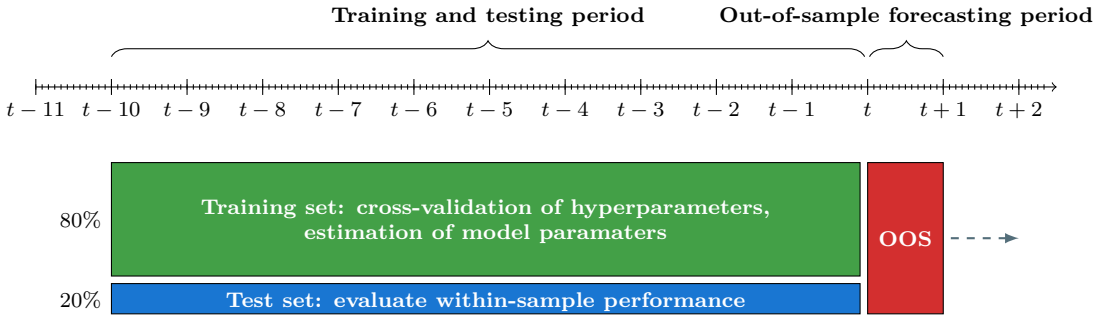
Finally, the most extreme cases are 5- and 10-Layer Neural Networks, where training time is so prohibitive that we had to manually tune all hyperparameters and simply renounce cross-validation. This is also why our results, according to which networks with high numbers of layers perform more poorly than their lower-level counterparts, should be taken with healthy skepticism even though Gu et al. (2020) reach the same conclusion: this could simply be due to the fact that current computational constraints make it difficult to fully search their hyperparameter space in a dynamic setting. Perhaps future research will find that better-tuned deep Neural Networks do as well as, or even better than, shallow Neural Networks.

5.2. Loop Component

The loop component is probably the most straightforward, since it applies the same basic procedure to each year t between 1958 and 2016. First, the data of the last ten years is aggregated and flattened into a single monthly dataset matching returns to firm-level and macro-level factors. For years to close to the datasets beginning date, only the available dates are used: for, say, t=1962, we use 1958, 1959, 1960 and 1961. Many other possibilities, with shorter and longer rolling intervals, or even linearly and non-linearly expanding ones could be tested, but we have adopted this alternative for its simplicity.

In particular, it implies that all observations are treated with equal weight, irrespective of whether they date from $t - 1$, i.e. are fairly recent, or $t - 10$, i.e. are fairly old. Additionally, subsequent returns from the same company become indistinguishable to the algorithm. This restriction is made necessary by the fact that not all future companies whose return needs to be predicted will have an accessible history, therefore the model needs to be trained *anhistorically*. Moreover, this limitation is partially offset because the dataset contains many factor variables pertaining to short-, medium- and long-term momentum.

Figure 4: Structure of the Machine Learning and Loop Components



This dataset of previous years' returns and corresponding factors is then split into a train and a test dataset, using an 80/20 split. The training data is then fed into the Machine Learning component described above (which will, in fact, split this training data again a few times to cross-validate the hyperparameters), which simply returns a model with tuned hyperparameters and trained parameters.

Finally, the performance of the method is then measured (a) on the testing data (in blue), to see how well the model predicts returns sampled from the same distribution as the training data, and (b) on data of year t (in red), to measure whether it actually holds true out-of-sample predictive power.

For each of these we use two scores to assess the quality of our models: (i) the R^2, which corresponds to a standard quadratic loss function and can be interpreted as in any standard OLS regression, (ii) the Spearman ρ, a non-parametric measure of rank-correlation, which reflects the fact that what matters in our prediction problem is the relative position among returns of a given company, and less the numerical value of its return.

6. Analyzing the Predictive Performance of ML Algorithms

Before turning to portfolios built using the Machine Learning predictions, we simply try to analyse the in- and out-of-sample predictive performance of the various algorithms described above. We first study the cross-sectional performance of the algorithm, i.e. how well ML techniques using factor information can explain the variance of stock-returns, and then turn to a time-series analysis, to see whether and how this explanatory power changes over time.

6.1. Cross-sectional analysis

To asses the cross-sectional explanatory power of a Machine Learning technique, we will always have to restrict ourselves to a certain month or a certain year. Indeed, since by construction in our approach the prediction model changes from year to year, this means that we cannot (or, rather, should not) test its predictive power over the full horizon of returns. In particular, we will see in the following sections that non-linear methods, which in generally have the best performance, increase this performance quite substantially over time, as the number of available years and observations increases. This effect would be muddled if we only studied predictive performance over the full sample.

To asses the in- and out-of-sample predictive performance of our models, we will rely on simple graphical inspection, but also on two scoring methods. The first score, undeniably the most traditional, is simply the R^2 of predictions against actual returns. Keeping with the notations above, the R^2 at time t would write:

$$R^2 = 1 - \frac{\sum_{i=0}^{N_t} \left(r_{i,t} - \widehat{r}_{i,t}\right)^2}{\sum_{i=0}^{N_t} \left(r_{i,t} - \overline{r}_t\right)^2} \tag{14}$$

The R^2 is the most important score, because it is the one we use in the cross-validation step to tune our hyperparameters and then to train our parameters. However, it has an important drawback, because in our context its behaviour is somewhat different from the one that readers familiar with linear regressions will expect: in in-sample linear regressions with a constant, minimizing squared predictions errors means maximizing the R^2; it it therefore always always positive, because the naive estimator always predicting the sample average $\overline{r}_t$ yields an R^2 of zero.

However, in our setting, most methods we use do not have a theoretically motivated R^2-maximizing property. More importantly, since we predict returns out-of-sample, the naive estimator would be tuned to the previous period's average return, $\overline{r}_{t-1}$, and not the current period's, which means that even a naive estimator could return a negative R^2. In plain English, the R^2 will not always be between 0 and 1: in fact, it will quite often be well below zero. This seemingly technical element has an important consequence for its appropriateness in our setting: the R^2 is affected by the mean of predicted returns, but since we are trying to build portfolios of stocks, we care more about their relative return rather than their true return.

Denoting $\overline{r}_t$ the true average return, and $\widehat{\overline{r}}_t$ the average of predictions, we can rewrite the R^2 as follows:

$$R^2 = 1 - \frac{\sum_{i=0}^{N_t} \left[(r_{i,t} - \overline{r}_t) - (\widehat{r}_{i,t} - \widehat{\overline{r}}_t) - (\widehat{\overline{r}}_t - \overline{r}_t) \right]^2}{\sum_{i=0}^{N_t} \left(r_{i,t} - \overline{r}_t \right)^2} \tag{15}$$

The first two terms of the sum in the numerator are the predicted or actual return minus the mean of the predicted or actual returns: they thus measure the relative performance of a stock in comparison to the true or predicted market, which is what we are actually interest in. However, the third term is the difference between the mean of true and actual returns, a measure that does not give us any information on the relative quality of our predictions, but nonetheless enters into the R^2: the score is thus blurred by this essentially irrelevant mean-error component, which makes it an imperfect score in our setting.

To overcome this issue, we choose to study and report a second score, which more precisely measures the relative performance of stocks: Spearman's rank correlation coefficient, or simply Spearman's ρ. This measure is very analogous to the better-known Pearson correlation coefficient, which measures the correlation between two variables. However, Spearman's ρ focusses exclusively on ranks of observations and not on their values: in fact, it is the Pearson correlation coefficient of the ranks of observations from both variables. In our case, if we denote $rk(r_{i,t})$ the rank of the true return of stock i among true returns, and $rk(\widehat{r}_{i,t})$ the rank of its predicted return among predicted returns, the score will be:

$$\rho = \text{Corr}\big(rk(r_{i,t}), rk(\widehat{r}_{i,t})\big) = \frac{\text{Cov}\big(rk(r_{i,t}), rk(\widehat{r}_{i,t})\big)}{\sigma_{rk(r_{i,t})}\sigma_{rk(\widehat{r}_{i,t})}} = 1 - \frac{6 \times \sum_{i=0}^{N_t} \big(rk(r_{i,t}) - rk(\widehat{r}_{i,t})\big)^2}{n(n^2-1)} \tag{16}$$

Spearman's rho, like Pearson's correlation coefficient, is bounded by 1, indicating perfect rank dependence, and -1, indicating perfect inverse rank dependence. This second score is a non-parametric measure of fit which, in our view, (i) better focusses on the relative performance of stocks and (ii) is less affected by the out-of-sample nature of our estimation exercise.

The main result from analysing these various scores, both within- and out-of-sample, is that non-linear methods, and in particular Regression Trees, random forest and Neural Networks, perform significantly better than linear methods like OLS, its penalized variations but also PCR and PLS. This can be seen quite clearly in Table 2, which shows both scores for all methods in the year 2016 (the last one in our sample). In-sample scores are clearly always higher than out-of-sample scores, but they are of course also less interesting.

Table 2: In- and out-of-sample scores for 2016

	In-sample (2006-2015)		Out-of-sample (2016)	
	R2	rho	R2	rho
OLS	0.11	0.40	0.06	0.34
Lasso	0.11	0.40	0.06	0.34
Ridge	0.11	0.40	0.06	0.34
Enet	0.11	0.40	0.06	0.34
PCR	0.10	0.38	0.06	0.33
PLS	0.10	0.38	0.06	0.33
Tree	0.21	0.69	-0.60	0.54
Forest	0.56	0.80	0.44	0.72
GBRT	0.45	0.65	0.34	0.61
NN1	0.30	0.66	0.23	0.65
NN2	0.32	0.67	0.30	0.67
NN3	0.38	0.72	0.32	0.71
NN5	0.32	0.70	0.27	0.69
NN10	0.31	0.68	0.27	0.68

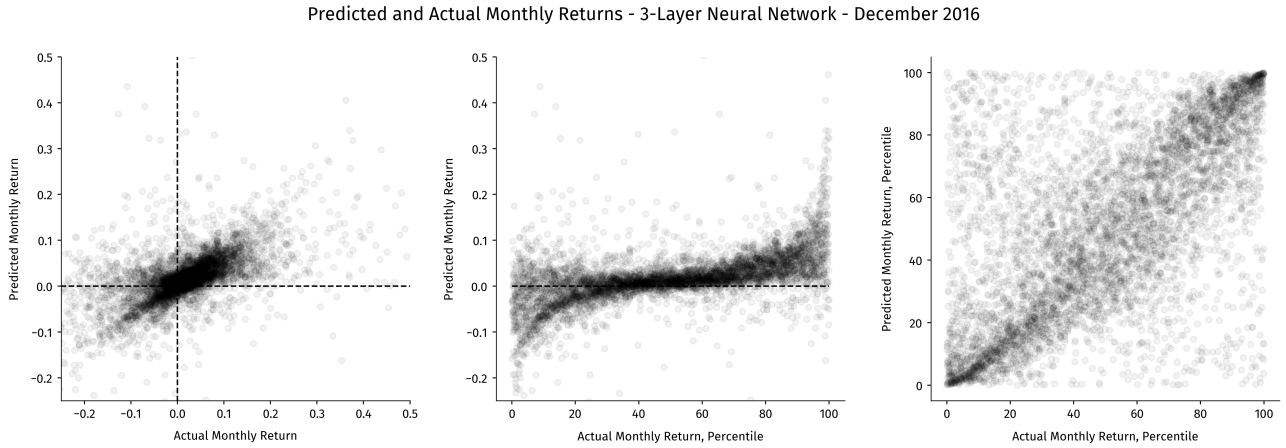
The best out-of-sample R^2 is achieved by the random forest, which reaches 0.44, while both Random Forests and 3-layer Neural Networks perform very well according to Spearman's rho, reaching 0.72 and 0.72 respectively. The worst scores is achieved by OLS and its penalized analogues Lasso, Ridge and Enet. Overall, we can see that all methods perform quite well, in particular on the rank-association score, but that the explanatory power of non-linear methods is extremely high. This dominance holds if we look at the scores of other dates, but explanatory power is sometimes generally lower.

To understand why non-linear methods perform better, we can try to directly analyse the cross-section of returns and of predicted returns. Figure 5 shows three ways of plotting predicted returns against actual returns. The first, on the right, simply shows predicted returns against actual returns in a scatter plot. Points are black but very transparent, so shades of grey can be interpreted as densities of points: quite clearly, there is a positive relationship, but it is difficult to see how well this holds within the deeply black area. On the right, we have transformed both measures into percentiles, i.e. we plot a stock's predicted return's percentile among predicted returns against its true return's percentile among true returns. We can now more clearly see the positive association.

Finally, the graph in the middle plot predicted returns against percentiles of actual returns, and it's this presentation we will be focussing on because, in our view, it is the best method to represent the variance of predicted returns. We can now try to see how this graph would look like for other prediction methods, which we show in Figure 6. Each panel represents return predictions for OLS, PCR, Random Forest and 3-Layer Neural Networks. As before, transparent black points represent individual observations of stocks and the black line shows the average predicted return. Blue dots show the average predicted return within twenty bins of actual returns, and can thus be seen as a local average of the cloud transparent black points. Finally, the dashed orange line shows the actual monthly returns: it is strictly increasing because we are plotting returns against percentiles of returns, which is of course a monotonic relationship.

Viewed differently, the orange line is where all points should be if the Machine Learning method perfectly predicted each stocks return. How scattered the points around the orange line are shows how much noise this predictions has. Moreover, for methods that do not work well, the observations will tend to group around the average of predictions, i.e. the black line, because the ML method will

Figure 5: Representing the Cross-section of Predicted and Actual Returns



not have learned to explain the variance of true returns. Thus, Figure 6 very clearly shows why non-linear methods, represented here by OLS and PCR, scored so poorly: their predictions are extremely compressed, which means they have not "learned" to use the available information to predict the difference between returns.

On the other hand, Random Forests and 3-layer Neural Networks forecast the variance of returns remarkably well, as can be seen by the blue dots (measuring predictions) that closely follow the orange line (measuring returns). We can also note that Random Forests in fact performs slightly better than Neural Networks, which can be seen on the graph but also in their R^2 scores (0.45 against 0.17). However, since there is still a positive relationship between predicted and actual returns in both cases, they perform similarly well in terms of Spearman ρ (0.73 against 0.62).

To conclude this section, we can also study the transposed version of this graph, which can be seen in Figure A3 from the annex. There again, Random Forests and Neural Networks perform very well, while OLS and PCR do not. In this form, the returns (black dots) are very scattered around the average predicted return, but the predicted returns (dashed blue line) are very close to it. Moreover, the orange dots, representing means of actual returns binned by vigintiles of predicted returns, show that the relationship between both is far from monotonous: this is why both R^2 and ρ score quite poorly for both methods.

6.2. Time series analysis

Now that we have seen that, in general, non-linear methods like trees, forests and Neural Networks vastly outperform linear methods, we can try to assess whether this explanatory power varies over time. Figure 7 shows the four combinations of scores, in-sample (test) and out-of-sample (oos), $R2$ and Spearman ρ. Since models are trained annually, scores are computed annualy: the exact annual scores are shown in pale. Above them, we have shown rolling, centered five-year averages to look beyond sharp variations between years and focus on longer trends.

Unsurprisingly, we can observe that in-sample scores are much higher than out-of-sample scores. Additionally, we can see that that in-sample scores are generally quite stable, while out-of-sample scores are less stable. This suggests that there is a part of overfitting in the in-sample learning, that performs poorly out-of-sample in some years and very poorly in others. Moreover, we note that Spearman rho's are more stable for both methods, which supports our earlier view that it is indeed a more appropriate measure of fit.

Comparing both, we can see that, again, Neural Networks perform better than Ordinary Least Squares. However, they experience an important breakdown in R^2 in the mid-2000s that is not

Figure 6: Cross-section of Predictions for different Machine Learning Algorithms

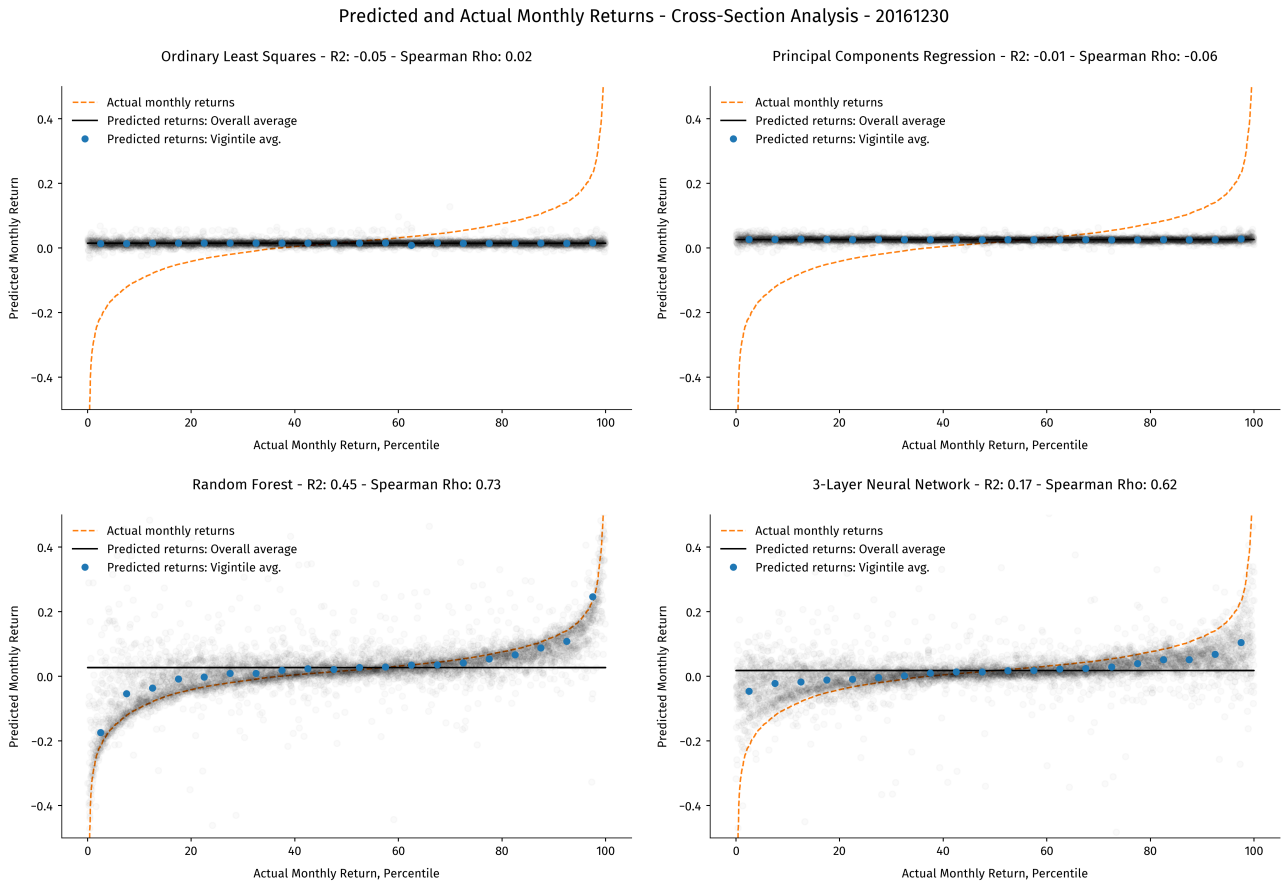
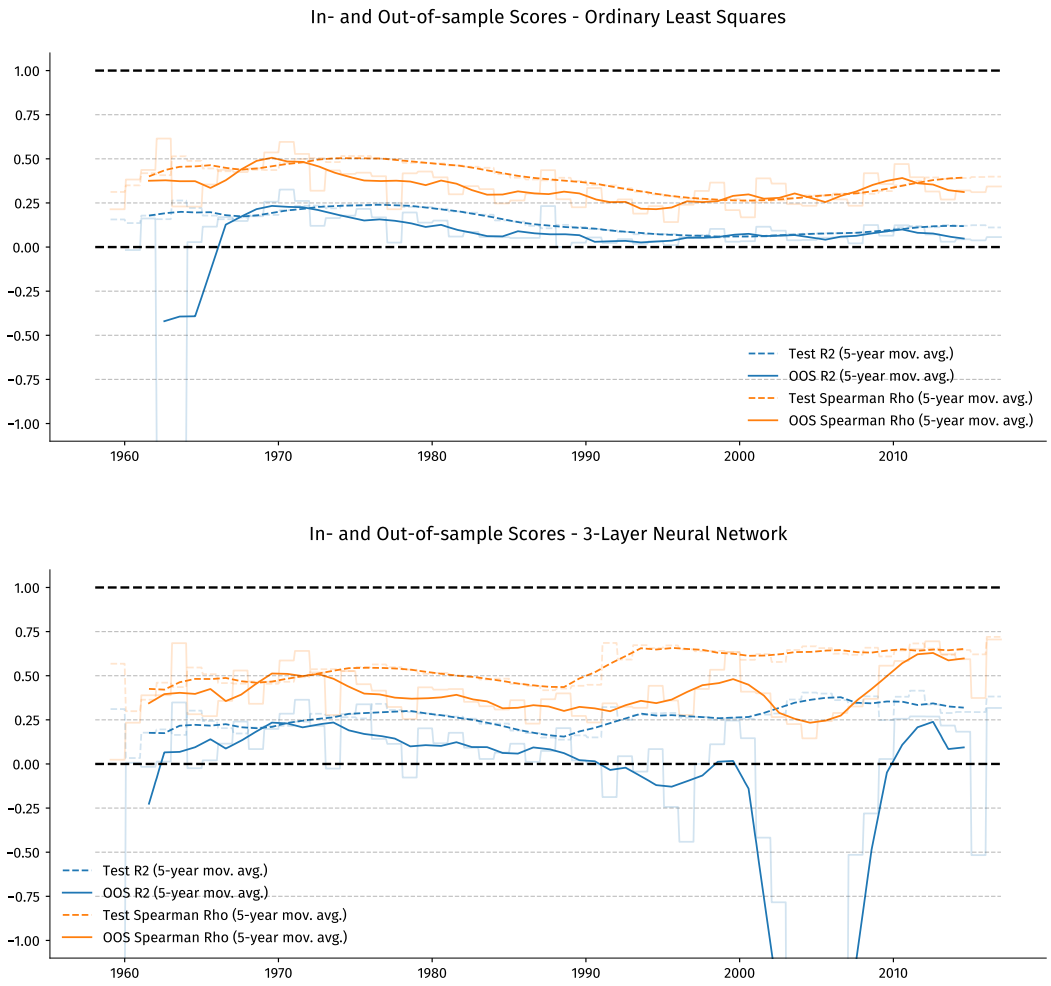


Figure 7: Time Series Analysis of In- and Out-of-sample Scores



matched by OLS and that is not accompanied by a change in Spearman ρ: this suggests that there are changes in markets or in the dataset that cause the mean expectation, as well as the size of errors, to diverge, while the pure ranking performance of the model stays relatively equal.

Figure A4 shows the same analysis for PCR and Random Forest. In particular, it shows a similar break-down in R^2 in the mid-2000s, but also one in the 1980s. However, here as well, the decrease in Spearman ρ is much smaller, which suggests that numerical predictions become imprecise but that the ranking performance remains robust. However, the exact cause of these swings remains unclear. Interestingly, they are not really reflected in the performance of portfolios built on the basis of these predictions.

6.3. The Role of Non-Linearity

What drives this superiority of more complex Machine Learning methods over traditional econometric tools? In our view, which is also that of Gu et al. (2020), it is the non-linearity of their estimators, which in the context of multivariate analysis has two decisive implications: (i) effects of an explanatory variable on the explained variable is not constant: for example, the size effect could be larger between small- and mid-cap firms than between mid- and small-cap firms ; (ii) the effect of a variable on the prediction can vary with the values of another one: for example, the size effect could be larger for firms with higher market beta. This second aspect, the interaction effect, can in fact involve the values of all explanatory variables included in our model.

This ability to detect complex patterns involving multiple variables is the key difference between complex methods, like Regression Trees, Random Forests and Neural Networks, and OLS, even though the former are in fact very different. We assess the contribution of non-linearities by studying the predictors these methods give out. We focus on the last predictor of our rolling forecasts, which is trained on data from 2005 to 2015, and tries to predict returns in 2016 out of sample.

More precisely, we try to analyse the direct effect as well as the interactions of four of the most famous return predictors, namely market beta, market capitalization, book-to-market ratio and six-month return momentum. Figure 8 plots the predicted return for an Ordinary Least Squares estimator and for a 3-Layer Neural Network, both trained over 2005-2015. Since all factors are standardized before the model is trained, we plot predictions for a standardized span of values between -2 and +2 standard deviations. [1] These correspond to the same factor in each column of the graph.

To show interaction effects, we also show, for each variable, the predicted return for different values of the other four variables: the shaded lines correspond to values of -2, -1, +0, +1 and +2 standard deviations around the mean. These values correspond to the same factor in each row of the graph. Of course, the diagonal graphs only show the direct effect of the variable on predicted return.

Two things can immediately be noted about the OLS plot, which mirror the two limitations of noted above: in the OLS plot, all lines are straight because the predictor is linear in each variable; moreover all interaction lines are parallel, because the predictor is also linear in all combinations of variables. This means that the marginal effect of one variable on the predicted return is always the same, whatever this variable's value is, and whatever all other variable's values are.

On the other hand, in the plot about the 3-Layer Neural Network, none of the direct effect lines are straight: the model has learned to predict that returns are a fairly complex, non-linear function of the input variables. Additionally, the interaction lines are never exactly parallel: the model has learned to predict that the effect of one variable on returns depends on the values of other variables, according to a fairly complex pattern.

[1] This procedure causes some problems when plotting the effect of log market value (*mvel1*): the standardized version of this variable is extremely skewed upwards, and the lowest value actually observed is only -0.28 stds. This means that the predictions below this value are essentially meaningless, and reflect imperfect learning at the very bottom of the market cap distribution: non-linear methods seem to show an absurdly strong size effect, but this effect becomes much more reasonable when one looks at the values predicted for values of *mvel1* that are actually observed.

Figure A5 in the Annex shows the same plots for the Principal Components Regression and for Regression Trees. The graph for PCR is not entirely fascinating, because just like OLS it is a linear combination of input variables, although this is the product of two linear stages (linear projection onto a smaller space, linear projection onto the explained variable). Random Forests on the other hand show a much more involved pattern, corresponding to the combination of regression trees that it represents. The interaction effects also take the form of step functions. Interestingly, the general direction of effects is sometimes at odds with that of the other models, in particular for beta, whose effect is foreseen as negative. This is perhaps due to the inclusion of many highly correlated factors (e.g. beta square), that make the direct effect of each variable less intuitive.

This analysis of predictors clearly shows that the key difference between simple ML methods that generally perform quite poorly, and more complex ML methods that perform quite well, is in the non-linearity of their predictors and in the interaction effects they flexibly capture. What remains to be seen, however, is which factors contribute most to the explanatory power of which methods.

6.4. Factor Importance and the Usefulness of Marginal Factors

While it has many drawbacks, OLS surpasses all other methods in one decisive respect: there is an immense literature and numerous sound theoretical results allowing us to identify which parameters make a statistically significant contribution to the prediction. For most other methods, these are cruelly lacking. Since the causal interpretation of Machine Learning methods is an extensive area of ongoing research, we chose to follow the simple but robust approach adopted by Gu et al. (2020).

To quantify the contribution of a variable to the explanatory power of the model, we change all its values to zero (recall that all variables are standardised) in the sample, and compute the change in R^2 and in Spearman ratio induced by such a masking. If the performance scores decrease a lot, the variable contributes a lot to the model's explanatory power. However if they change very little, it means the variable contributes very little. In fact, there are a few cases where the performance scores increase very slightly once a variable is masked, which is to be expected if the variable has no explanatory power and only adds variance to the forecast.

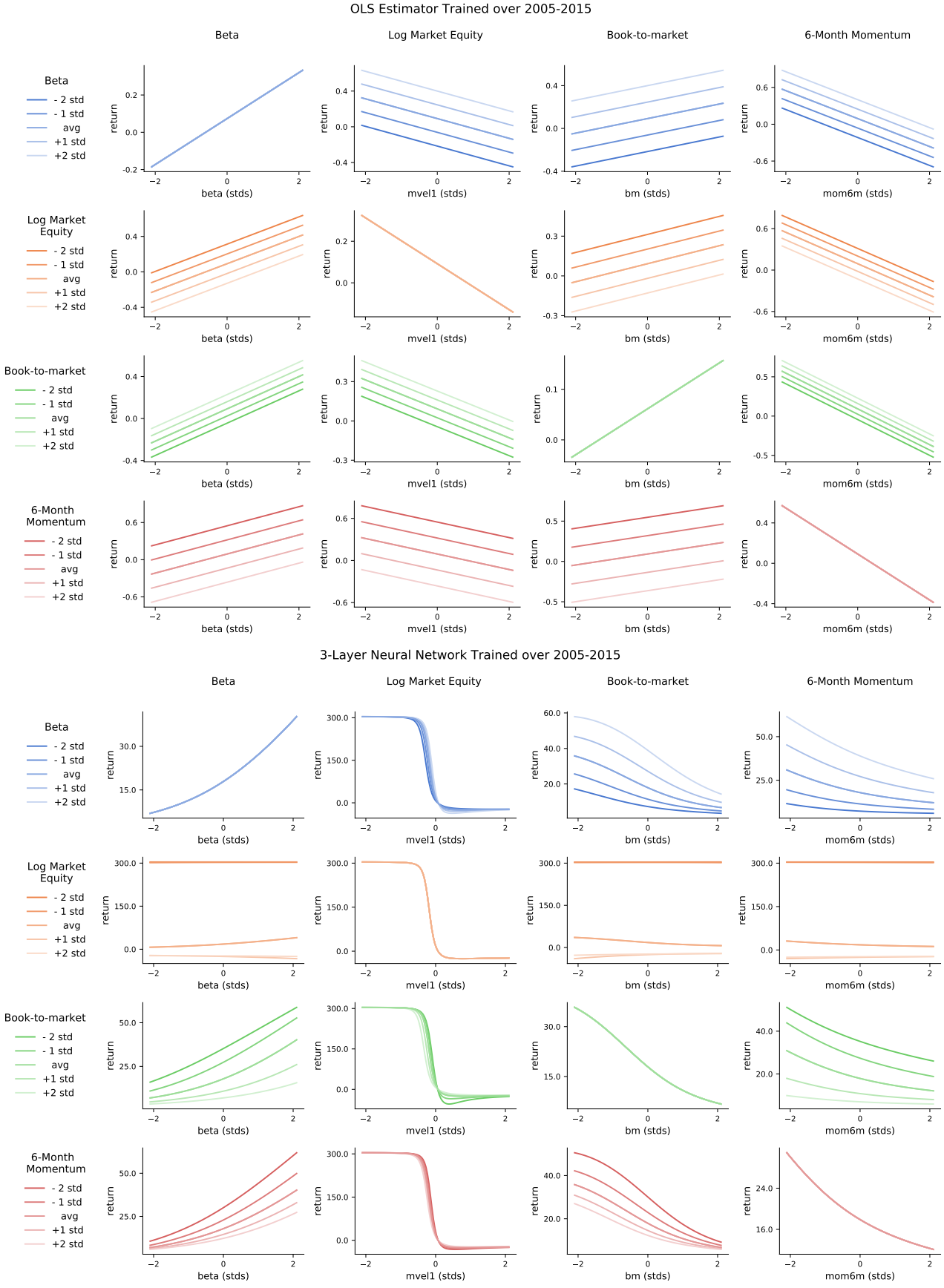
Figure 9 shows the percentage decrease in R^2 through a colour scale going from light for a small decrease to dark for a strong decrease. Factors are then sorted according to their average contribution across models. We can see that most models seem to agree on which factors decisively explain the cross-section of stock returns. However, the non-linear methods ranging from Regression Trees to 10-Layer Neural Network seem to give many variables explanatory power, while OLS and related estimators are only affected by a few of them. The ability of these more complex methods to seize on information contained in seemingly less relevant factors seems to be another reason for their exceptional performance.

Not very surprisingly, the variables that best explain the cross-section of stock returns are mainly macro-level variables that are the same across factors: measures of the risk-free rate (*rfree*, *rf*, *tbill*) and of the market's excess return (*mkt-rf*, *crsp-spvw*, *crsp-spvwx*) explain the cross-section of returns in the training sample, because it compresses observations from different periods in a single, undifferentiated dataset. The factor model portfolio returns, *smb*, *mom*, *rmw* and to a lesser extent *cma* and *hml* also feature prominently.

On the other hand, the stock-specific factors that explain the cross-section of returns are size (*mktcap*, i.e. market capitalization, and *mvel1*, i.e. log market capitalization), momentum (*mom1m*, *mom6m*, *mom12m*, corresponding to different horizons), beta (and also beta squared). More surprising factors, like dollar trading volume (*dolvol*), return volatility (*retvol*) and industry momentum (*indom*).

For completeness, we report the same result for the Spearman ρ in Figure A6 in the Annex: conclusions are broadly the same, althought the explanatory contribution seems a bit less spread across factors. In Figures A7 and A8 we show the same table for test scores, and report out-of-

Figure 8: Non-Linearities and Interaction Effects in Machine Learning Predictors



sample scores in Figures A9 and A10. Since out-of-sample forecasting is much more variable, far fewer variables cause a substantial decrease in R^2 when they are masked.

Additionally, the apparent consensus between methods about which factors matter seems to break down out of sample: many factors matter a lot in some models but not at all in others. The importance of variables changes (all tables are re-sorted), although macro-level factors still reign supreme. In our view, this shows that Machine Learning methods are not *magic*: they all overfit in-sample, and behave very heterogeneously out-of-sample. In short, more advanced ML methods allow us to better forecast the cross-section of returns, both in and out of sample, but they fall far short from definitively learning a true, stable and coherent asset pricing model.

If more advanced ML methods perform much better than traditional ones, and if the explanatory power of many factors seems to be quite low, one is naturally led to ask : would advanced methods using less factors still outperform their traditional peers? Put differently, what is the marginal explanatory power of an additional factor, and how does it compare to the marginal power of a better ML method? Clearly, this line of questioning also leads us to evaluate the overall usefulness of the algorithmic market efficiency concept: if the marginal benefit of information is much higher than that of algorithms, it is doomed to be much less of a driving force than conventional informational market efficiency.

To study this question, we re-train our algorithms on a subset of factors. We randomly assemble datasets containing only 11% of factors (i.e. 11 factors), 25% (29), 50% (58), 75% (87) and 90% (105). In fact, we do this 10 times for each subset, and then average scores across splits to estimate marginal contributions more precisely. Resulting R^2's and Spearman rho's, both for the train and test samples (2005-2015) and out-of-sample (2016) are presented in Figure 10.

Results are remarkably stark: while all methods' scores increase with the number of factors, advanced methods like Tree-based regressors and Neural Networks generally perform better with few factors than OLS with all factors. In short, in our narrow setting, the algorithmic margin seems to be much more important than the informational margin. Moreover, the explanatory power of additional factors analysed by OLS flattens out fairly quickly, which indicates that the marginal utility of academic research identifying new pricing factors is actually quite low. If the goal is to explain the cross-section of returns, it seems that much more research should be devoted to finding new ML methods to forecast predicted returns using the already discovered factors, rather than discovering new ones.

Even worse, this flattening out is much less pronounced for more advanced methods, where even though the baseline level is high, additional factors seem to steadily improve the explanatory power of the model. This can probably be explained by the interaction effects that are better captured by these newer methods, and that seem to make out a sizeable share of return predictability. Research in asset pricing should thus generally focus more on prediction methods, but even research on asset pricing factors should take them into account since they seem to be much better than traditional OLS at recognizing the explanatory usefulness of additional factors.

More broadly, this finding also supports the conceptual importance of algorithmic market efficiency: since better algorithms seem more 'useful' than better information, the reaction of market prices to the discovery of new ML methods should be also be much more pronounced than to the discovery of new factors.

Figure 9: Percentage Decrease in Train R^2 Induced by the Masking of each Factor

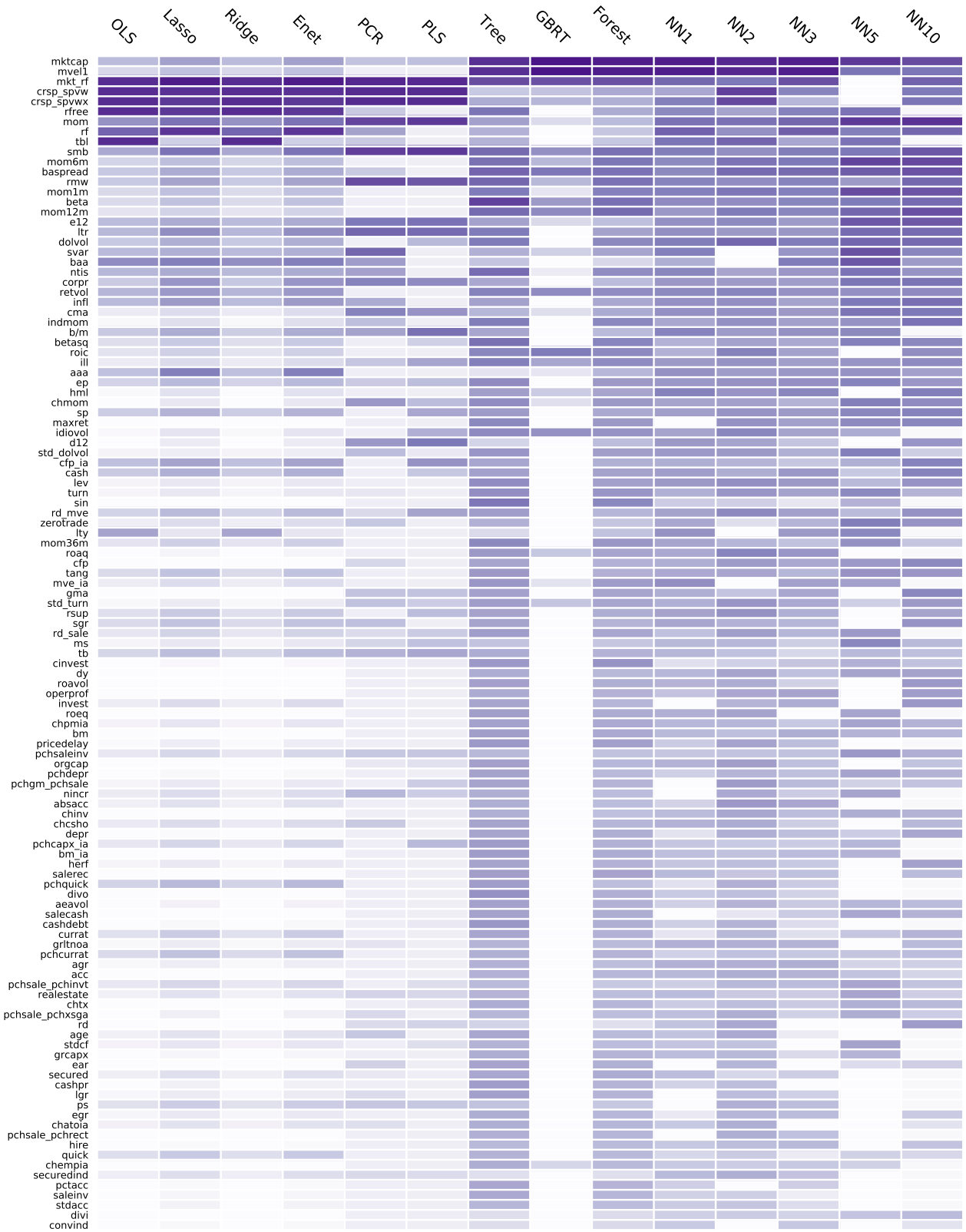
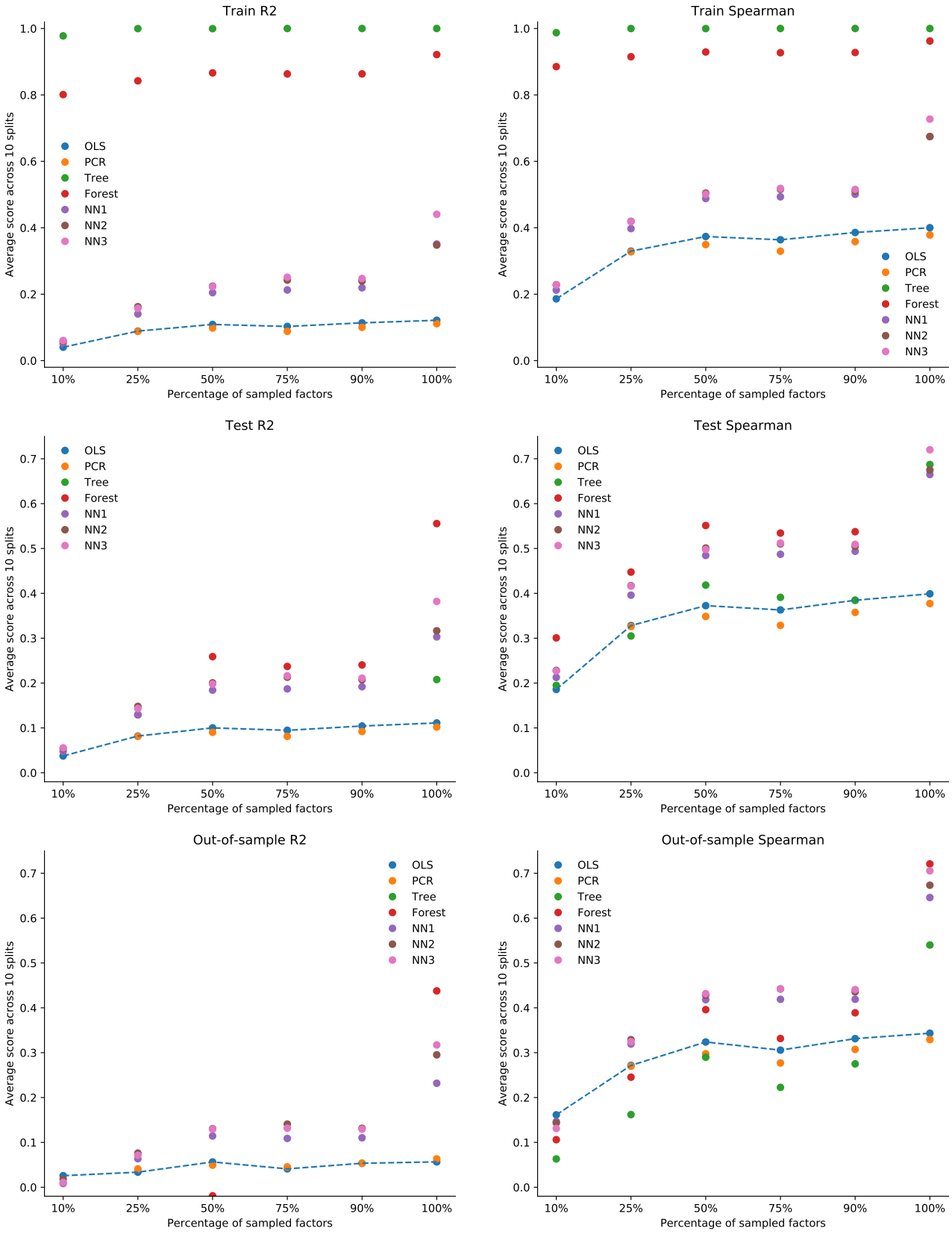


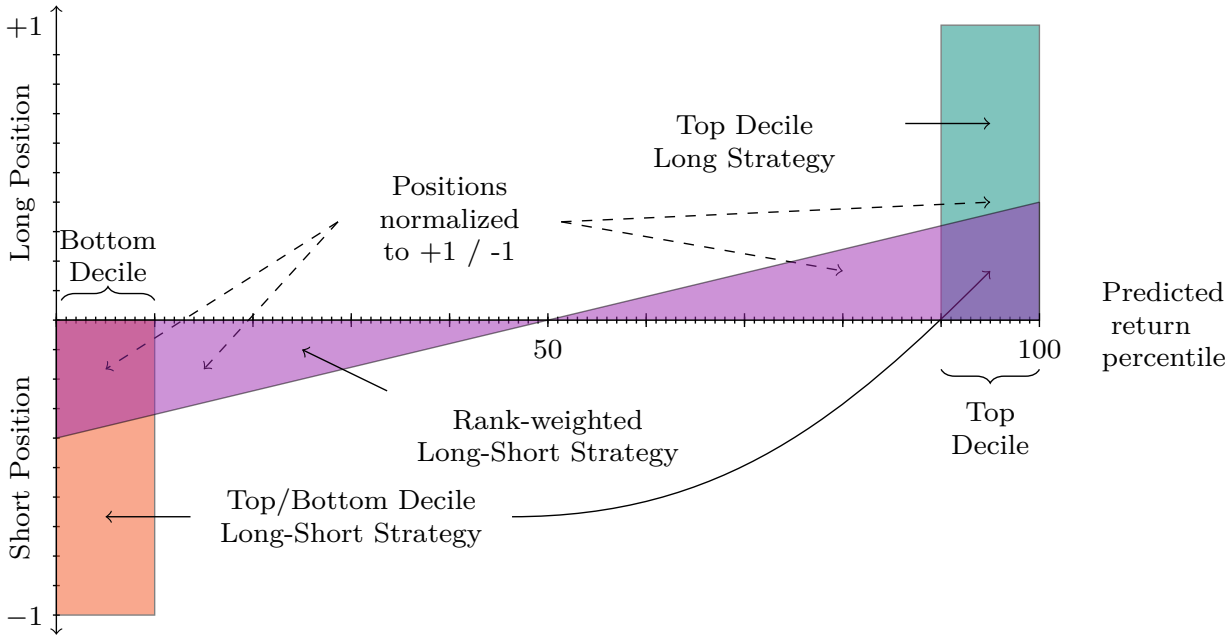
Figure 10: Marginal Contribution of Factors and of Machine Learning Methods



7. Performance of Machine Learning Portfolios

After analysing their purely statistical performance, we will try to assess whether this predictive power translates into portfolios that can outperform the market. While many different strategies could have been tested, we restrict ourselves to three : the top decile long strategy, the top/bottom decile long/short strategy and the rank-weighted strategy. We compute each strategy using each method's predictions, and then compute a number of performance metrics for all combinations. The subsequent briefly explains how each is set up, and then describes its financial performance.

Figure 11: Illustration of the Three Tested Strategies



7.1. Top Decile Long Strategy

The *top decile long strategy* simply buys the 10% of stocks whose return is predicted to be highest among all stocks. The natural counterparty to this strategy is simply the overall stock market: if Machine Learning algorithms were blind, i.e. had zero predictive power, such a strategy would have an expected return that converges to that of the stock market. Indeed it should be underscored that, since this strategy does not have a zero initial capital (we have implicitly normalized it to 1$ on January 1st, 1958), EMH does not imply that this strategy should have a zero risk-adjusted return.

Performance of the strategy is summarized in Table 4 as well as in Figure 12. The table reports the average return of a long strategy, as well as its Sharpe ratio: both serve as very basic measures of portfolio performance, but do not account for risk factor loading. To test whether this overperformance remains after adjusting for risks, the table then reports the α of the portfolio as measured by four work-horse asset pricing models: the CAPM, the Fama-French 3-factor model, the Carhart 4-factor model and the Fam-French 5-factor model. We compute alphas over the whole sample, and use Newey-West autocorrelation-robust estimators to compute t-statistics.

Equally-weighed returns for all methods vastly outperform the market average, and value-weighted returns outperform the market average for all methods but PLS. Even very basic methods like OLS and its regularized variations outperform the market average, with average yearly returns of 27% against 19% for the market average. Moreover, alpha is significant in all specifications. This implies

that the factor dataset includes sufficiently strong information about returns and/or risk loadings so that simple, non-linear methods already detect them.

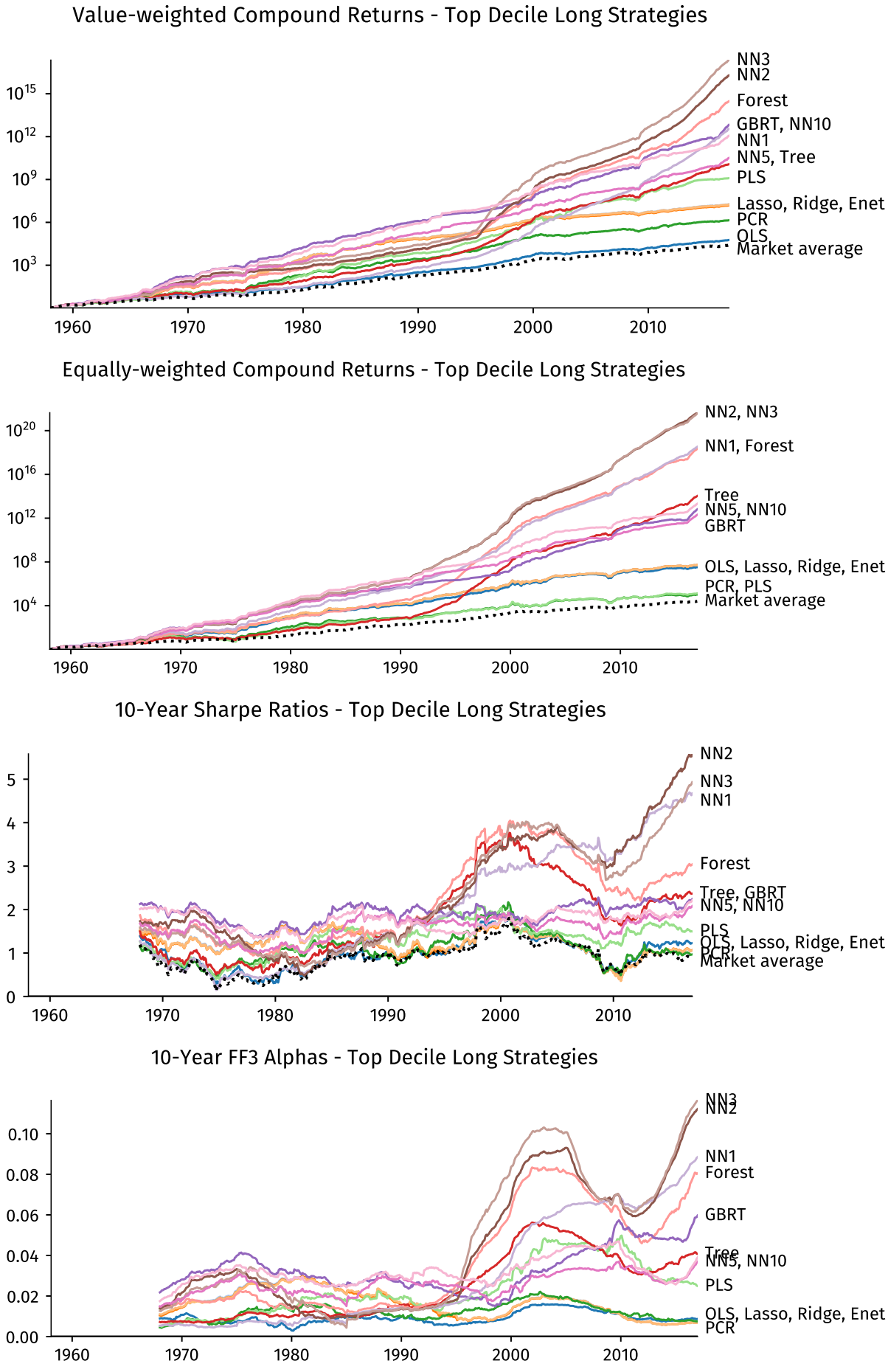
However, more complex methods perform much better. The best-performing methods seem to be the Random Forest, which achieves an average annual return of 165% over 1958-2016, with a Sharpe ratio of 1.87 and an annual α of 94% in the CAPM and of 102% in the Fama-French three factor model. Neural Networks one, two or three layers also perform remarkably well, with e.g. a 151% average return, 1.82 Sharpe ratio, 87% α in the CAPM and 93% for FF3. The comparatively subpar performance of Neural Networks with many layers is a finding similar to those of Gu, Kelly, and Xiu (2020), probably indicative of the high signal-to-noise ratio of financial data.

Table 3: Top Decile Long Strategies - Performance of Machine Learning Portfolios over 1958-2016

	1958-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	1.56	1.03	45	-19.0	-32.0	0.80	(9.15)	0.88	(7.89)	0.84	(8.41)	0.88	(6.73)
Lasso	2.39	1.21	104	-20.2	-33.0	1.56	(4.20)	1.51	(5.77)	1.42	(5.65)	1.47	(5.36)
Ridge	2.36	1.20	104	-20.2	-32.8	1.54	(4.14)	1.49	(5.71)	1.40	(5.62)	1.44	(5.30)
Enet	2.38	1.21	104	-20.2	-33.1	1.56	(4.23)	1.51	(5.86)	1.42	(5.73)	1.46	(5.44)
PCR	2.02	1.06	89	-22.0	-35.1	1.18	(6.02)	1.11	(7.10)	1.02	(6.15)	1.07	(5.69)
PLS	3.00	1.26	89	-30.4	-46.4	2.18	(4.78)	2.16	(4.23)	2.24	(3.44)	2.50	(4.18)
Tree	3.33	1.78	162	-21.8	-32.4	2.49	(3.92)	2.45	(3.75)	2.54	(3.91)	2.56	(3.84)
Forest	4.81	2.15	151	-18.0	-27.3	4.02	(3.73)	3.94	(3.55)	4.03	(3.60)	4.11	(3.60)
GBRT	4.24	1.89	143	-11.9	-27.9	3.44	(6.66)	3.27	(7.14)	3.60	(6.96)	3.36	(6.96)
NN1	4.15	2.05	79	-16.7	-30.1	3.36	(2.82)	3.35	(2.80)	3.52	(2.90)	3.45	(2.85)
NN2	5.43	2.40	101	-16.2	-32.1	4.69	(3.28)	4.73	(3.30)	4.83	(3.34)	4.85	(3.33)
NN3	5.80	2.26	109	-16.1	-29.1	5.04	(3.39)	4.94	(3.17)	5.20	(3.36)	5.02	(3.14)
NN5	3.47	1.61	108	-14.7	-26.0	2.64	(10.78)	2.51	(9.81)	2.74	(9.81)	2.54	(8.64)
NN10	3.99	1.79	127	-21.2	-33.3	3.16	(11.15)	3.09	(10.49)	3.25	(10.89)	3.17	(9.92)

Average returns, maximum losses, maximum drawdown and alphas are given in monthly percentages, while Sharpe ratios are annualized. t-statistics are computed using a Newey-West autocorrelation-robust estimator for standard errors, allowing for up to $10 \times 12 = 120$ lags. FF3 refers to the Fama-French three-factor model, CH4 to the Charhart four-factor model and FF5 to the Fama-French five-factor model.

Figure 12: Performance of Top Decile Long Strategies



7.2. Top/Bottom Decile Long/Short Strategy

The *top/bottom decile long/short strategy* buys the 10% of stocks whose return is predicted to be highest while going short the 10% of stocks whose return is predicted to be lowest. In our computations, we normalize the strategy so that it goes long 1$ and short 1$ in 1958. Since this strategy does not require any initial investment, its risk-adjusted return should be zero under the Efficient Market Hypothesis.

As can be observed quite clearly in Table 4 and Figure 13, this is far from being true: while some methods, like PCR and PLS, do indeed yield negative returns, most methods have strikingly high average monthly returns. This is particularly true for non-linear methods, like Regression Trees, Random Forests and 1-, 2- and 3-Layer Neural Networks. 1-Layer Neural Networks report the highest returns, with an average monthly return of 6.27%. Because of the high volatility of the strategy, the Sharpe ratios are again somewhat less impressive, but again 1-Layer Neural Networks obtain the highest scores with 2.17.

Additionally, we can see that turnover is quite high, with up to 168% monthly turnover in the case of GBRT, with most strategies reaching turnover values in the 100%-130% range. These strategies do remain quite risky, as they report quite high maximum losses (e.g. -30.3% for Random Forests and -22% for 1-Layer Neural Networks). Maximum drawdowns are naturally even more significant, reaching -44.9% for Random Forests, -41.2% for 1-Layer Networks but also a worrying -93.8% for the 3-layer version.

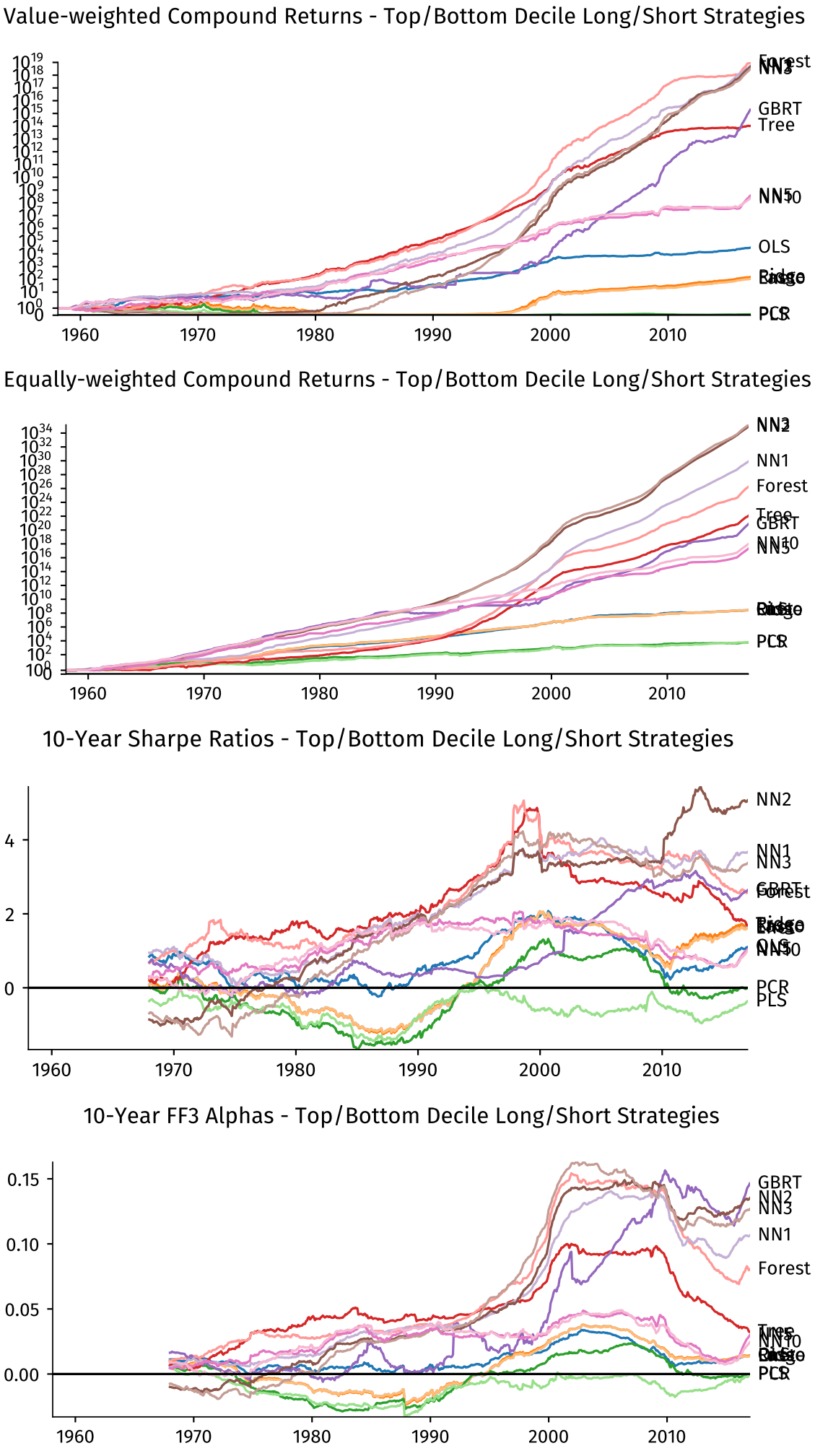
Most Machine Learning strategies nevertheless translate into positive alpha, e.g. 6.51% for Random Forests and 6.22% for 1-Layer Neural Networks in the FF5 Model. Moreover, most t-statistics are significant, in particular those for non-linear methods.

Table 4: Top/Bottom Decile Long/Short Strategies - Performance of ML Portfolios over 1958-2016

	1958-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	1.46	0.84	62	-21.8	-36.2	1.08	(3.32)	1.30	(3.47)	1.11	(3.87)	1.33	(3.19)
Lasso	0.67	0.26	106	-25.2	-98.2	0.68	(1.19)	0.86	(1.29)	0.80	(1.38)	1.26	(1.81)
Ridge	0.71	0.29	107	-25.3	-98.3	0.72	(1.24)	0.89	(1.33)	0.83	(1.44)	1.29	(1.84)
Enet	0.66	0.26	106	-25.1	-98.2	0.68	(1.19)	0.85	(1.29)	0.78	(1.36)	1.26	(1.81)
PCR	-0.30	-0.21	84	-40.1	-99.7	-0.18	(-0.37)	-0.03	(-0.06)	-0.08	(-0.16)	0.37	(0.61)
PLS	-1.30	-0.57	65	-32.5	-100.0	-0.63	(-1.87)	-0.63	(-1.73)	-0.91	(-2.58)	-0.65	(-1.87)
Tree	4.67	1.88	164	-21.1	-36.2	4.20	(3.71)	4.49	(3.90)	4.59	(4.04)	4.81	(4.25)
Forest	6.36	2.11	150	-30.3	-44.9	5.87	(3.17)	6.19	(3.31)	6.24	(3.38)	6.51	(3.48)
GBRT	5.07	1.01	168	-30.0	-77.0	5.28	(2.49)	5.32	(2.56)	5.75	(2.45)	5.76	(2.66)
NN1	6.27	2.17	85	-22.0	-41.2	5.66	(2.97)	6.02	(3.18)	6.18	(3.22)	6.22	(3.20)
NN2	6.24	1.97	101	-24.7	-87.3	5.84	(2.42)	6.16	(2.57)	6.07	(2.55)	6.41	(2.66)
NN3	6.18	1.84	108	-25.0	-93.8	5.81	(2.30)	6.07	(2.45)	6.39	(2.53)	6.33	(2.49)
NN5	2.80	1.09	126	-29.4	-37.0	2.24	(4.52)	2.43	(4.32)	2.62	(5.40)	2.58	(4.71)
NN10	2.74	1.13	131	-32.8	-41.4	2.20	(4.09)	2.42	(4.06)	2.49	(4.39)	2.52	(4.60)

Average returns, maximum losses, maximum drawdown and alphas are given in monthly percentages, while Sharpe ratios are annualized. t-statistics are computed using a Newey-West autocorrelation-robust estimator for standard errors, allowing for up to $10 \times 12 = 120$ lags. FF3 refers to the Fama-French three-factor model, CH4 to the Charhart four-factor model and FF5 to the Fama-French five-factor model.

Figure 13: Performance of Top/Bottom Decile Long/Short Strategy



7.3. Rank-weighted Long/Short Strategies

Like the previous one, this strategy is a long-short strategy, but it takes a position on the whole spectrum of stocks although this time the weight placed on each stock varies with a stock's predicted return rank. More precisely, the portfolio weight of stock i at time t is equal to

$$w_{t,i} = \left[rk(\widehat{r}_{i,t}) - \frac{n_t + 1}{2} \right] \times W \tag{17}$$

where $rank_i$ is stock i's predicted return rank among all others stocks at time t, and n_t is the number of stocks that are traded at date t. W is a normalization factor, which we have chosen so as to set the total exposure on the long side of the portfolio equal to 1$ in 1958. Since both are symmetric, this also implies that the short side will be short 1$ in 1958: this set-up makes the rank-weighed and top-bottom-decile strategies directly comparable.

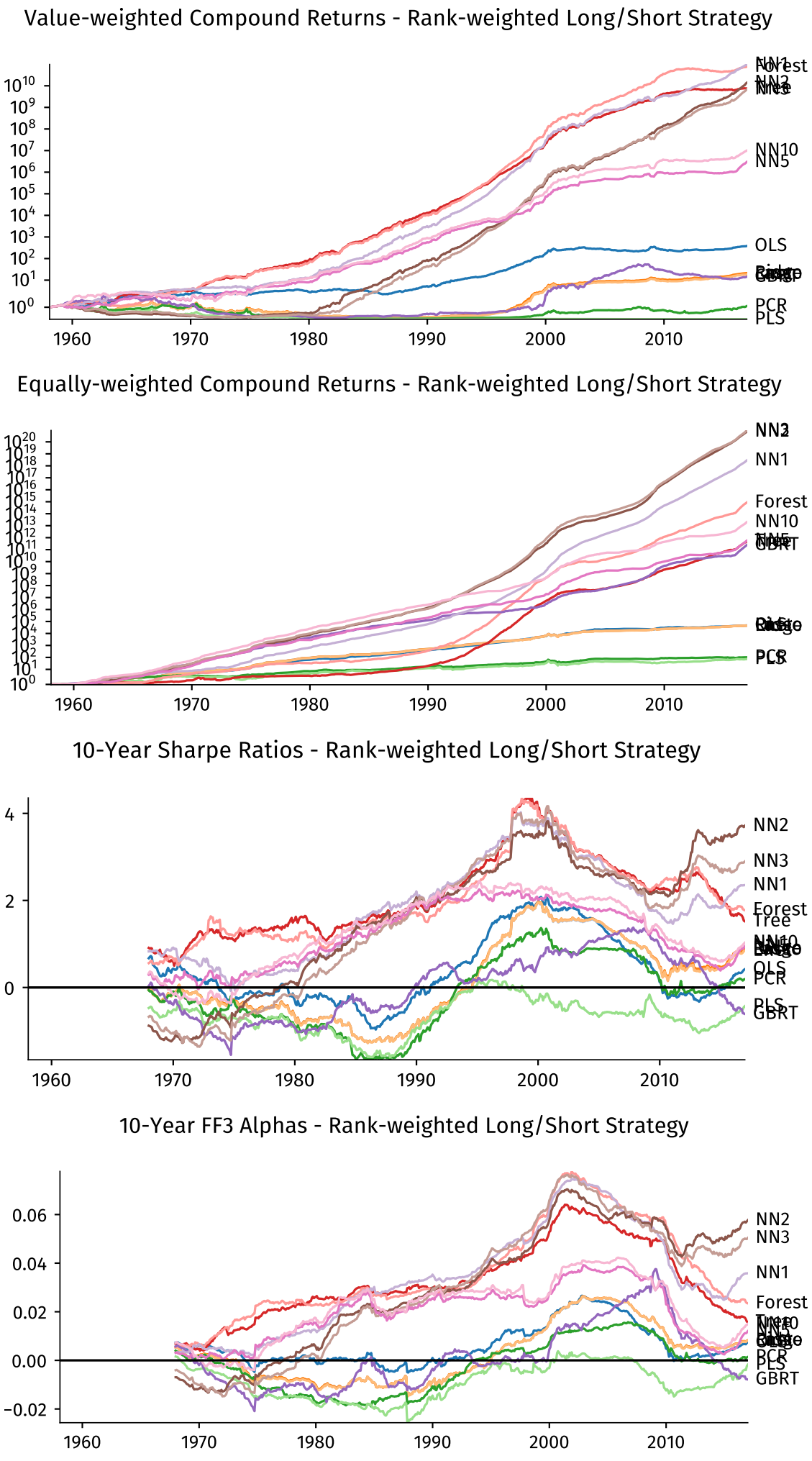
As can be noted in table 5 and Figure 14, Machine Learning methods outperform the market very substantially. In comparison to the decile spread strategy, we can note that average monthly returns are usually lower, but that Sharpe ratio are in the same range. Again, most strategies generate significant alpha under all specifications, for example 4.25% per month for Random Forests and 4.10% per month for 1-Layer Neural Networks in the FF5 model.

Table 5: Rank-weighted Long/Short Strategies - Performance of ML Portfolios over 1958-2016

	1958-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	0.84	0.47	52	-11.4	-44.6	0.56	(1.82)	0.76	(2.18)	0.60	(2.27)	0.83	(2.16)
Lasso	0.42	0.11	67	-20.0	-91.7	0.31	(0.81)	0.51	(1.13)	0.41	(1.05)	0.79	(1.60)
Ridge	0.43	0.12	67	-20.1	-91.6	0.32	(0.82)	0.52	(1.14)	0.41	(1.07)	0.79	(1.60)
Enet	0.41	0.11	67	-20.0	-92.1	0.31	(0.78)	0.50	(1.11)	0.40	(1.01)	0.78	(1.57)
PCR	0.01	-0.16	52	-32.4	-97.3	-0.11	(-0.34)	0.01	(0.04)	-0.01	(-0.03)	0.35	(0.82)
PLS	-0.90	-0.63	37	-27.0	-99.9	-0.62	(-2.20)	-0.52	(-1.65)	-0.76	(-2.55)	-0.46	(-1.53)
Tree	3.27	1.86	100	-10.9	-23.1	2.63	(3.86)	2.80	(4.05)	2.86	(4.26)	2.89	(4.24)
Forest	3.60	1.79	85	-27.0	-35.4	2.93	(3.51)	3.15	(3.73)	3.13	(3.84)	3.27	(3.92)
GBRT	0.37	0.10	95	-19.0	-94.8	-0.04	(-0.08)	0.11	(0.22)	0.15	(0.30)	0.43	(0.74)
NN1	3.64	1.79	63	-21.0	-37.1	2.93	(3.39)	3.19	(3.67)	3.19	(3.79)	3.36	(3.74)
NN2	3.35	1.59	65	-18.9	-87.2	2.74	(2.52)	2.95	(2.76)	2.96	(2.78)	3.11	(2.89)
NN3	3.24	1.52	70	-19.8	-86.7	2.67	(2.38)	2.89	(2.65)	2.93	(2.66)	3.09	(2.78)
NN5	2.12	1.17	83	-18.1	-28.6	1.56	(3.52)	1.75	(3.51)	1.80	(3.93)	1.84	(3.97)
NN10	2.30	1.24	90	-20.4	-44.4	1.77	(3.52)	1.94	(3.60)	2.00	(3.95)	2.06	(3.93)

Average returns, maximum losses, maximum drawdown and alphas are given in monthly percentages, while Sharpe ratios are annualized. t-statistics are computed using a Newey-West autocorrelation-robust estimator for standard errors, allowing for up to $10 \times 12 = 120$ lags. FF3 refers to the Fama-French three-factor model, CH4 to the Charhart four-factor model and FF5 to the Fama-French five-factor model.

Figure 14: Performance of Rank-weighted Long/Short Strategies



7.4. Robustness checks on Portfolio Outperformance

We perform supplementary checks on the performance of our Machine Learning Portfolios along three axes: (i) we change the time horizon over which we evaluate our strategies; (ii) we restrict the investment set of available stocks; and (iii) we try to account for transaction costs related to the strategies high turnovers.

In the Annex, we present the same performance metrics for all three strategies, but over the time horizons 1980-2016 and 2000-2016. Overall, most Machine Learning methods perform better over more recent time horizons, in particular the non-linear methods like Regression Trees, Random Forests and Neural Networks. For the top decile long strategy, average monthly returns go up to 7.58% for Random Forests and 9.46% for Regression Trees over 2000-2016.

This effect is even stronger for the decile spread strategy, where average monthly returns reach up to 12.79% for 2-layer Neural Networks using decile spreads over 2000-2016. Interestingly however, this effect is not as strong for the rank-weighted equivalents. All in all, we can thus note that financial performances does not substantially decrease over more recent time horizons. Importantly, virtually all alphas remain positive, high and highly significant. We thus conclude that ML-outperformance is robust to changes in the time horizon, and in particular that is has not decayed over the last few decades.

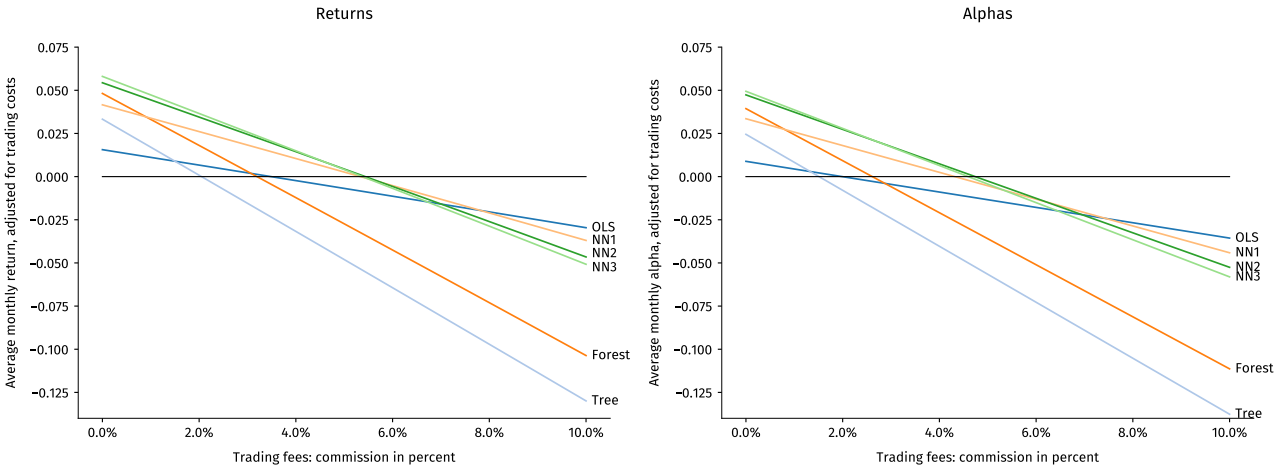
Although our use of value-weighted returns already reduces this risk, we further test whether this outperformance of Machine Learning Portfolios is not due to small, illiquid stocks that could not have been traded in real time without facing prohibitive trading costs. We thus restrict our investment opportunity set to the 1000 largest and 100 largest companies by market capitalization for each year. On average across the full period, these restrictions cover respectively 89% and 49% of total market capitalization. We expect these companies to be highly liquid and therefore to trade.

The associated portfolio performances are represented in Tables A14 through A18 in the Annex. For top decile long strategies, returns are generally lower among Top 1000 companies, but generally higher among Top 100 companies. However, all of them remain high, with positive and highly significant alphas in all specifications. For top/bottom decile long/short portfolios, average returns are roughly the same among Top 1000 companies, but slightly lower among Top 100 companies. For Regression Trees, alphas obtained among Top 1000 companies even become (modestly) insignificant, but for all other methods the presence of positive and highly significant alphas holds. For the rank-weighted strategy, performance is generally slightly higher among Top 1000 strategies but roughly the same for Top 100 strategies, with again positive and significant alphas. In our view, this allows us to conclude that the outperformance of Machine Learning portfolios is robust to changes in the investment opportunity set that account for stock liquidity and small size selection bias.

Finally, we try test whether our outperfomances are robust to the inclusion of transaction costs. To model the implementation shortfall, we use a proportional fee, which makes transaction costs proportional to turnover. Since the turnover of our strategies is generally quite high, we thus expect even low transaction costs to quickly reduce profitability. Figure 15 shows the average monthly return over 1958-2016 of various ML Portfolios for different levels of trading fees: we simply take proportional commissions, where fees are a fraction of the traded volume. The different slopes of different strategies corresponds to different levels of turnover. We can see that the average return becomes negative after 2%-4% for OLS, Trees and Forests, but only after about 5.5% for Neural Networks.

The right subfigure shows the same analysis for alphas, which displays the same trends but (unsurprisingly) becomes negative for smaller transaction costs. Although this test does indeed show that transaction costs can reduce the performance of these portfolios, we note that reasonable levels (around 1%) still generate very substantial alphas. Moreover, we have not at all tried to tune our strategies so as to reduce turnover, which could generate positive alpha for higher trading fees.

Figure 15: Robustness Check : Performance of ML Portfolios after Transaction Costs



We fully recognize that our method to account for transaction costs is quite crude, and at best only realistic for a small, marginal trader with essentially no price impact. If we wanted to study the costs faced by larger institutions, more complex methods would have to be devised, which could notably take into account bid-ask spreads, trading volumes, liquidity under stress etc. Arnott et al. (2019) presents interesting ideas to account for transaction costs when backtesting ML-Portfolios, but we choose to leave these promising explorations to future research.

In our view, these robustness checks have largely confirmed the initial findings: portfolios based on simple Machine-Learning predictions significantly outperform the market average across a wide array of strategies, time horizons and investment sets.

8. ML and Arbitrage Activity: Evidence from Short Interest

We set out to asses whether Machine Learning detects mispricings, or whether it learns fundamental asset pricing models, by analysing its relationship with arbitrage activity. This endeavour fundamentally rests on the hypothesis that, like good information, good algorithms are costly, and certain agents will therefore have an edge on the overall market. If Machine Learning was truly a source of arbitrage, we would expect some sophisticated agents to exploit this arbitrage, at least for as long as the market has not corrected the mispricing revealed by ML techniques.

As we have seen in the literature review, most existing research on arbitrage activity has focussed on short interest, notably the hallmark contribution by Hanson and Sunderam (2014). This follows the intuition that market prices are often determined by large, institutional investors (like asset managers, pension funds, insurance companies etc.), who face a regulatory prohibition to short-sell or avoid doing so because of the associated risks, but that smaller, sophisticated actors (typically hedge funds) can and generally do go short. We will thus try to detect the presence of arbitrageurs exploiting ML techniques through short-selling.

We first briefly present partial and general equilibrium models of short interest, which give some theoretical justification to our approach. We then study how much short interest is associated with the Machine-Learning Portfolios we have analysed, after which we turn to the direct relationship between Machine Learning predictions and short interest. Since these analyses generally yield no precise results, we then try to asses whether short interest can be detected among small-, mid- or large-cap companies, or whether using days-to-cover shows more of an effect. In almost all specifications, we fail to find a relationship between short interest and ML predictions that would be consistent with algorithmic market inefficiency.

8.1. A Short Model of Short Interest

Before moving to a general equilibrium approach, we first briefly present a partial equilibrium framework for short interest to motivate our identification approach. This first model crucially relies on the assumption that the class of informed agents is too small to have an effect on prices, which means that ML-related anomalies do not dissipate, but large enough to be detected in cross-section analyses of short interest. Readers are free to question the meaningfulness of such an approach...

Assume we have a very stylized financial market with two types of agents, uninformed and informed. The former make up all mass on the market and are price-makers, while the latter are an infinitesimally small class, which is thus price-taker. Uninformed agents can be seen as institutional investors, e.g. mutual or pension funds, while informed agents are arbitrageurs, e.g. hedge funds.

Beyond a risk-free asset delivering zero return $rf = 0$, there is a single risky asset with mean excess return $\mu = r_{risky} - rf$, whose return between t and $t+1$ is given by:

$$r_{t+1} = \mu + a(\mathbf{z}_t) + \varepsilon_{t+1} \tag{18}$$

Informed agents have access to the function $g \in \mathcal{A}_t^{informed}$, and thus observe $g(\mathbf{z}_t^A)$. Uninformed agents only have access to $\mathcal{A}_t^{uninformed} \subset \mathcal{A}_t^{informed}$ and markets are efficient with respect to $\mathcal{A}_t^{uninformed}$ (i.e. semi-strongly algorithmically efficient). This means that no algorithm $a' \in \mathcal{A}_t^{uninformed}$ provides additional information on r_{t+1}. Disturbances are centred and independent, $\mathbb{E}(a(\mathbf{z}_t)) = \mathbb{E}(\varepsilon_{t+1}) = \mathbb{E}(a(\mathbf{z}_t))\varepsilon_{t+1}) = 0$, while their variances are respectively σ_ε^2 and $\sigma_{a(\mathbf{z}_t)}^2$.

We further assume that both types of agents to have mean-variance preferences with risk aversion coefficients γ_u and γ_i. We can thus write their allocation of the risky asset as:

$$q_t^u = \frac{1}{\gamma_u} \times \frac{\mu}{\sigma_{a(\mathbf{z}_t)}^2 + \sigma_\varepsilon^2} \qquad\qquad q_t^i = \frac{1}{\gamma_u} \times \frac{\mu + a(\mathbf{z}_t)}{\sigma_\varepsilon^2} \tag{19}$$

Since $\mu > 0$ by design, the uninformed investor's allocation will be positive, and the standard Markowitz allocation respects his no-short-sale constraint. On the other hand, the sign of q_t^i will clearly depend on the realisation of the random variable $a(\mathbf{z}_t)$. However, if the private algorithm $a(\mathbf{z}_t)$ is what we decide to call *strongly informative*, i.e. if $\mathbb{P}(a(\mathbf{z}_t) < -\mu) > 0$, then the probability of the informed investor being short the risky asset is strictly positive as well: $\mathbb{P}(q_t^i < 0) = \mathbb{P}(a(\mathbf{z}_t) < -\mu) > 0$.

Additionally, we denote $S = -q_t^u \mathbb{1}_{\{q_t^u < 0\}} - q_t^i \mathbb{1}_{\{q_t^i < 0\}} = -q_t^i \mathbb{1}_{\{q_t^i < 0\}}$ the total amount of open short interest. We can see that:

$$\mathrm{Cov}\big(q_t^i, a(\mathbf{z}_t)\big) = \frac{1}{\gamma_u} \times \frac{1}{\sigma_\varepsilon^2} \times \mathrm{Cov}\big(a(\mathbf{z}_t), a(\mathbf{z}_t)\big) = \frac{\sigma_{a(\mathbf{z}_t)}^2}{\gamma_u \times \sigma_\varepsilon^2} > 0$$

$$\begin{aligned}
\mathrm{Cov}\big(\mathbb{1}_{\{q_t^i<0\}}, a(\mathbf{z}_t)\big) &= \mathrm{Cov}\big(\mathbb{1}_{\{a(\mathbf{z}_t)<-\mu\}}, a(\mathbf{z}_t)\big) = \mathbb{E}\big((\mathbb{1}_{\{a(\mathbf{z}_t)<-\mu\}} - P(a(\mathbf{z}_t) < -\mu))\, a(\mathbf{z}_t)\big) \\
&= P(1-P)\mathbb{E}(a(\mathbf{z}_t)|a(\mathbf{z}_t) < 0) + (1-P)(0-P)(\mathbb{E}(a(\mathbf{z}_t)|a(\mathbf{z}_t) > 0) \\
&= P(a(\mathbf{z}_t) < -\mu)(1 - P(a(\mathbf{z}_t) < -\mu))\Big[\mathbb{E}(a(\mathbf{z}_t)|a(\mathbf{z}_t) < -\mu) - \mathbb{E}(a(\mathbf{z}_t)|a(\mathbf{z}_t) > -\mu)\Big] < 0
\end{aligned}$$

where we have written P instead of $P(a(\mathbf{z}_t) < \mu)$ in the intermediary step for legibility. We can then turn to S:

$$\begin{aligned}\text{Cov}\big(S, a(\mathbf{z}_t)\big) &= \text{Cov}\big(-q_t^i \mathbb{1}_{\{q_t^i<0\}}, a(\mathbf{z}_t)\big) = \frac{-1}{\gamma_u \times \sigma_\varepsilon^2} \times \text{Cov}\big(a(\mathbf{z}_t)\mathbb{1}_{\{a(\mathbf{z}_t)<-\mu\}}, a(\mathbf{z}_t)\big) \\ &= \frac{-1}{\gamma_u \times \sigma_\varepsilon^2} \times \mathbb{E}\Big(\big(a(\mathbf{z}_t)\mathbb{1}_{\{a(\mathbf{z}_t)<-\mu\}} - \mathbb{E}(a(\mathbf{z}_t)\mathbb{1}_{\{a(\mathbf{z}_t)<-\mu\}})\big)\, a(\mathbf{z}_t)\Big) \\ &= \frac{-1}{\gamma_u \times \sigma_\varepsilon^2} \times P(a(\mathbf{z}_t) < -\mu) \times \mathbb{E}\big[a(\mathbf{z}_t)^2 \mid a(\mathbf{z}_t) < -\mu\big] < 0\end{aligned} \tag{20}$$

Since $-\mu < 0$, the *strong informativity* assumption implies that this covariance is non-zero, and an inspection of the factors' signs shows that it is in fact strictly negative. We have thus proven that the relationship between the predictive signal $a(\mathbf{z}_t)$ and the total short interest on the risky asset is negative. Moreover, *ceteris paribus*, it will be more negative when the risk-aversion of the informed type γ_u decreases, when the remaining variance of returns σ_ε^2 decreases and when the informativeness of the signal increases.

We should thus expect a cross-sectional regression of short interest S on the predictive signal $a(\mathbf{z}_t)$ to return a negative coefficient. Moreover, it is clear that this coefficient should be zero when all agents use $\mathcal{A}_t^{uninformed}$, i.e. when all agents are uninformed because markets are semi-strongly algorithmically efficient.

We now turn to a general equilibrium approach in which there are many stocks, and where informed agents make up a share $k \in [0,1]$ of the total population of investors. This model is largely based on the one developed in Hanson and Sunderam (2014). We adopt vector notations throughout the rest of the exposition, and write the vector of stock returns between t and $t+1$ as $\mathbf{r}_{t+1}$. As before, there are uninformed agents whose expectation of returns is $E^u(\mathbf{r}_{t+1})$, and informed agents who trade using the signal algorithmic signal $a(\mathbf{z}_t)$ and thus expect $E^i(\mathbf{r}_{t+1}) = E^u(\mathbf{r}_{t+1}) + a(\mathbf{z}_t)$.

We further assume that the uninformed agents are *radically uninformed*, i.e. they do not know that they are uninformed, and that the signal is not stochastic, which very conveniently gives both agents the same variance-covariance matrix: $\mathbb{V}_t^u(\mathbf{r}_{t+1}) = \mathbb{V}_t^i(\mathbf{r}_{t+1})$. Moreover, we assume that the signal induces cross-sectional mispricing but no aggregate mispricing, which writes $\mathbf{w}'a(\mathbf{z}_t) = 0$, where $\mathbf{w}$ is the vector of each stock's share of the overall market capitalisation.

Assuming that both types of investors are mean-variance investors with risk aversion coefficient γ, we get the following portfolio allocations, or demands:

$$\begin{aligned}\mathbf{q}_t^i &= \frac{1}{\gamma} \times \mathbb{V}^{-1}(\mathbf{r}_{t+1}) \times \mathbb{E}^i(\mathbf{r}_{t+1}) \\ \mathbf{q}_t^u &= \frac{1}{\gamma} \times \mathbb{V}^{-1}(\mathbf{r}_{t+1}) \times \mathbb{E}^u(\mathbf{r}_{t+1}) = \frac{1}{\gamma} \times \mathbb{V}^{-1}(\mathbf{r}_{t+1}) \times \big(\mathbb{E}^u(\mathbf{r}_{t+1}) - a(\mathbf{z}_t)\big)\end{aligned} \tag{21}$$

Since the total supply of stocks is given by the vector $\mathbf{w}$, market clearing writes:

$$\begin{aligned}\mathbf{w} &= (1-k) \times \frac{1}{\gamma} \times \mathbb{V}^{-1}(\mathbf{r}_{t+1}) \times \big(\mathbb{E}^u(\mathbf{r}_{t+1}) - a(\mathbf{z}_t)\big) + k \times \frac{1}{\gamma} \times \mathbb{V}^{-1}(\mathbf{r}_{t+1}) \times \mathbb{E}^i(\mathbf{r}_{t+1}) \\ \Rightarrow \quad \mathbf{w} &= \frac{1}{\gamma} \times \mathbb{V}^{-1}(\mathbf{r}_{t+1}) \times \mathbb{E}^i(\mathbf{r}_{t+1}) - (1-k) \times \frac{1}{\gamma} \times \mathbb{V}^{-1}(\mathbf{r}_{t+1}) \times a(\mathbf{z}_t) \\ \Rightarrow \quad \mathbb{E}^i(\mathbf{r}_{t+1}) &= (1-k) \times a(\mathbf{z}_t) + \gamma \times \mathbb{V}(\mathbf{r}_{t+1}) \times \mathbf{w} \\ \Rightarrow \quad \mathbb{E}^i(\mathbf{r}_{t+1}) &= (1-k) \times a(\mathbf{z}_t) + \frac{\text{Cov}(\mathbf{r}_{t+1}, r_m)}{\mathbb{V}(r_m)} \times \mathbb{E}(r_m)\end{aligned} \tag{22}$$

Where we have written the last line by denoting r_m the return on the market. The second part of the equation corresponds to CAPM-beta and, correspondingly, the first part can be identified as alpha: the alpha term is increasing with the signal $a(\mathbf{z}_t)$, and its overall magnitude decreases with the share of informed agents k, converging to zero when the latter approaches 1.

Assuming they initially have an equal allocation $k \times \mathbf{w}$, the total position of informed agents is:

$$k \times q^i = k \times \mathbf{w} + k(1-k) \times \frac{1}{\gamma} \times \mathbb{V}^{-1} \times a(\mathbf{z}_t) \tag{23}$$

We compute the Short Ratio of the informed agents by dividing by the market share of stocks:

$$SR^i = -k \times q^i = -k \times \mathbf{w}^{-1} - k(1-k) \times \frac{1}{\gamma} \times \mathbb{V}(r_{t+1})^{-1} \times a(\mathbf{z}_t) \times \mathbf{w}^{-1} \tag{24}$$

Since in the general equilibrium framework we have not assumed that uninformed investors are barred from going short, the overall short ratios on stocks will be given by $SR = SR^i + SR^u$. However, because uninformed agents do not observe the signal $a(\mathbf{z}_t)$, it will be independent of their Short Ratio SR^u. This implies that the dependence of SR and $a(\mathbf{z}_t)$ can be deduced from SR^i:

$$\text{Cov}(SR, a(\mathbf{z}_t)) = -k(1-k) \times \frac{1}{\gamma} \times \mathbb{V}(r_{t+1})^{-1} \times \mathbb{V}(a(\mathbf{z}_t)) \times \mathbf{w}^{-1} < 0 \tag{25}$$

Since all elements in the last term are positive except for the minus sign, each term of this (somewhat ill-defined) covariance vector will be negative. In plain English, a cross-sectional regression of short ratios on the signal should return a negative coefficient : the general equilibrium analysis induces the same main result as the partial equilibrium approach.

Three caveats should be stressed about our general equilibrium approach: (i) it crucially assumes that uninformed investors do not know that they are uninformed, and that informed investors benefit from a deterministic and not stochastic signal: this means they face the same variance-covariance matrix of returns, without which all the analysis breaks down and the model becomes essentially intractable; (ii) similarly, to derive its solution, the model supposes that informed and uninformed agents have the same risk aversion coefficient γ, which seems very unlikely to be true in reality given their fundamentally different nature; (iii) in this part, the term Short ratio constitutes an abuse of language, because it does not exclusively focus on stocks that are held short, and its definition does not guarantee that it is positive; however, adding an indicator to make this precise is tedious, does not add any intuition and is thus left as an exercise to the overly curious reader.

Albeit to a lesser extent, this third issue plagues our partial equilibrium approach as well, and most models of short interest more broadly: the probability of non-zero short interest is quite low in both models, but short interest data shows that virtually all stocks have at least some open short interest. This shows that models with much more pronounced disagreement among agents are needed to accurately describe arbitrage activity, but that quandary is far beyond the reach of our modest inquiry.

In short, both the partial and general equilibrium approaches show that, in a world with informed agents that trade on an algorithmic signal before the general public, we should expect a negative cross-sectional relationship between short ratios and Machine Learning signals. This is the basic insight on which we construct our test of algorithmic market efficiency.

8.2. Data on Short Interest

The most comprehensive available data on short interest comes Compustat's Supplemental Short Interest File. This dataset contains short interest levels for stocks traded on the NYSE and AMEX exchanges as far back as 1973, but only from July 2003 for stocks traded on NASDAQ. It is notably used in the hallmark study on arbitrage activity by Hanson and Sunderam (2014) : the authors augment it with short interest data from NASDAQ covering 1988-2003, but this data has not been made public since.

The Supplemental Short Interest File contains 2,221,113 observations for open short interest on the 15th of each month and at month-end. Rows are identified by *gvkey* for the relevant and by *iid* for the relevant stock issue. To cross this data with the main factor and return dataset, we first had to merge it onto the CRSP / Compustat Merged Security Monthly dataset. This dataset, which has 6,130,813 rows, additionally contains information on monthly trading volume per stock issue (*cshtrm*), which we also use in our analysis.

We then collapse the new merged dataset by *permno* (the new, CRSP-compatible firm identifier) and date, while summing all observations for short interest and traded volume across different stock issues (*iid*'s). This matched dataset contains 2,353,103 observations for monthly traded volume (with 1,375,708 missing) and 1,157,513 observations for monthly open short interest (with 2,571,298 missing).

We then collapse the new merged dataset by *permno* (the new, CRSP-compatible firm identifier) and date, while summing all observations for short interest and traded volume across different stock issues (*iid*'s). This matched dataset contains 2,353,103 observations for monthly traded volume (with 1,375,708 missing) and 1,157,513 observations for monthly open short interest (with 2,571,298 missing).

For reasons that are not entirely obvious, perhaps linked to regulatory filings or selective disclosure by exchanges, this dataset exhibits sharp monthly oscillations in the number of observations. For a given firm, some months are missing even when there are many available data points both before and after. Moreover, these missing observations usually only concern an isolated month.

To overcome this (admittedly somewhat intriguing) limitation, we apply a simple linear interpolation to both monthly trading volume and month-end short interest, while fixing the maximum length of allowed interpolation to a quarter, i.e. three months. This simple interpolation, an example of which on Ford is shown in Figure A2 to demonstrate its reasonableness, brings the number of total observations up to 3,357,416 for monthly trading volume, and to 1,705,369 for monthly short interest. In the annex, we present most main analyses on the uninterpolated data, which show the large swings in observations, but also that the interpolation has little effect on the main results.

To obtain proper measures of trading and arbitrage activity, we normalize these quantities by the number of total shares outstanding, *shrout*, from CRSP. This value is available for all relevant observations, and thus produces no further missing values. Finally, to correct for broken extreme values which probably arise from mismatches with *shrout* (the highest short ratio is 598...), we winsorize both datasets at the 1st and 99th percentile of each year.

To check the robustness of our results, we also use the days to cover measure of short interest: days to cover are equal to the total amount of short interest (*shortint*) divided by monthly trading volume (*cshtrm*) (and again divided by $253/12 \approx 21$ to obtain a measure in days). It measures the number of days a short-seller would need to cover his short position in the market and is an alternative, probably better micro-founded normalization for short interest. Most of our results will be shown to be robust to this change of specification, but some are not (generally, the pattern of days to cover is more erratic).

This is perhaps due to the different economic nature of these two measures. Since this is a potentially infinite discussion, we leave our readers with the wise words of Asquith, Pathak, and Ritter (2005) : "If one views short interest as indicative of future buying pressure as short sellers cover

their positions, the days to cover ratio is arguably the best measure. But if one views short interest as reflecting the information of informed investors, then the short interest to shares outstanding ratio is arguably the best measure. In any event, these two measures are positively correlated."

8.3. Short Interest around ML-Portfolios

We first try to assess whether there is excess short interest along the Machine Learning Portfolios we have analysed earlier. Obviously, this can only be applied to the long-short portfolios, i.e. to the top/bottom decile and to the rank-weighted strategy. We compute the short interest associated with a strategy as the weighted sum of individual stock ratios. For strategy $\mathcal{S}$ giving weight $w_{i,t}(\mathcal{S})$ to stock i at time t (such that all stock weights at time t sum to 1), and denoting $SR_{i,t}$ the short ratio of stock i at time t, the strategy's weighted short ratio at time t writes:

$$SR_t(\mathcal{S}) = \sum_{i=0}^{N_t} w_{i,t}(\mathcal{S}) \times SR_{i,t} \tag{26}$$

Note that since all strategies are value-weighted, this also means that the short-ratio will depend much more on the short ratios of companies with large market capitalizations. The models above have shown that, if there are arbitrageurs going short on the short side of the portfolio, we expect this quantity to be negative: $w_{i,t}(\mathcal{S})$ is positively correlated with the ML signal and $\times SR_{i,t}$ is negatively correlated with the ML signal, so the correlation between the two will be negative. In simpler terms, we expect arbitrageurs to go short the short side, where SR will be positive, and to go long the long side, where SR is null (or at least small), so overall the strategy should have a negative short ratio. On the other hand, if there is no correlation between stock's Short Ratio and the ML signal, the strategy's Short Ratio should be null.

When computing strategy-level short ratios, we treat missing short ratio observations as zero. This could introduce biases, but we have not found a more robust method do deal with this issue. As the plots in Figure A2 show, there are virtually no observations before the 1970s, and data only becomes truly reliable after 2000. Yet, since we compute short ratios for value-weighted strategies in which large companies are dominant and since observations are likely to be available for these companies fairly early, the data for the 1980s and especially 1990s seems reasonably reliable to us.

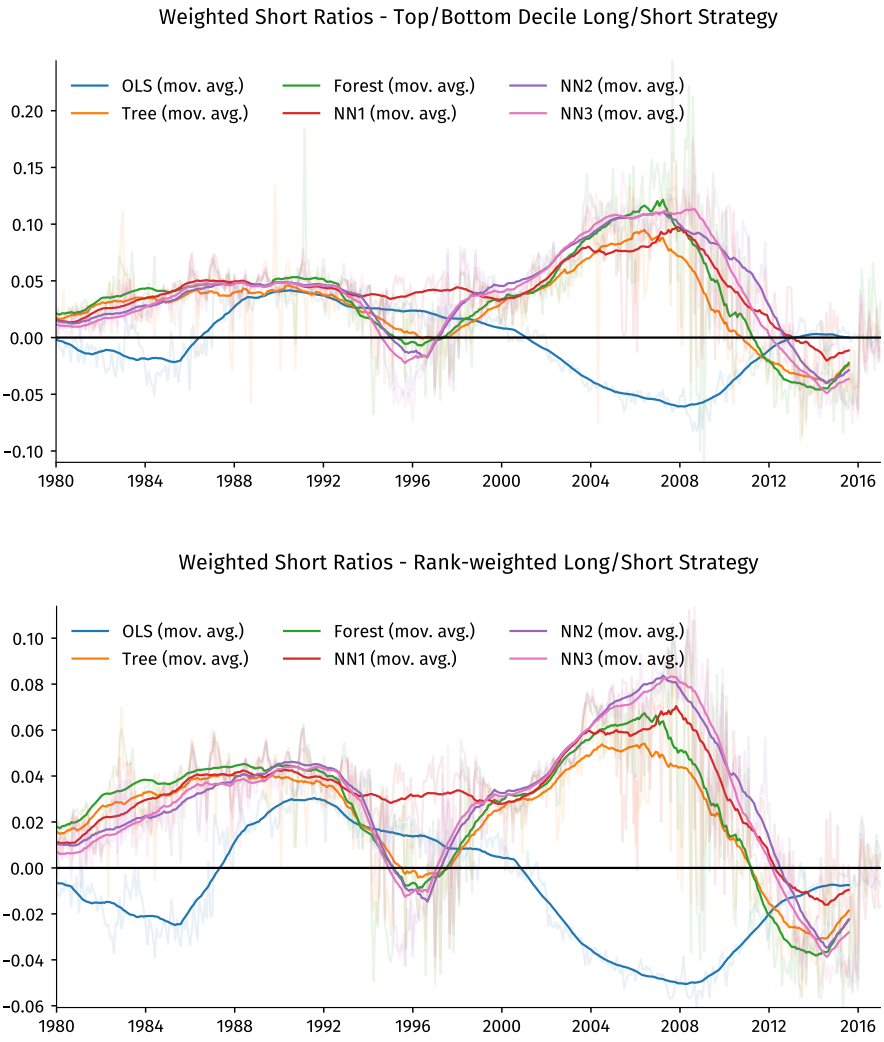
Figure 16 shows the weighted short ratios for selected Machine Learning strategies, i.e. OLS and the non-linear methods that performed best in the previous sections. Actual short ratios are shown with transparency, while the solid line show centred 3-year moving averages. Non-linear methods have surprisingly similar evolutions for decile spreads, since their Short Ratios are mostly positive in the 1980s, turn slightly negative around 1995, and then again become very positive with a peak around 2005 before declining again and reaching negative territory in the mid-2010s. For OLS, the pattern is markedly different, since it is mostly positive until the 2000s and then moves on to spend a decade in negative territory, before converging to zero by 2010.

This effect is remarkably similar for the rank-weighted portfolios, albeit the magnitude of the swings is lower, probably because of the lower weights assigned to the extremes of predicted returns. While strategy short ratios are far from being zero, which supports the first hypothesis of our approach (that Short Ratios are not uncorrelated with Machine Learning signals), this dependence is almost always positive: although the degree varies, non-linear methods have generally trodden firmly within positive territory. This suggests a *positive* relationship between Machine Learning forecasts and shorting activity, which runs counter to the intuition built in our modelling exercise.

In the Annex, we present the same figures using days to cover as a measure of short interest activity. We use two measures of days to cover for a portfolio. The first takes all the days to cover from the short side of the portfolio and multiplies them by their weight. This quantity has no economical

interpretation, but can be viewed as an importance-weighted average of the days to cover present in the portfolio: although some positions will take longer than this to be covered, they will impact the measure less if their share of the portfolio is very mall. The second measure is simply the highest day to cover of any stock in the short side of the portfolio: this is the actual days to cover measure of the portfolio, but will obviously give disproportionate importance to stocks that have a small weight in the portfolio.

Figure 16: Weighted Short Ratio in Selected Machine Learning Portfolios



Results for the first measure are shown in Figure A11, while results for the second measure are shown in Figure A12. The weighted days to cover follow trends that correspond to the Short Ratios, for both decile spread and rank-weighted strategies: for non-linear methods, it is very small when the Short Ratios are positive, indicating little-to-no arbitrage activity, and becomes sizeable when the Short Ratios become negative, which indicates increased arbitrage activity. Interestingly, the 1-Layer Neural Network does not follow the increase of the other non-linear methods around 1995. OLS exhibits an inverse pattern, again matching its inverse behaviour in the Short Ratio analysis. On the other hand, the maximum-based measure of short interest is volatile and hard to interpret for all strategies and methods, undoubtedly because it is exclusively driven by outliers. On the whole, we thus view this as corroborating evidence for our previous findings.

We also repeat the analysis for Trading Ratios, which presents a similar pattern to Short Ratios

(see Figure A13): this suggests that the evolution we observe is less due to arbitrage activity along ML Portfolios, but rather to the nature of stocks the ML strategies tend to select (and to short) over time. Finally, we present the same analysis on Short Interest using the uninterpolated data in the Annex (Figure A14). Unsurprisingly, this exhibits wild and frequent swings because missing observations are treated as zero, but the centred 3-year average gives essentially the same result: the pattern of Short Ratio evolution for the uninterpolated data is the same as for the interpolated data, which shows that this is not manifestly due to our data preparation process.

To further analyse this effect, and to verify that it is not simply a figment of our portfolio construction approach, we now turn to an analysis of the direct relationship between Machine Learning forecasts and short interest.

8.4. Short Interest around Machine Learning Predictions

Indeed, we can directly study the relationship between stock-level performance forecasts and open short interest. The fairly intuitive results from our brief modelization concern the cross-section of returns and predicted returns directly, rather than portfolio-level short interests as we have analysed them before. Moreover, since the portfolio-level analysis has yielded some seemingly counter-intuitive results, we will try to corroborate them by looking directly at the underlying data.

To further clarify the testable predictions from Section 8.1, we first plot both variables against each other for an example year that works "as it should", 2015, and a Machine Learning method, 3-Layer Neural Networks. In Figure 17, the subfigure to the left plots monthly short ratio against monthly returns as predicted by the 3-Layer Neural Network for all observations in the year 2015. Short ratio is presented in log scale, and all 68,919 points are shown with a low transparency, so darkness serves as a (primitive) measure of point density. However, there is still considerable bunching around 0%-0.1% predicted return and 10^{-4} - 10^{-1} short ratio.

To better showcase this relationship, the right subfigure plots monthly short ratio percentile against monthly predicted return percentile. Essentially, this is a projection of the previous subfigure onto a unit square with homogeneous density along each axis. This latter graph shows quite clearly the expected inverse relationship: the are more points (the shade is darker) in the top left, where stocks forecasted to do poorly are shorted more than average, and bottom right, where stocks forecasted to do well are shorted less than average. Consequentially, there are fewer points on the bottom right and on the top left.

Figure 17: Intuitive Short Interest Example : 3-Layer Neural Network in 2015

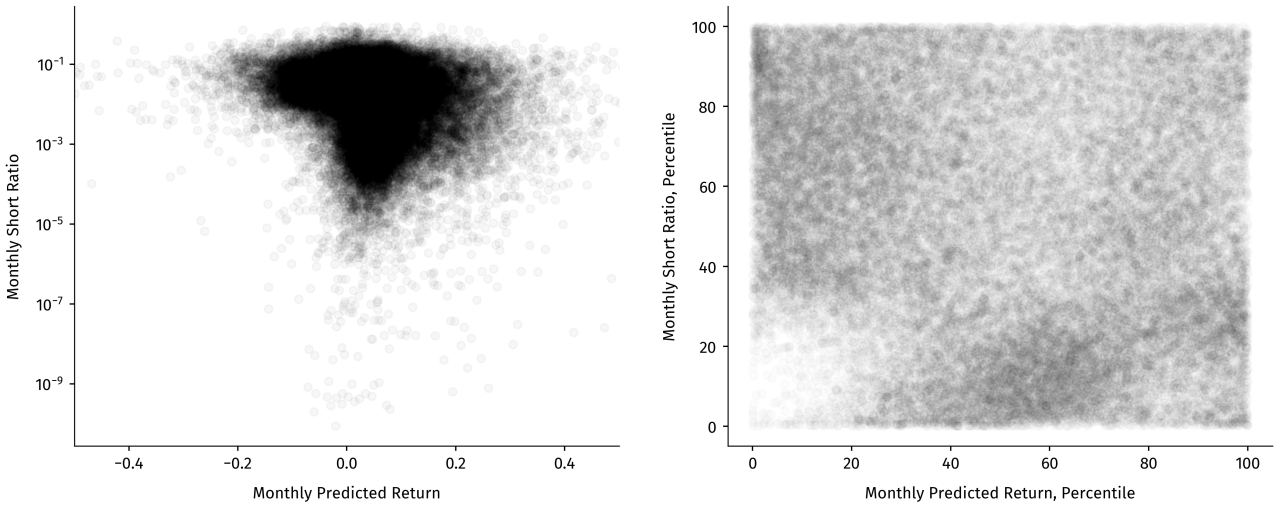
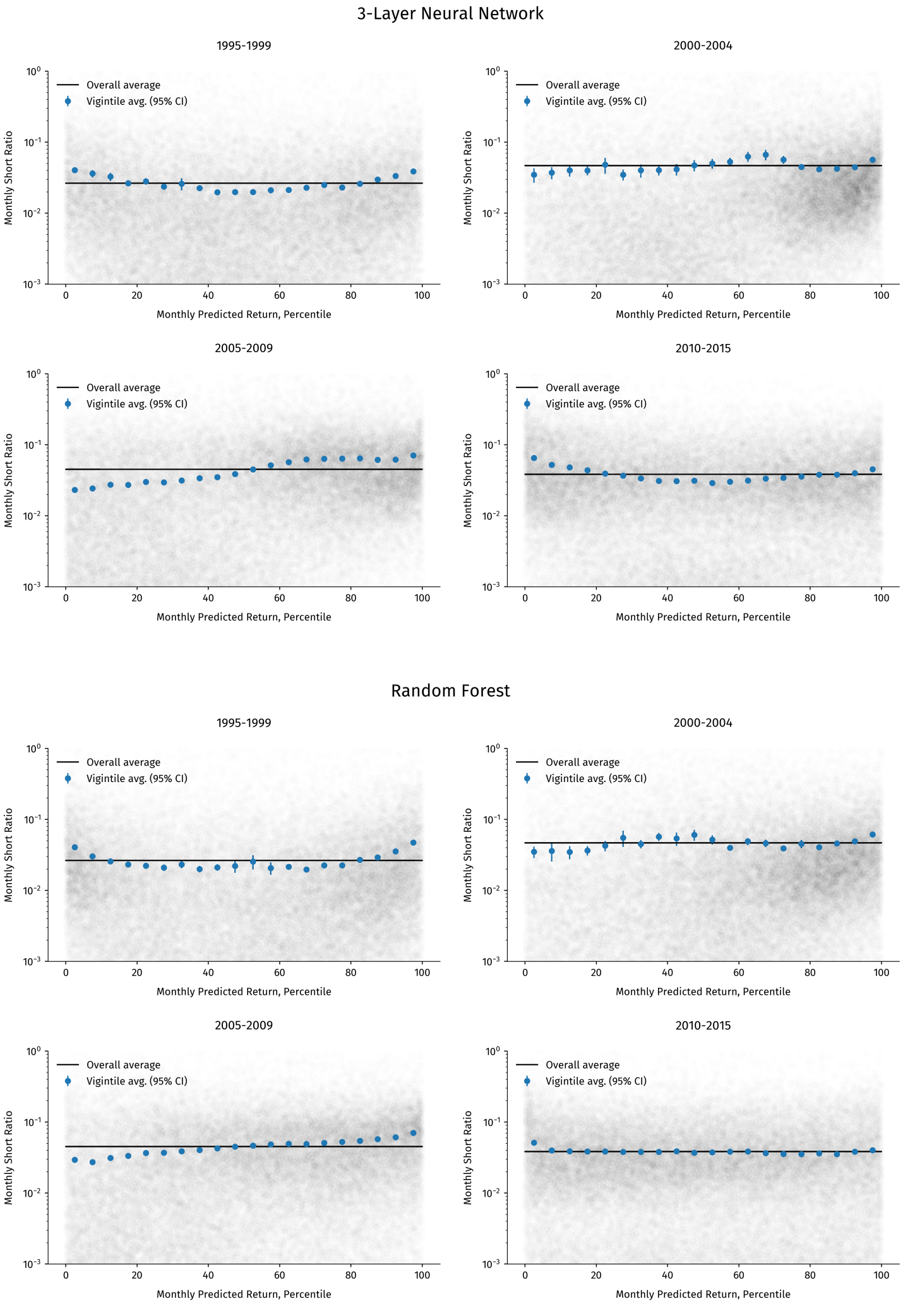


Figure 18: Cross-Section Analysis of Short Ratios along Machine Learning Predictions



The main issue with this intuitive finding is that it does not hold in general. In fact, we mostly observe either no or a positive relationship between forecasted returns and shorting activity. Figure 18 repeats this cross-sectional analysis for the four most recent quinquennia available in our dataset, 1995-1999, 2000-2004, 2005-2009 and 2010-2015. For each period, we plot the monthly short ratio against the monthly predicted return percentile. The light black points are individual observations (we have randomly sampled 50,000 in each graph), so darkness again serves as a proxy for density. Additionally, the black line plots the overall average short ratio. Finally, the blue points represented 20 local averages by bins of monthly predicted returns, while the lines around them represent 95% confidence intervals. Because there are so many observations, the latter are usually quite small, if not invisible. Missing observations are not plotted, and not taken into account in means and binned means. A these analyses rely on the uninterpolated data for Short Interest.

The first graph shows this analysis for the 3-Layer Neural Network, and the second for the Random Forest. Both show fairly similar trends, wherein the relationship is very weak between 1995 and 1990, slightly positive between 2000 and 2009, and then weak to slightly negative over 2010-2015. However, the starkest result seems to be the lack of results: there is no clear monotonic, or even non-monotonic, pattern emerging. Moreover, the minute patterns that do emerge do not hold constant over time.

In the Figures A15 and A16, we show similar graphs for OLS, Regression Trees as well as 1- and 2-Layer Neural Networks. The three non-linear methods display similarly muddled behaviour: the relationship is erratic in the late 1990s, generally positive over 2000-2005 and 2005-2009 and usually negative in the early 2010s. OLS shows a very different, but all the more interesting, pattern: in all four quinquennia, the amount of open short interest on the bottom percentiles is significantly higher than the market average. Moreover, confirming a tendency already looming in 2000-2004, there is significantly less open short interest in the top predicted return percentiles than in the overall market over 2005-2009 and 2010-2015. Although these results should be taken with caution, this pattern seems highly indicative of the presence of arbitrageurs exploiting fundamental information and Ordinary Least Squares Regression to go long and short on the stock market.

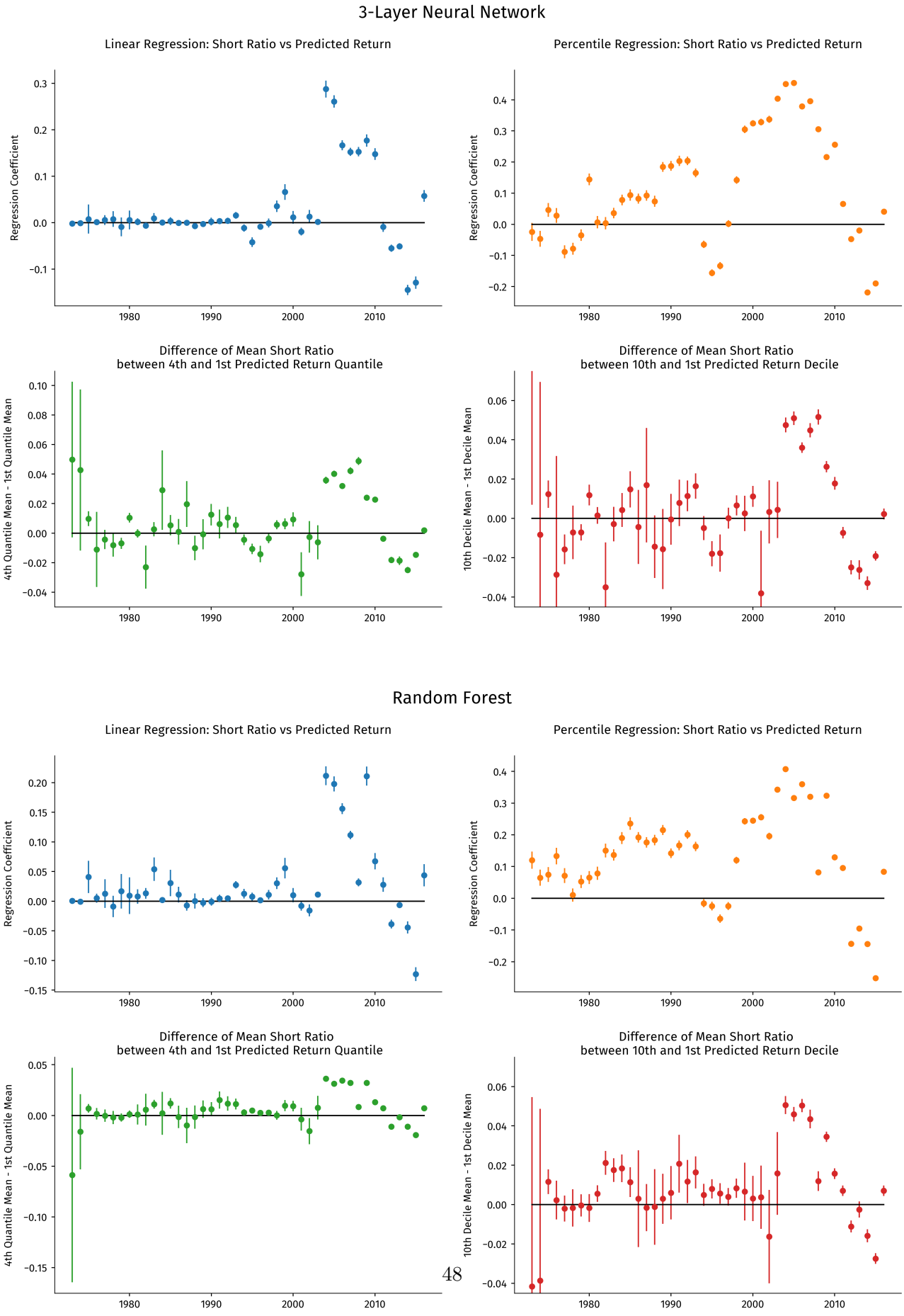
We can then extend these cross-sectional analyses to more precisely study time-series variation. In Figure 19, we represent four measures of the relationship between predicted monthly return and short interest ratio: (i) the parameter for the short ratio in a simple linear regression on the monthly predicted return; (ii) the parameter for the short ratio in a percentile regression on the monthly predicted return; (iii) the difference between the average short ratio within the 4th and 1st quartiles of predicted returns; (iv) the difference between the average short ratio within the 10th and 1st deciles of predicted returns. Again, we do so for Random Forests and 3-Layer Neural Network.

All these quantities, albeit always somewhat differently, measure the sign of the relationship between open short interest and ML-predicted returns: in the expected setting in which there is excess short interest around stocks which are predicted to do poorly, we would be expecting an inverse relationship between both variables and therefore a negative value for each measure. Conversely, a positive measure would go against the expected result, because it is indicative of exacerbated short interest around the stocks that Machine Learning techniques predict should do well.

We compute 95% confidence intervals for all point estimates. They are based on standard t-tests for the first two regression parameters, and on a Welch two-sample t-test (a variation of the standard two-sample t-test that allows for unequal variances) for the two latter measures. In the early periods (where many observations are missing and therefore dropped) estimates are often insignificant, but in the later periods they are usually very precisely estimated.

We can observe that, as we had seen before in the ML-portfolio level analysis and then in the various cross-sectional analyses, there seems to be a positive relationship between short interest and returns predicted by Random Forests and 3-Layer Neural Networks. All four measures find little significant evidence of a strong relationship between both variables before $\sim$1995, when there is a first

Figure 19: Time Series Analysis of Short Ratios around Machine Learning Predictions



increase into positive territory that is shortly followed by a sharp drop. Afterwards, we see a long period of positive dependence in the 2000s, which erodes after 2008-2009 to reach negative territory in the 2010s. Remarkably, the pattern is very similar for the 3-Layer Neural Network and for the Random Forest.

In the Annex, we again show the same analysis for OLS and Regression Trees (Figure A15) as well as 1- and 2-Layer Neural Networks (Figure A16). As before, the three non-linear methods show a roughly similar pattern to the two methods studied above, which usually treads in the null-or-positive territory. However, here again, OLS exhibits a markedly negative relationship between short interest and predicted return. This relationship is negative over most of the time period since the 1990s, although it has reached a particularly high levels in the mid-2000s and receded since then. This pattern is consistent with a story in which OLS captures simple, linear asset-pricing models that were mostly discovered in the 1990s and subsequently "exploited" by specialized Funds, but that have yielded relatively disappointing results and thus received decreasing enthusiasm in the 2010s. However, here again, these results should be taken with caution.

8.5. Robustness tests : Firm Size and Days to Cover

To verify the robustness of these non-results, we try to explore alternatives measures of short interest, i.e. days to cover, and we try to assess whether there is a detectable effect among a subset of variables by distinguishing between small, mid-size and large firms.

To check for robustness, we repeat the cross-section and time-series analysis for both non-linear methods, Random Forests and 3-Layer Neural Networks, using the days to cover measure of short interest. The cross-section analysis, represented in Figure A19, shows that while seemingly more erratic than Short Ratios, there still seems to be a roughly negative relationship in the late 1990s, replaced by a roughly positive one in 2000-2004 and 2005-2009 and then a roughly null dependence over the early 2010s. This applies for both methods, and is confirmed by the time series analysis, presented in Figure A20.

For good measure, we also showcase the previous two analyses for OLS in Figure A21. Since this very basic method has shown a consistent and intuition-conform pattern for Short Ratios, we wanted to verify whether this holds for days to cover as well, but this time the robustness verification is less clear-cut. Indeed, the cross-section graph shows some excess short interest for bottom predictions in the late 1990s, and scant short interest for top predictions over 2005-2015. However, it is difficult to make out a clear pattern overall. Similarly, the time series analysis presents a picture that varies from measure-to-measure. The simple regression and the decile-binned tail spread are often insignificant, but sometimes very positive and more rarely very negative. On the other hand, the percentile regression and the quartile-binned tail spread are significantly negative after the mid-1990s, with a return towards zero in the early 2010s. The two latter results coincide quite neatly with the evolution observed for Short Ratios around OLS predictions. On the other hand, the mercurial behaviour of the other two measures, and the general lack of a clear pattern in the cross-section, seems harder to explain and should not be hand-waved.

Indeed, the two last measures, which are based on differences in quantile means and thus fairly local measures, show very spasmodic evolution, that probably reflect the more erratic behaviour observed in the cross-section. However, the two first measures, which are based on regressions and thus take the full sample into account, show a more stable pattern that is generally highly significant and that mimics the evolution already observed for Short Ratios. Since this was also the evolution suggested by the cross-section analysis, we are led to conclude that our results are robust to a change in the exact specification of short interest.

We then try to assess whether these conclusions hold for all subsets of stocks, or if is not simply driven by a very large number of small-cap stocks that are not reflective of the overall stock market. To do so, we partition the stock universe into three (entirely arbitrary) categories: *big cap* companies, which have market capitalizations above the 95th percentile in a given year, *mid cap* companies, which lie between the 75th and 50th percentile, and *small cap* companies that are below the 50th percentile. See Figure 20. In 2016, this method yields cutoffs at 20,4B$ and 2,4B$ respectively, which seem quite reasonable.

Figure 20 repeats the cross-sectional analysis for 3-Layer Neural Networks and Random Forests, but draws the binned means in different colours for different groups: red for small caps, orange for mid caps and green for large caps. Again, bars represent 95% confidence intervals for the point estimates. To ensure comparability, predicted return bin edges are computed first on all observations and then separated by size, which means that the number of stocks described by one point varies. On the whole, the pattern (or rather, the lack of pattern) from the undifferentiated analysis holds: while the average level of short interest varies between firm sizes, the relationship between predicted returns and short interest, i.e. the slope of points, is generally identical and close to null.

Especially for Neural Networks, the same pattern seems to hold true: the slope of the points is slightly negative in the later 1990s, slightly positive in the late 2000s and again slightly negative in the 2010s across all three sizes. A notable exception is 2000-2004, where both large and small caps show wild variation: while is hard to explain, the second is probably due to the small sample size (by construction, 5% of the overall sample) for small cap companies. For Random Forests, the effect is qualitatively the same, although again the figure is quite erratic for 2000-2004.

Figure 21 repeats the time series analysis from the previous section, but this time focusses on only one measure: the percentile regression (previously in the top right corner). Indeed, it allows us to take the full sample into account, which is important in our reduced-sample setting, but the percentile normalization also ensures that coefficients are comparable across methods (because predicted returns have very heterogeneous variances across methods) and across market capitalizations (because the distribution of short ratios is far from uniform, so market cap groups that will be shorted more or less will also have over-proportionally large or small short ratios). Since all four previous measures were quite strongly correlated, this choice only has a limited effect on the qualitative results anyway.

For 3-Layer Neural Networks, the pattern that emerges looks remarkably similar across market caps, and closely resembles the initial result: the association between short ratios and predicted returns is slightly positive in the late 1990s, highly positive in the 2000s and then converges back to zero in the 2010s. This is very clear for large caps and mid caps (the latter even turning negative around 1995 like the overall measure), but the pattern is less obvious for small caps, for which coefficients are significant but vary from year to year, which suggests that no systematic relationship exists.

For Random Forest, the qualitative result is similar although it appears more blurred, like in the undifferentiated analysis. Big caps in particular often have insignificant coefficients in a given year, but the multi-year alignment of coefficients suggests that a pattern similar to mid-caps, which themselves are very close to the overall trend, is fairly likely. Again however, the relationship is very mercurial for small caps. This result is perhaps explained by the fact that small caps have much higher idiosyncratic risks, which makes short-selling more expensive and thus discourages systematic traders, and even potentially informed arbitrageurs, to short-sell large amounts in that market.

Although they are not shown, both cross-sectional and time-variation results are broadly the same for Regression Trees as well as 1- and 2-Layer Neural Networks.

Figure 20: Cross-Section of Short Ratios along ML Predictions by Market Capitalization

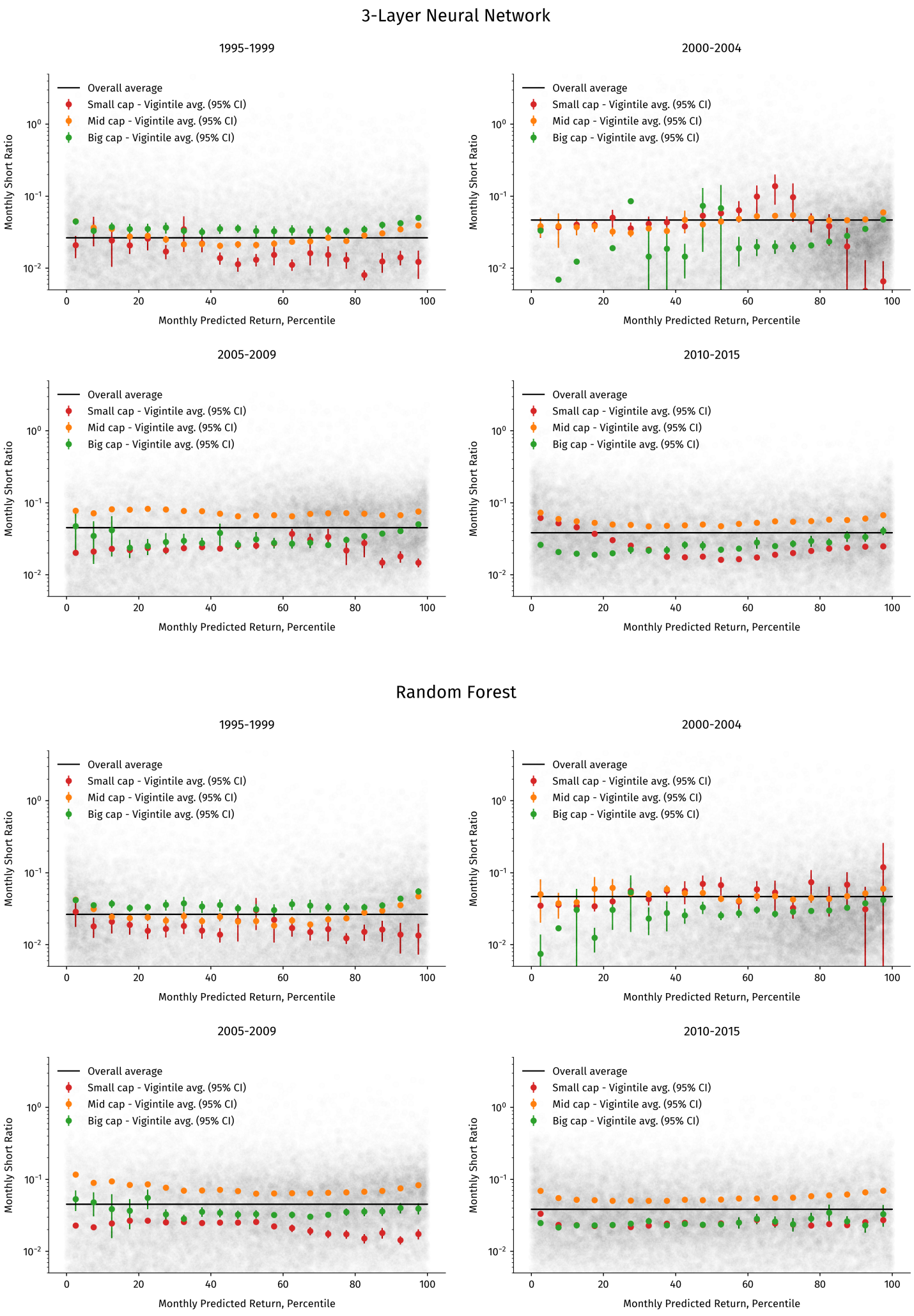


Figure 21: Time Series of Short Ratios around ML Predictions by Market Capitalization

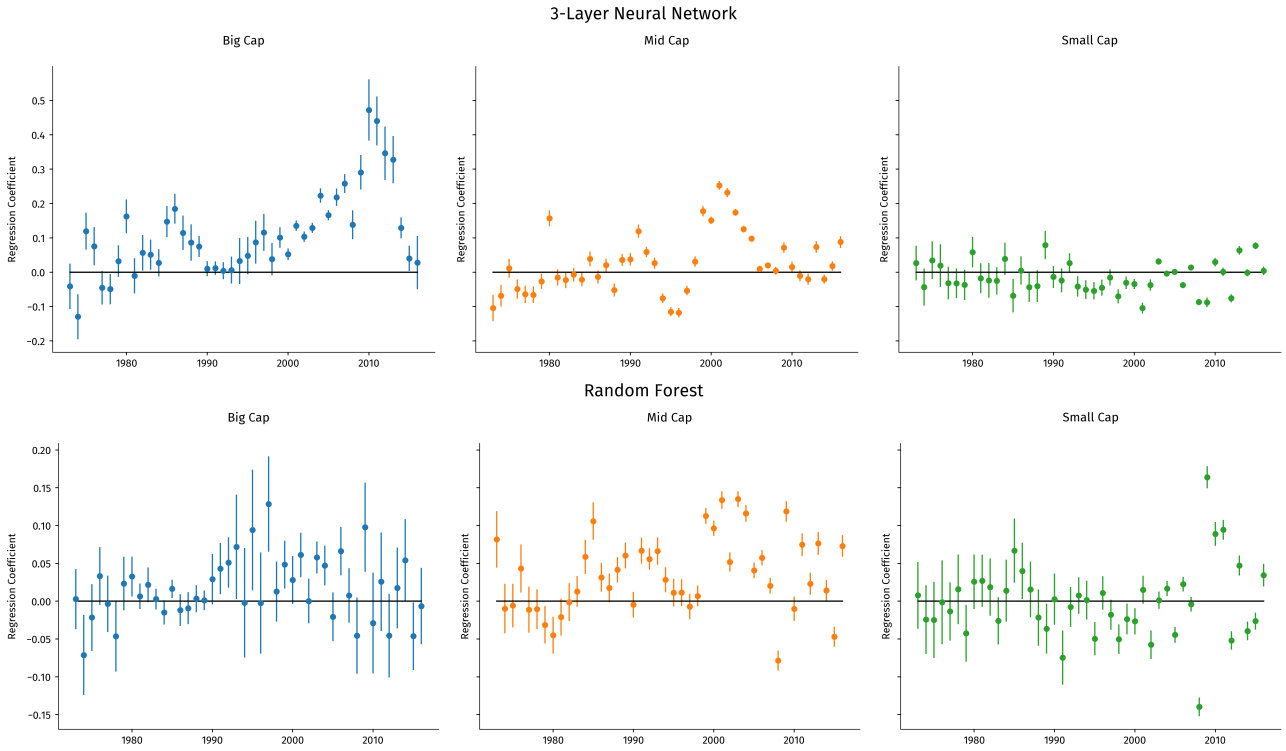


Figure A22 repeats both the cross-sectional and time series analysis by market capitalization for OLS. The marked positive relationship which we had identified earlier holds remarkably well : although the level of short interest varies by market cap groups, the slope of the points is always very clearly negative. Moreover, this is true in all periods, although it seems more pronounced over 2000-2009. Unsurprisingly, the associated time series analysis yields very consistently negative slopes: the coefficients are very highly negative in all years after 1992 for big caps, but they are significantly negative in almost all years for mid- and small-caps as well. This shows that the pattern, whereby OLS works *as expected* for a Machine Learning strategy exploited by better-informed arbitrageurs, is remarkably robust to restrictions in market capitalization.

These two refined analyses, focussing on Days to cover and market capitalizations, have thus broadly confirmed our initial result: the relationship between returns predicted by Machine Learning techniques and open short interest varies over time but is generally negative. A significant exception to this pattern is Ordinary Least Squares, for which the expected inverse relationship holds very consistently over time.

This suggests three very different interpretations. The first is that, assuming they exist, informed arbitrageurs using Machine Learning techniques are particularly inept at implementing arbitrage strategies and continually get the directionality of their trades wrong.

The second is that the relationship between predicted returns and short interest rather reflects the *nature* of stocks predicted to do well or to do poorly by different ML methods, and that the similar pattern observed between methods shows that they have learned similar forecasting functions; in turn, this implies that arbitrage trading exploiting ML methods is non-existent or at least very small.

The third interpretation simply says that absence of evidence should not be seen as absence of evidence, and that better data, more refined specifications or better identification strategies could show the existence of arbitrage capital allocated to Machine Learning strategies.

In our modest opinion, the truth probably lies somewhere between two and three.

9. Is there Post-publication Decline in Machine Learning?

Another piece of evidence we can bring to the great puzzle of Machine Learning in asset pricing is to study whether there is post-publication dissipation of returns, i.e. whether the discovery and wide dissemination of a ML technique leads to a decline in the (potentially arbitrage) profits it generates. This closely follows the approach originated by McLean and Pontiff (2016), although the authors use it both to study market efficiency implications of cross-sectional anomalies and to assess the data-mining concerns inherent to the literature on return-predicting factors.

In the case of Machine Learning, evidence of such a decline would violate super-strong algorithmic market efficiency, because it would show that there were (and thus probably also still are) some undiscovered algorithms whose analytical contribution was not already reflected in market prices. On the other hand, if this decline is very gradual, it would be evidence for strong-form AEMH, because some arbitrageurs probably trade the strategy before others which introduces a drift. Put briefly however, we find absolutely no evidence of a reliable decline of returns after the publication of a Machine Learning method : Machine Learning research does not destroy stock return predictability.

Table 6: Publication Dates of Machine Learning Methods

Method	First Publication	Notes
OLS	Legendre (1805)	
Lasso	Santosa and Symes (1986), Tibshirani (1996)	Developed independently
Ridge	Tikhonov (1943), Hoerl (1962)	Invention, dissemination into statistics
Enet	Zou and Hastie (2005)	
PCR	Pearson (1901)	First publication for PCA
PLS	Wold (1966)	
Tree	Belson (1959)	See Loh (2014) for a detailed history
Forest	Ho (1995)	
GBRT	Friedman (2002)	
NNs	McCulloch and Pitts (1943)	Initial idea for ANNs
	Werbos (1974)	First suggested to use backpropagation
	McClelland et al. (1986)	Popularization of backpropagation for NNs
	LeCun et al. (1989)	Popularization of backpropagation for CNNs

Table 6 shows the publication dates of all Machine Learning methods we use in our analysis, which starts with the discovery of linear regression in 1805. Some methods have a fraught history, for example both Lasso and Ridge have been discovered independently in two different settings : for these, we take the earliest discovery as a reference. For Principal Components Regression, which is simply a combination of Principal Components Analysis and Linear Regression, we take the invention of PCA in 1901 as reference, and not directly the first use of PCR.

Neural Networks are particularly troublesome, because their history is very complex. Strictly speaking, their "invention" dates back to 1943, when McCulloch and Pitts (1943) first described ANNs combining linear regressors in a multi-layered architecture under the poetic name of a *"logical calculus of the ideas immanent in nervous activity"*. However, the estimation of such networks remained an essentially untractable problem until Werbos (1974) suggested, in his PhD disseration, to use the backpropagation technique in 1974 (interestingly enough, his work was meant to be directly applied to Social Sciences). McClelland et al. (1986) then contributed to the widespread popularization of back-propagation for Neural Networks. We thus take 1986 as a reference for Neural Networks, although we also mark 1974 to show that this choice does not change the qualitative conclusion.

All in all, this means there are only eight methods that have been discovered during our sample, i.e. between 1958 and 2016. Moreover, since predictions not based on the 10 full years of data should

be viewed with some skepticism, we have to set aside methods discovered before 1968, namely Ridge, PLS and Regression Trees. This leaves us with only five analysable methods: Lasso, Enet, Random Forests, Gradient Boosted Regression Trees and Neural Networks.

Figure 22 shows the out-of-sample Spearman rho (i.e. a measure of purely statistical predictive power), the monthly returns as well as the 3-year Fama-French 3-factor alpha for each of these methods. We use the returns and alphas of a top/bottom decile long/short strategy, but the effects on rank-weighted strategies are almost exactly the same. Moreover, since monthly returns (in light blue) are very volatile, we superimpose a 3-year centred moving average (in dark blue) which serves as a visual aide.

On each subgraph, we have indicated the publication date and article of each Machine Learning method. Since patterns can always be read into tea leaves, we encourage the reader to try to identify a pattern that emerges systematically across methods. In particular, we should ask ourselves whether we would have seen pattern emerge if the discovery date had been placed randomly somewhere else on the time scale. The best candidate for such a decline would probably be Random Forests, but we can note that they were discovered just before the dot-com bubble crash and that their performance quickly recovered afterwards. In our view, no systematic pattern of post-publication decline emerges.

To verify the lack of a time trend, we then plot these graphs but aligned by publication dates in Figure 23. It shows the same measures as before, but shows years before and after publication on the horizontal axis. Again, we do not see any substantial decline in either of these measures, nor a consistent downward pattern. In fact, we can see quite well that most of the variation is due to exogenous factors, which is why the return and alpha graphs look like shifted versions of each other (because they are!).

So that our analysis does not rely entirely on visual inspection, we use a last test following the regression approach used by McLean and Pontiff (2016). They use a simple regression of returns from a factor strategy on a dummy equal to 1 after the publication of the strategy (and a constant). Since they do this while combining all factor strategies in a single regression, such a simple specification is enough to identify post-publication decline. Additionally, the authors test quite a few different specifications afterwards to check the robustness of their results.

However, because we would like to study post-publication decline for each methods individually, we have to use a slightly refined approach. Instead, we regress the returns from a given strategy on an indicator equal to 1 after publication, a constant and a linear time trend.

$$r_t(\mathcal{S}) = \beta \times \mathbb{1}_{t \geq T_{pub}(\mathcal{S})} + \gamma_0 + \gamma_1 \times t \tag{27}$$

In fact, we then also test a quadratic and even quintic (fifth-order polynomial) time trend for each method. Each regression includes monthly returns up to 15 years before and 15 years after the publication of a method. In each regression, we only report the coefficient on the dummy, β, and the associated standard errors, which we compute using a Newey-West autocorrelation robust estimator allowing up to $10 \times 12 = 120$ lags.

Results for returns of decile spread strategies are presented in Table A19. Clearly, the coefficient on the post-publication dummy is not significantly different from zero in most specifications. In fact, we generally see a decrease in significance as the order of the controlling time polynomial increases because the shape of the return curve is better approximated. The only exception to this rule are Random Forests, but we have discussed above why the dot-com bubble is probably the cause of this break-down rather than actual post-publication decline. Table A21 shows the same regressions for 3-year FF3 alphas and results are qualitatively the same: most regressions are insignificant, especially after correcting for higher order time trends, and Regression Trees are the only notably exceptions.

Figure 22: Score, Returns and Alpha against Publication Dates of various ML Methods

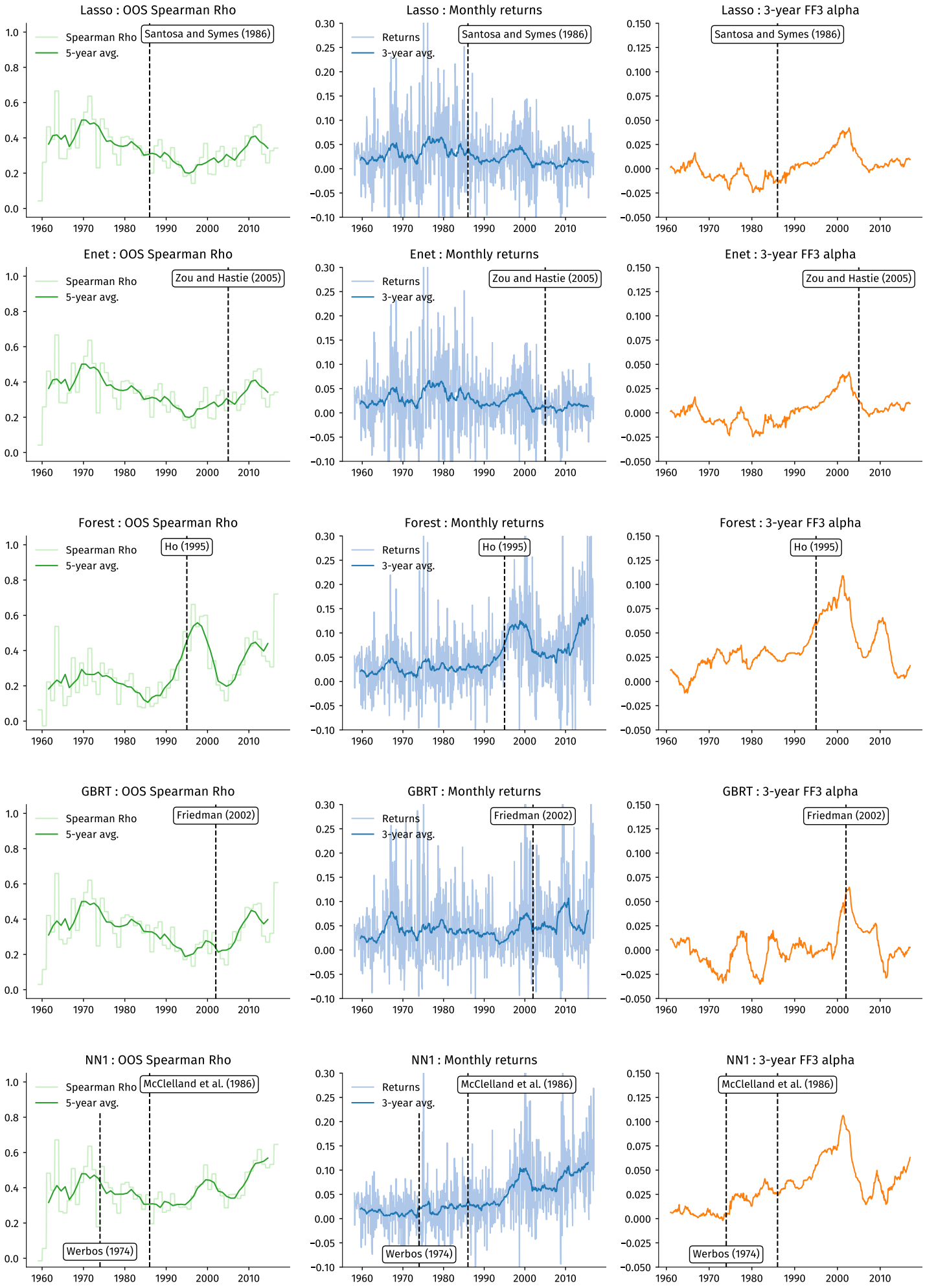
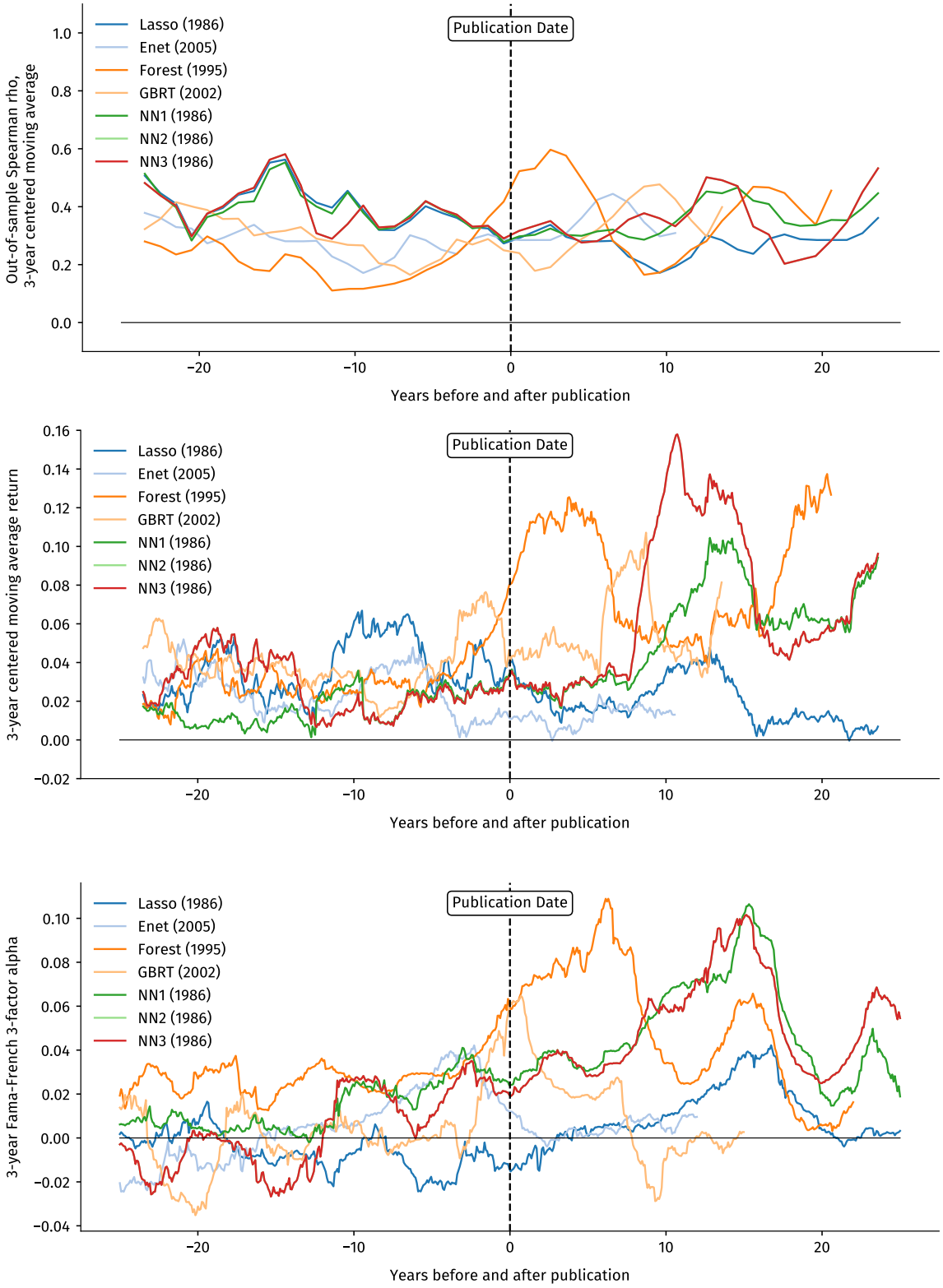


Figure 23: Before- and Post-Publication Score, Returns and Alpha of various ML Methods



Although we have previously focussed only on top/bottom decile long/short strategies, we present the same analyses for rank-weighted strategies in Tables A20 and A22. Again, the results are essentially a lack of results: most coefficients are insignificant, those that are stop being significant after a better polynomial control, and Regression Trees stand out. This clearly corroborates our initial lack of results.

Indeed, it should be stressed that this section only allows us to conclude on a lack of results: again, absence of evidence is not evidence of absence. More importantly, we do not claim to have found a magical statistical procedure to reject our null hypothesis,. However, we view these regressions as complementary to the previous analyses which showed no trend.

On the whole, we are thus confident enough to say that there is probably no post-publication decline for strategies built using the same data, the same forecasting architecture and more importantly the same reference years for the dissemination of ML methods. These are important caveats, and they should perhaps be seen less as caveats than as encouragements for future research: there are probably special setting or better identification strategies that will show at least *slight* publication decline in ML arbitrage strategies, and discovering them is an interesting challenge!

10. Machine Learning and Market Prescience

Before concluding, we explore a last aspect of the relationship between Machine Learning and arbitrage activity: the interplay between discoveries in the field learning and the general presence of informed arbitrageurs in the market. More precisely, we set out to analyse the relationship between the open short interest on a stock at the beginning of a month and a stock's actual return over that month. We decide to call this relationship *market prescience*, although *arbitrageur prescience* would perhaps have been more precise, since it describes correct predictions of returns by arbitrageurs that have not been correctly anticipated by the general market, which we assume to be short-sale restricted.

We can first study this as a cross-sectional phenomenon. Essentially, we reproduce our analysis of the interrelation between short interest and Machine Learning predictions, but taking actual returns instead of the predicted return. Again, if there are informed traders we would expect a negative relationship in Figure 24. In general, the curves seem quite flat, although there are some patterns emerging. However, it is hard to make out their precise direction in this cross-sectional representation.

Figure 25 plots the usual time series analyses we have used for Machine Learning predictions, but again using actual returns. We can see that the coefficient of market prescience varies from year-to-year and from measure-to-measure, but that it generally treads within negative territory. This is particularly true over the last years, i.e. after 1990, for which there are more observations. We read this result as indicating the existence of informed arbitrageurs correctly betting short on the stock market. However, given the low strength and significance of this relationship, it seems that their overall size is relatively small and that, to put it plainly, *they don't always get it right.*

Moreover, we have overlaid the publication dates of all in-sample Machine Learning methods, namely Neural Networks, Lasso, Random Forests, GBRT and Elastic Net. Here again readers are free to read infinitely complex patterns into the proverbial tea leaves, but on the whole it seems difficult to make out a simple, parsimonious pattern that is consistent across methods and measures. On this basis, we cautiously conclude that there does not seem to be an obvious relationship between the amount of informed short-sellers active in financial markets and Machine Learning research.

Finally, we can try to compare the magnitude of the relationship with Short interest between true returns and ML-predicted returns. To do so, we simply compute the Spearman's ρ between monthly predicted returns for each method and short interest, as well as between actual return and short interest. Since this is a non-parametric measure depending only on ranks, it ensures comparability

Figure 24: Cross-Section Analysis of Market Prescience i.e. Short Interest vs Actual Returns

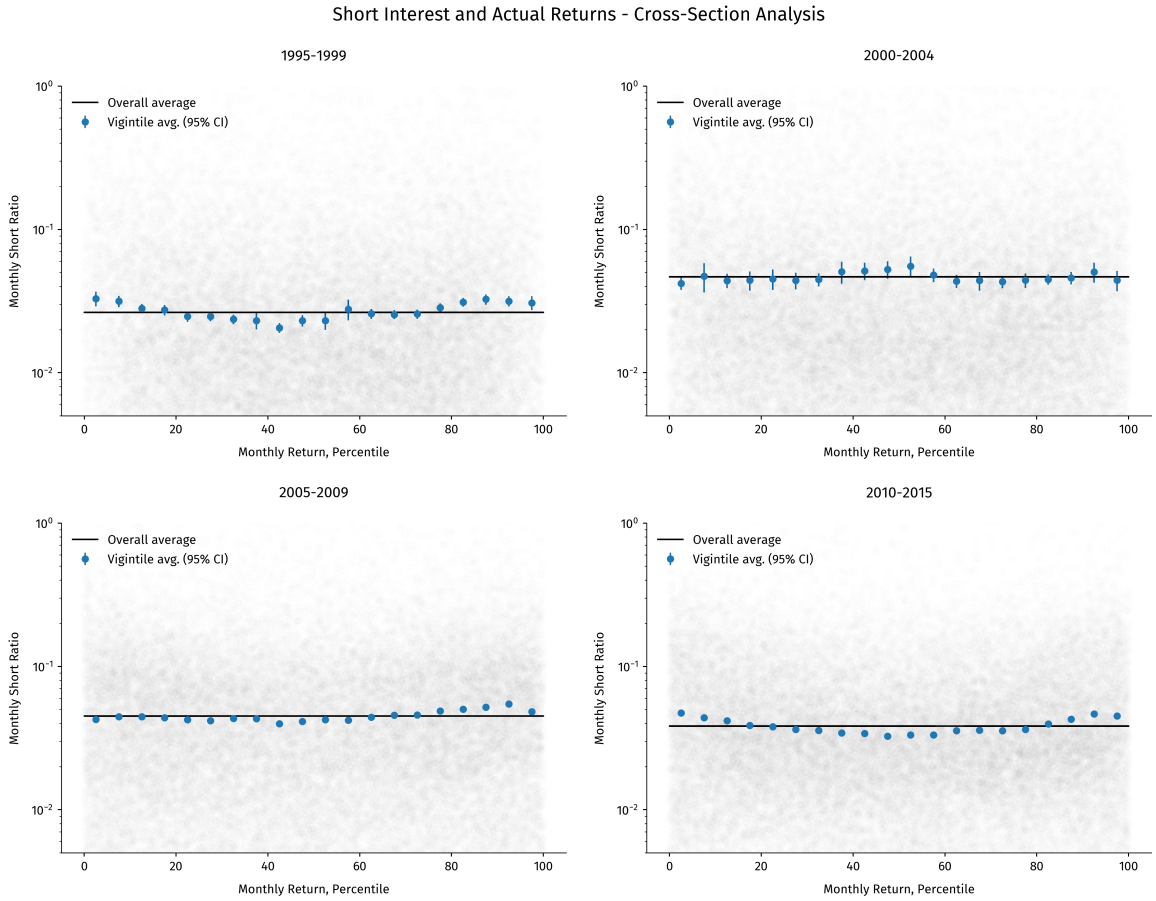
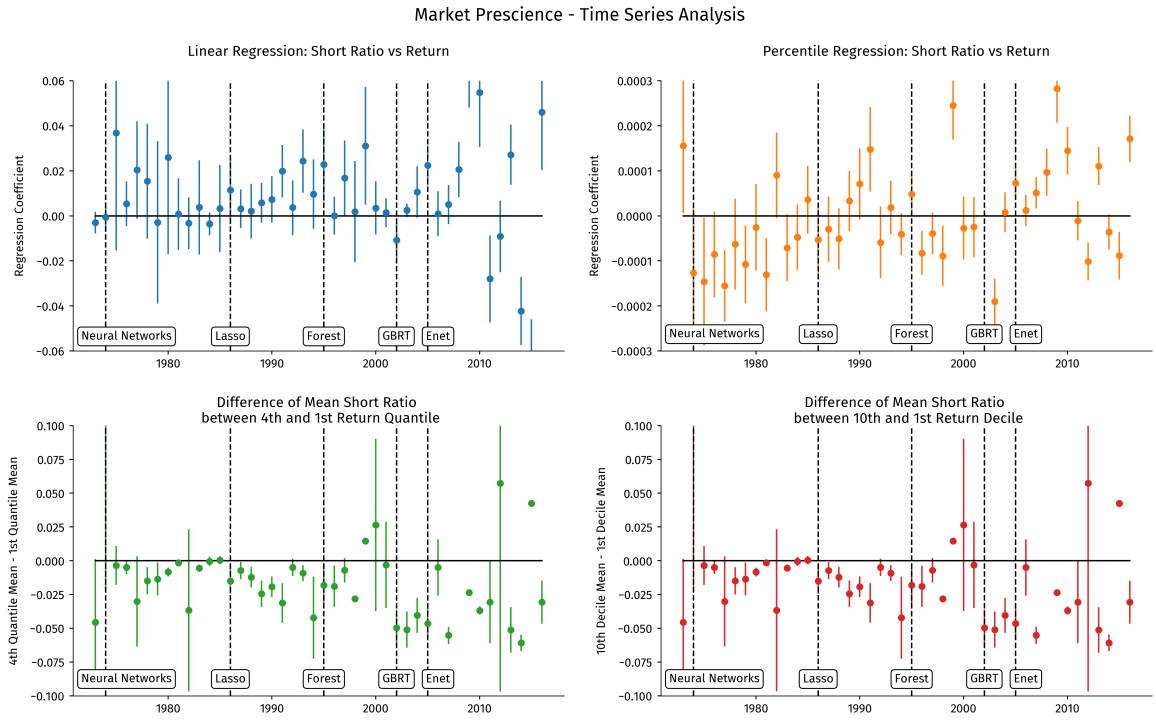
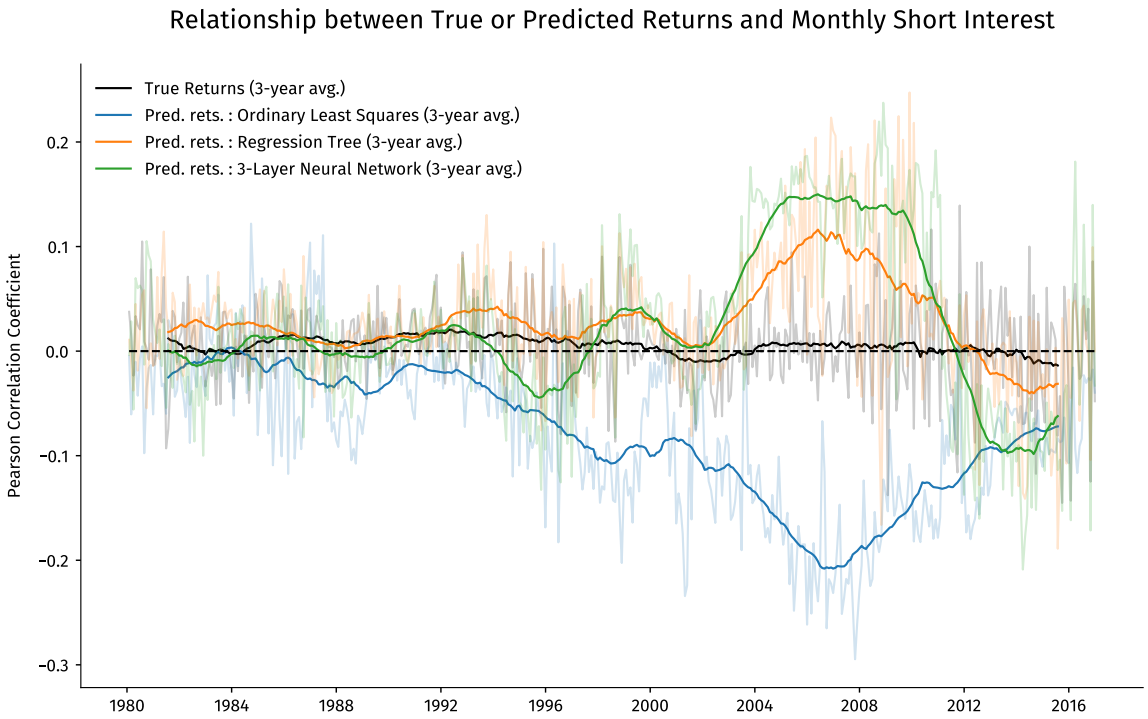


Figure 25: Time Series of Market Prescience and Machine Learning Discoveries



across measures. Results are plotted in Figure 26 : clearly, the positive association for Regression Trees and 3-Layer Neural Networks far outweigh the (generally still positive) association for actual returns. On the other hand, the association for OLS is very strongly negative, with roughly the same magnitude as the two other ML methods.

Figure 26: Associations with Short Interest of Predicted Returns and Actual Returns



This further suggests that the relationship between Machine Learning predictions and short interest does not reflect arbitrage activity, but some other characteristics of stocks that are selected (or anti-selected) by the ML algorithm. Indeed, if private algorithms in the form of ML techniques allowed us to explain a fraction of the private information available to arbitrageurs, we would expect the magnitude of the association between predictions and short interest to be (i) positive and (ii) a similar fraction of the association between true returns and short interest. We find the exact opposite.

Although, as always, absence of evidence should not be seen as evidence of absence, we see this as comforting the previous findings. Overall, this further suggests that Machine Learning does not represent an arbitrage opportunity, or at least that there are no informed agents exploiting Machine Learning techniques either today or in the past.

11. Conclusion

When we set out to study whether Machine Learning represents an arbitrage opportunity, we intended to proceed in two main steps: (i) show that Machine Learning outperforms the market by replicating the findings of Gu et al. (2020) and (ii) use the concept of *algorithmic market efficiency* to study whether there are arbitrageurs exploiting Machine Learning techniques, which would of course indicate that it indeed represents an arbitrage.

The first part of our endeavour has yielded very convincing results, showing that Machine Learning techniques can achieve higher returns than the market but also generate positive alphas in all workhorse asset pricing models. This is particularly true for more advanced, non-linear methods like Regression Trees and Neural Networks, here again confirming the findings of Gu et al. (2020).

The second part has proved, largely by design, deceptive: we have not uncovered evidence of arbitrageurs systematically exploiting Machine Learning techniques. Indeed, the association between ML-predictions and short interest is generally positive, i.e. counter to what we would expect, for the best-performing methods. Moreover, its magnitude is too large to represent a subset of the private information held by arbitrageurs in the market, which suggests that it rather represents features of selected stocks and not arbitrage activity in itself. Put simply, either markets are profoundly algorithmically inefficient or they are at least *semi-strongly* algorithmically efficient.

Moreover, we have not found any evidence of a post-publication decline : since Machine Learning research does not destroy stock return predictability, this strongly suggests that it does indeed learn to forecast risk loadings instead of arbitrage signals. Briefly put, markets appear to be *super-strongly* algorithmically efficient: if there are arbitrage signals, they are not easy to detect up with conventional ML techniques, or at least with our methods. In plain English, this means that markets incorporate information that cannot be entirely described, or *learned*, by current Machine Learning techniques. Conversely, it would also imply that the outperformance of Machine Learning techniques does not reflect arbitrage opportunities but simply latent risk loadings.

However, *semper idem*, absence of evidence is not evidence of absence. We have, by design, only tested fairly basic Machine Learning methods. Larger datasets, better predictive algorithms and more refined identification strategies could very well detect such arbitrage activity, and thus show that even strong algorithmic EMH does not hold. Finding creative avenues to do this seems like a fascinating avenue for future research.

For example, one could try to focus on price-only information and study whether the invention of Long-Short-Term Memory Networks by Hochreiter and Schmidhuber (1997) has led to the disappearance of patterns in historical stock prices. Or one could adopt a fund-level approach, in the giant's footsteps of Jensen (1968), and assess whether firms that had higher initial levels of machine-learning-trained workforce increased their returns more than others after the invention and publication of a powerful new ML method. Another promising topic for future research, that is as interesting as it is hard to study, is the development of secret, specialized Machine Learning algorithms by banks or hedge funds: these could have profound implications for market efficiency, but also for the financial system and even macro-financial stability as a whole.

A. Annex

A.1. Data and Data Sources

Figure A1: Factor variables from Green et al. (2017)

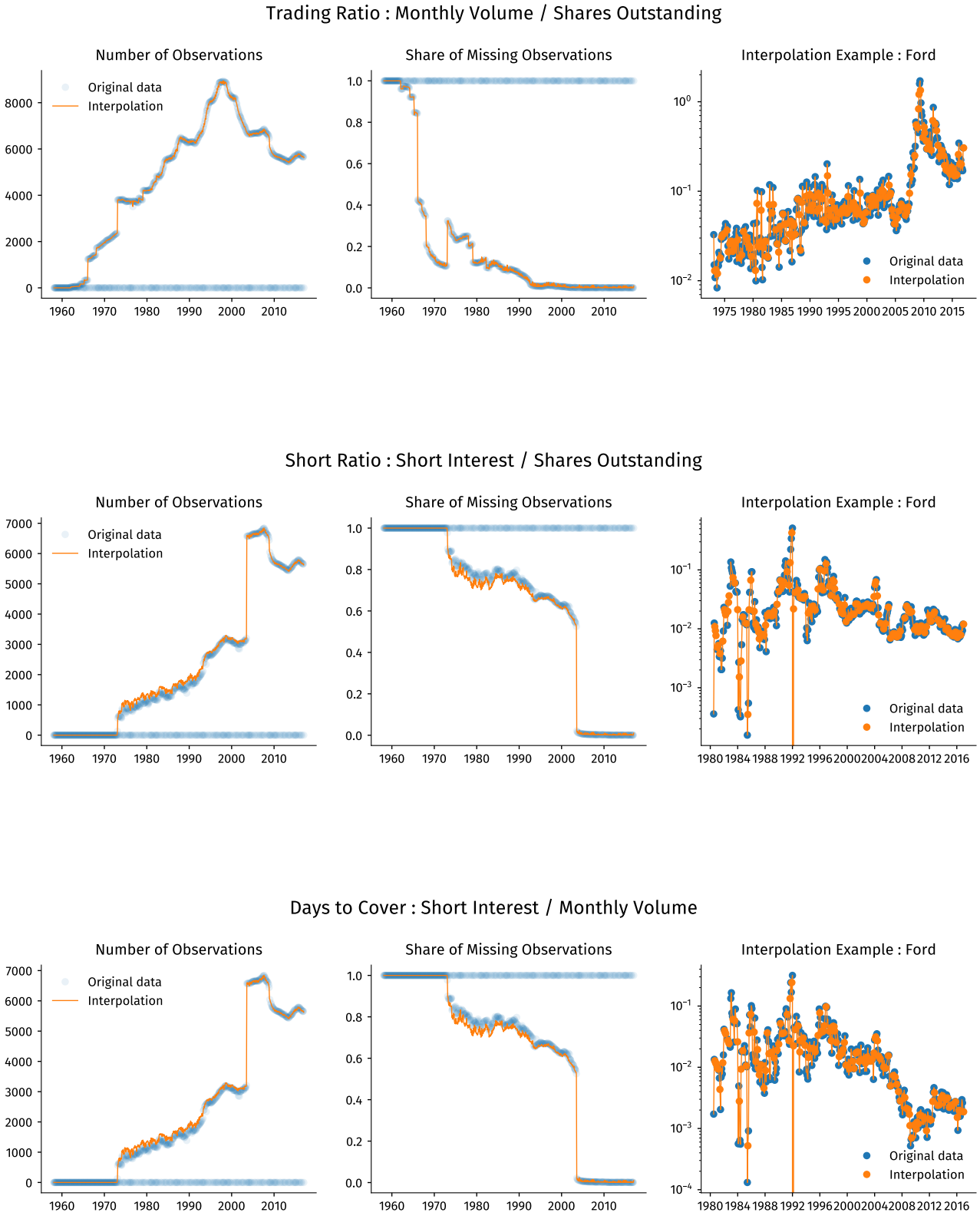
Explanatory table taken from Gu et al. (2020)

No.	Acronym	Firm characteristic	Paper's author(s)	Year, Journal	Data Source	Frequency
1	absacc	Absolute accruals	Bandyopadhyay, Huang & Wirjanto	2010, WP	Compustat	Annual
2	acc	Working capital accruals	Sloan	1996, TAR	Compustat	Annual
3	aeavol	Abnormal earnings announcement volume	Lerman, Livnat & Mendenhall	2007, WP	Compustat+CRSP	Quarterly
4	age	# years since first Compustat coverage	Jiang, Lee & Zhang	2005, RAS	Compustat	Annual
5	agr	Asset growth	Cooper, Gulen & Schill	2008, JF	Compustat	Annual
6	baspread	Bid-ask spread	Amihud & Mendelson	1989, JF	CRSP	Monthly
7	beta	Beta	Fama & MacBeth	1973, JPE	CRSP	Monthly
8	betasq	Beta squared	Fama & MacBeth	1973, JPE	CRSP	Monthly
9	bm	Book-to-market	Rosenberg, Reid & Lanstein	1985, JPM	Compustat+CRSP	Annual
10	bm_ia	Industry-adjusted book to market	Asness, Porter & Stevens	2000, WP	Compustat+CRSP	Annual
11	cash	Cash holdings	Palazzo	2012, JFE	Compustat	Quarterly
12	cashdebt	Cash flow to debt	Ou & Penman	1989, JAE	Compustat	Annual
13	cashpr	Cash productivity	Chandrashekar & Rao	2009, WP	Compustat	Annual
14	cfp	Cash flow to price ratio	Desai, Rajgopal & Venkatachalam	2004, TAR	Compustat	Annual
15	cfp_ia	Industry-adjusted cash flow to price ratio	Asness, Porter & Stevens	2000, WP	Compustat	Annual
16	chatoia	Industry-adjusted change in asset turnover	Soliman	2008, TAR	Compustat	Annual
17	chcsho	Change in shares outstanding	Pontiff & Woodgate	2008, JF	Compustat	Annual
18	chempia	Industry-adjusted change in employees	Asness, Porter & Stevens	1994, WP	Compustat	Annual
19	chinv	Change in inventory	Thomas & Zhang	2002, RAS	Compustat	Annual
20	chmom	Change in 6-month momentum	Gettleman & Marks	2006, WP	CRSP	Monthly
21	chpmia	Industry-adjusted change in profit margin	Soliman	2008, TAR	Compustat	Annual
22	chtx	Change in tax expense	Thomas & Zhang	2011, JAR	Compustat	Quarterly
23	cinvest	Corporate investment	Titman, Wei & Xie	2004, JFQA	Compustat	Quarterly
24	convind	Convertible debt indicator	Valta	2016, JFQA	Compustat	Annual
25	currat	Current ratio	Ou & Penman	1989, JAE	Compustat	Annual
26	depr	Depreciation / PP&E	Holthausen & Larcker	1992, JAE	Compustat	Annual
27	divi	Dividend initiation	Michaely, Thaler & Womack	1995, JF	Compustat	Annual
28	divo	Dividend omission	Michaely, Thaler & Womack	1995, JF	Compustat	Annual
29	dolvol	Dollar trading volume	Chordia, Subrahmanyam & Anshuman	2001, JFE	CRSP	Monthly
30	dy	Dividend to price	Litzenberger & Ramaswamy	1982, JF	Compustat	Annual
31	ear	Earnings announcement return	Kishore, Brandt, Santa-Clara & Venkatachalam	2008, WP	Compustat+CRSP	Quarterly

No.	Acronym	Firm characteristic	Paper's author(s)	Year, Journal	Data Source	Frequency
32	egr	Growth in common shareholder equity	Richardson, Sloan, Soliman & Tuna	2005, JAE	Compustat	Annual
33	ep	Earnings to price	Basu	1977, JF	Compustat	Annual
34	gma	Gross profitability	Novy-Marx	2013, JFE	Compustat	Annual
35	grCAPX	Growth in capital expenditures	Anderson & Garcia-Feijoo	2006, JF	Compustat	Annual
36	grltnoa	Growth in long term net operating assets	Fairfield, Whisenant & Yohn	2003, TAR	Compustat	Annual
37	herf	Industry sales concentration	Hou & Robinson	2006, JF	Compustat	Annual
38	hire	Employee growth rate	Bazdresch, Belo & Lin	2014, JPE	Compustat	Annual
39	idiovol	Idiosyncratic return volatility	Ali, Hwang & Trombley	2003, JFE	CRSP	Monthly
40	ill	Illiquidity	Amihud	2002, JFM	CRSP	Monthly
41	indmom	Industry momentum	Moskowitz & Grinblatt	1999, JF	CRSP	Monthly
42	invest	Capital expenditures and inventory	Chen & Zhang	2010, JF	Compustat	Annual
43	lev	Leverage	Bhandari	1988, JF	Compustat	Annual
44	lgr	Growth in long-term debt	Richardson, Sloan, Soliman & Tuna	2005, JAE	Compustat	Annual
45	maxret	Maximum daily return	Bali, Cakici & Whitelaw	2011, JFE	CRSP	Monthly
46	mom12m	12-month momentum	Jegadeesh	1990, JF	CRSP	Monthly
47	mom1m	1-month momentum	Jegadeesh & Titman	1993, JF	CRSP	Monthly
48	mom36m	36-month momentum	Jegadeesh & Titman	1993, JF	CRSP	Monthly
49	mom6m	6-month momentum	Jegadeesh & Titman	1993, JF	CRSP	Monthly
50	ms	Financial statement score	Mohanram	2005, RAS	Compustat	Quarterly
51	mvel1	Size	Banz	1981, JFE	CRSP	Monthly
52	mve_ia	Industry-adjusted size	Asness, Porter & Stevens	2000, WP	Compustat	Annual
53	nincr	Number of earnings increases	Barth, Elliott & Finn	1999, JAR	Compustat	Quarterly
54	operprof	Operating profitability	Fama & French	2015, JFE	Compustat	Annual
55	orgcap	Organizational capital	Eisfeldt & Papanikolaou	2013, JF	Compustat	Annual
56	pchcapx_ia	Industry adjusted % change in capital expenditures	Abarbanell & Bushee	1998, TAR	Compustat	Annual
57	pchcurrat	% change in current ratio	Ou & Penman	1989, JAE	Compustat	Annual
58	pchdepr	% change in depreciation	Holthausen & Larcker	1992, JAE	Compustat	Annual
59	pchgm_pchsale	% change in gross margin - % change in sales	Abarbanell & Bushee	1998, TAR	Compustat	Annual
60	pchquick	% change in quick ratio	Ou & Penman	1989, JAE	Compustat	Annual
61	pchsale_pchinvt	% change in sales - % change in inventory	Abarbanell & Bushee	1998, TAR	Compustat	Annual
62	pchsale_pchrect	% change in sales - % change in A/R	Abarbanell & Bushee	1998, TAR	Compustat	Annual

No.	Acronym	Firm characteristic	Paper's author(s)	Year, Journal	Data Source	Frequency
63	pchsale_pchxsga	% change in sales - % change in SG&A	Abarbanell & Bushee	1998, TAR	Compustat	Annual
64	pchsaleinv	% change sales-to-inventory	Ou & Penman	1989, JAE	Compustat	Annual
65	pctacc	Percent accruals	Hafzalla, Lundholm & Van Winkle	2011, TAR	Compustat	Annual
66	pricedelay	Price delay	Hou & Moskowitz	2005, RFS	CRSP	Monthly
67	ps	Financial statements score	Piotroski	2000, JAR	Compustat	Annual
68	quick	Quick ratio	Ou & Penman	1989, JAE	Compustat	Annual
69	rd	R&D increase	Eberhart, Maxwell & Siddique	2004, JF	Compustat	Annual
70	rd_mve	R&D to market capitalization	Guo, Lev & Shi	2006, JBFA	Compustat	Annual
71	rd_sale	R&D to sales	Guo, Lev & Shi	2006, JBFA	Compustat	Annual
72	realestate	Real estate holdings	Tuzel	2010, RFS	Compustat	Annual
73	retvol	Return volatility	Ang, Hodrick, Xing & Zhang	2006, JF	CRSP	Monthly
74	roaq	Return on assets	Balakrishnan, Bartov & Faurel	2010, JAE	Compustat	Quarterly
75	roavol	Earnings volatility	Francis, LaFond, Olsson & Schipper	2004, TAR	Compustat	Quarterly
76	roeq	Return on equity	Hou, Xue & Zhang	2015, RFS	Compustat	Quarterly
77	roic	Return on invested capital	Brown & Rowe	2007, WP	Compustat	Annual
78	rsup	Revenue surprise	Kama	2009, JBFA	Compustat	Quarterly
79	salecash	Sales to cash	Ou & Penman	1989, JAE	Compustat	Annual
80	saleinv	Sales to inventory	Ou & Penman	1989, JAE	Compustat	Annual
81	salerec	Sales to receivables	Ou & Penman	1989, JAE	Compustat	Annual
82	secured	Secured debt	Valta	2016, JFQA	Compustat	Annual
83	securedind	Secured debt indicator	Valta	2016, JFQA	Compustat	Annual
84	sgr	Sales growth	Lakonishok, Shleifer & Vishny	1994, JF	Compustat	Annual
85	sin	Sin stocks	Hong & Kacperczyk	2009, JFE	Compustat	Annual
86	sp	Sales to price	Barbee, Mukherji, & Raines	1996, FAJ	Compustat	Annual
87	std_dolvol	Volatility of liquidity (dollar trading volume)	Chordia, Subrahmanyam & Anshuman	2001, JFE	CRSP	Monthly
88	std_turn	Volatility of liquidity (share turnover)	Chordia, Subrahmanyam, &Anshuman	2001, JFE	CRSP	Monthly
89	stdacc	Accrual volatility	Bandyopadhyay, Huang & Wirjanto	2010, WP	Compustat	Quarterly
90	stdcf	Cash flow volatility	Huang	2009, JEF	Compustat	Quarterly
91	tang	Debt capacity/firm tangibility	Almeida & Campello	2007, RFS	Compustat	Annual
92	tb	Tax income to book income	Lev & Nissim	2004, TAR	Compustat	Annual
93	turn	Share turnover	Datar, Naik & Radcliffe	1998, JFM	CRSP	Monthly
94	zerotrade	Zero trading days	Liu	2006, JFE	CRSP	Monthly

Figure A2: Data Sources on Trading Volume and Short Interest



A.2. Predictive Performance of Machine Learning Algorithms

Figure A3: Cross-section of Predictions for Different Machine Learning Algorithms

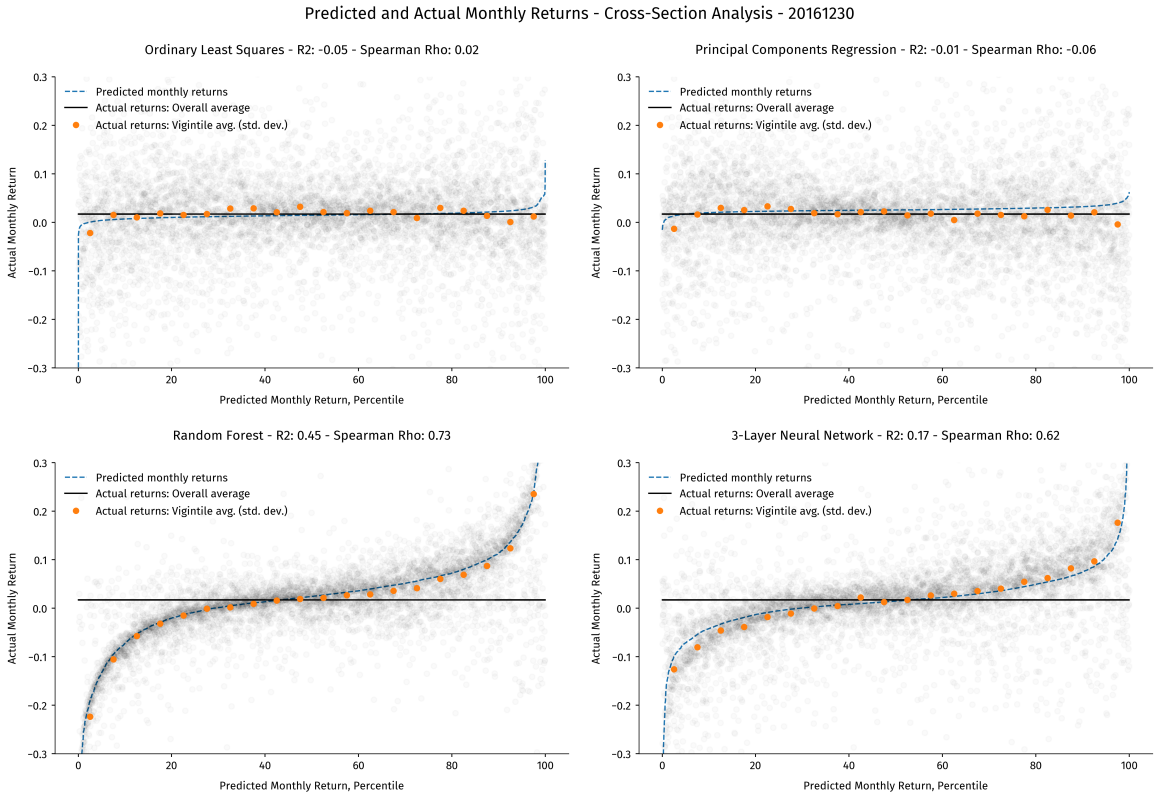


Figure A4: Time Series Analysis of In- and Out-of-sample Scores

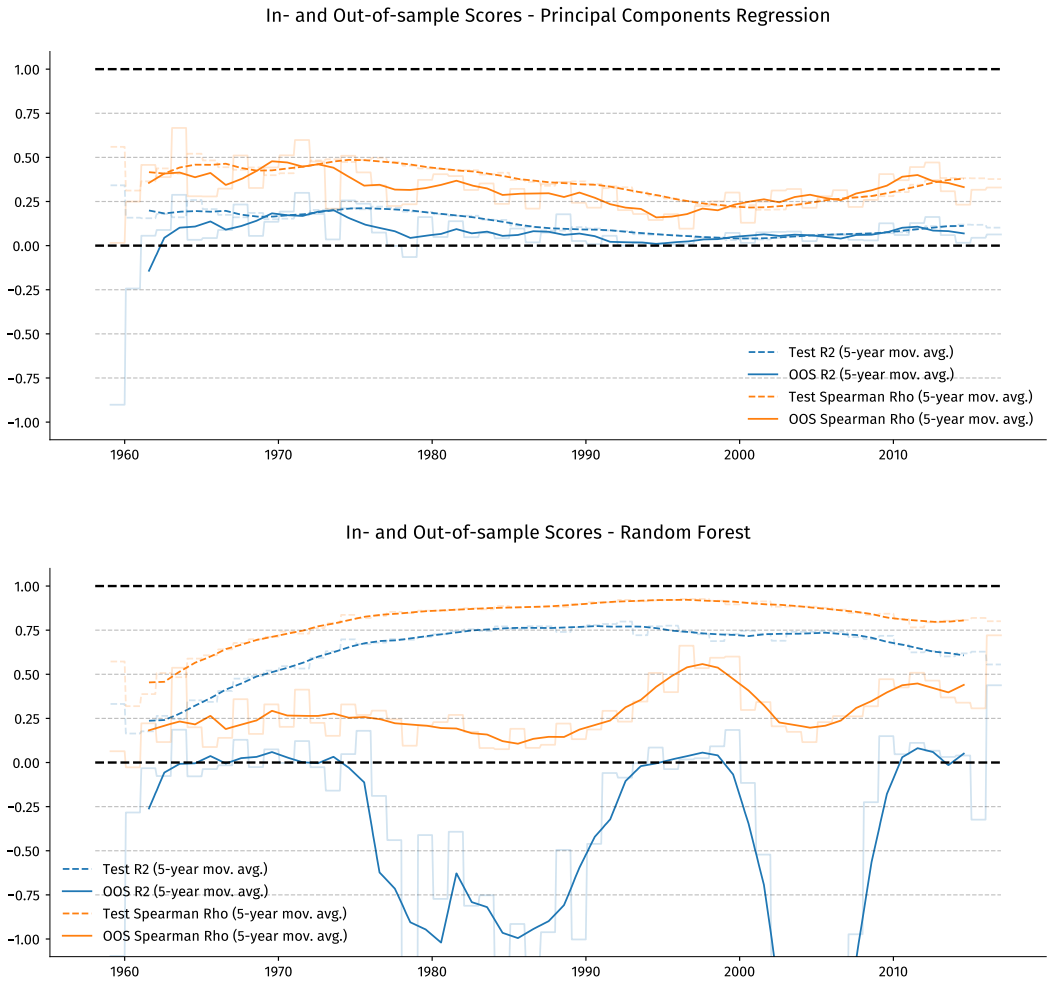


Figure A5: Non-Linearities and Interaction Effects in Machine Learning Predictors

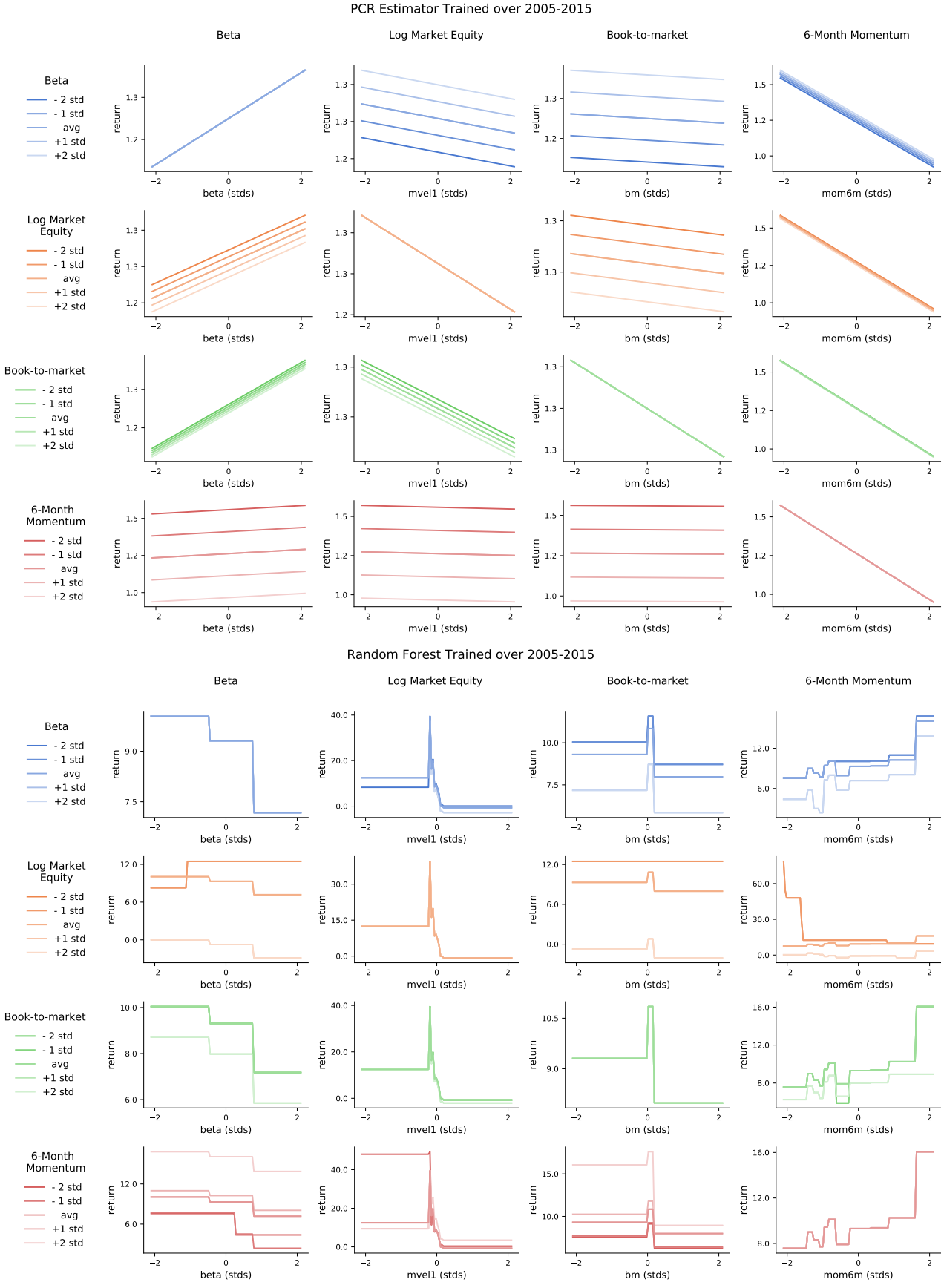


Figure A6: Percentage Decrease in Train Spearman ρ Induced by the Masking of each Factor

Figure A7: Percentage Decrease in Test R^2 Induced by the Masking of each Factor

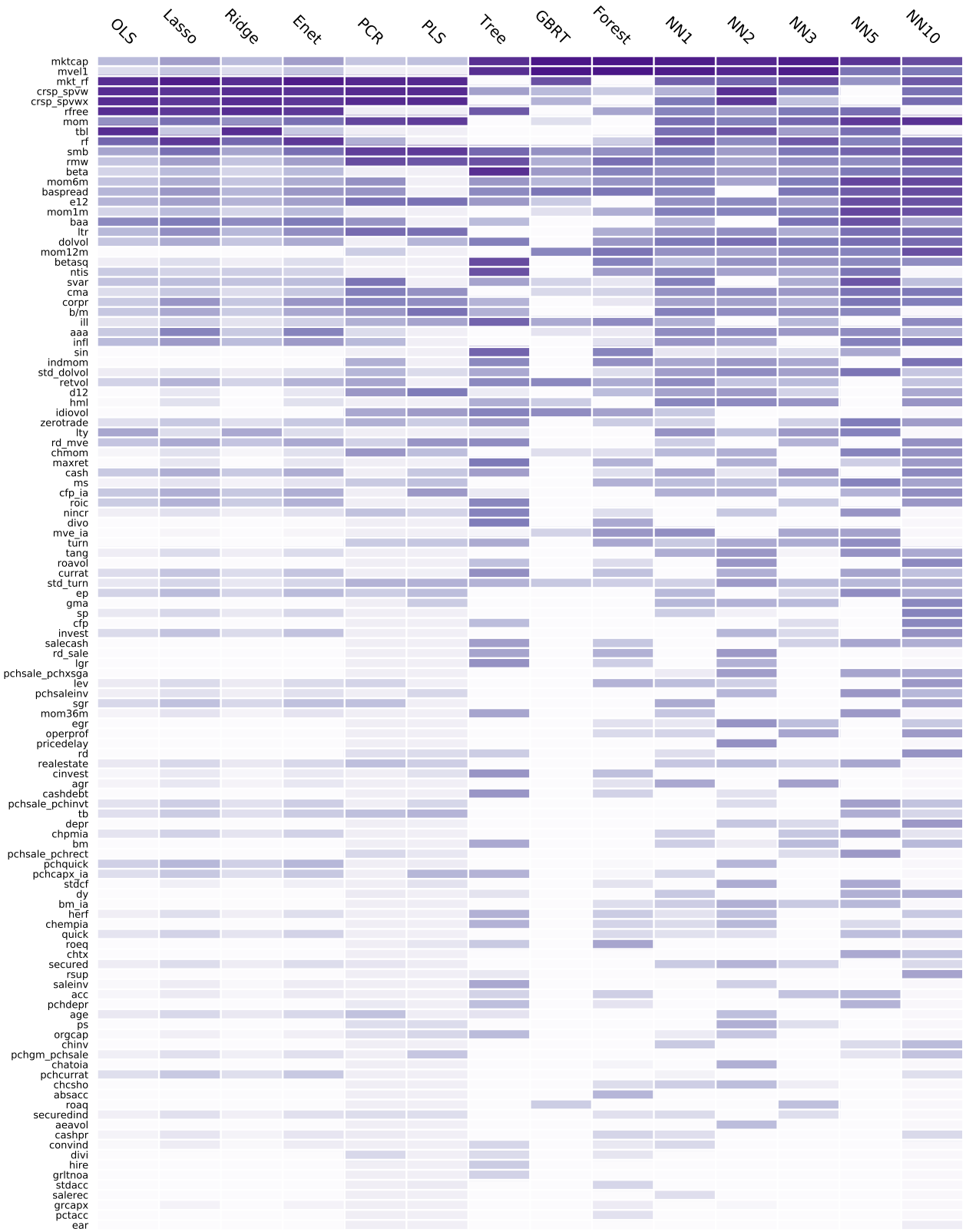


Figure A8: Percentage Decrease in Test Spearman ρ Induced by the Masking of each Factor

Figure A9: Percentage Decrease in Out-of-sample R^2 Induced by the Masking of each Factor

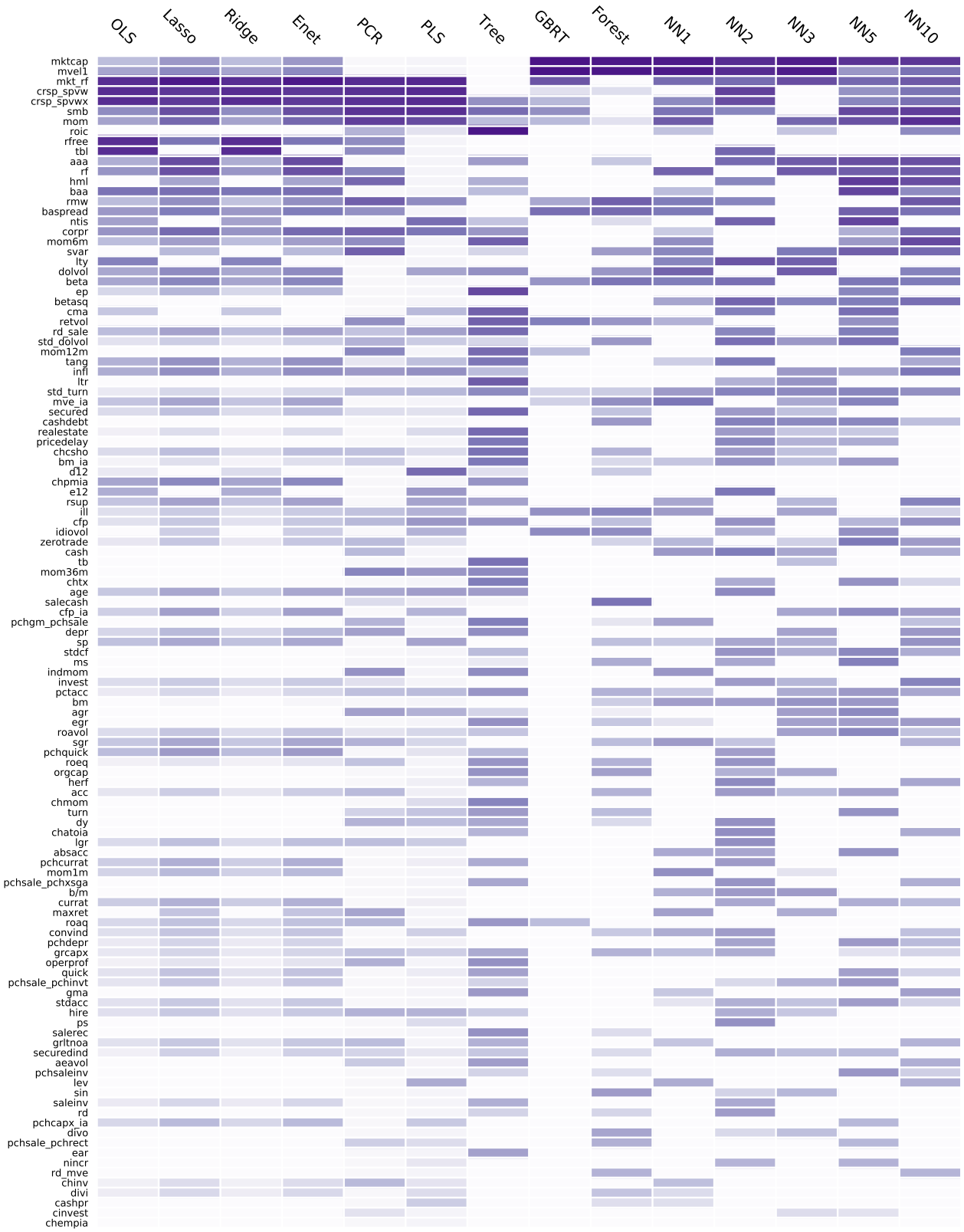
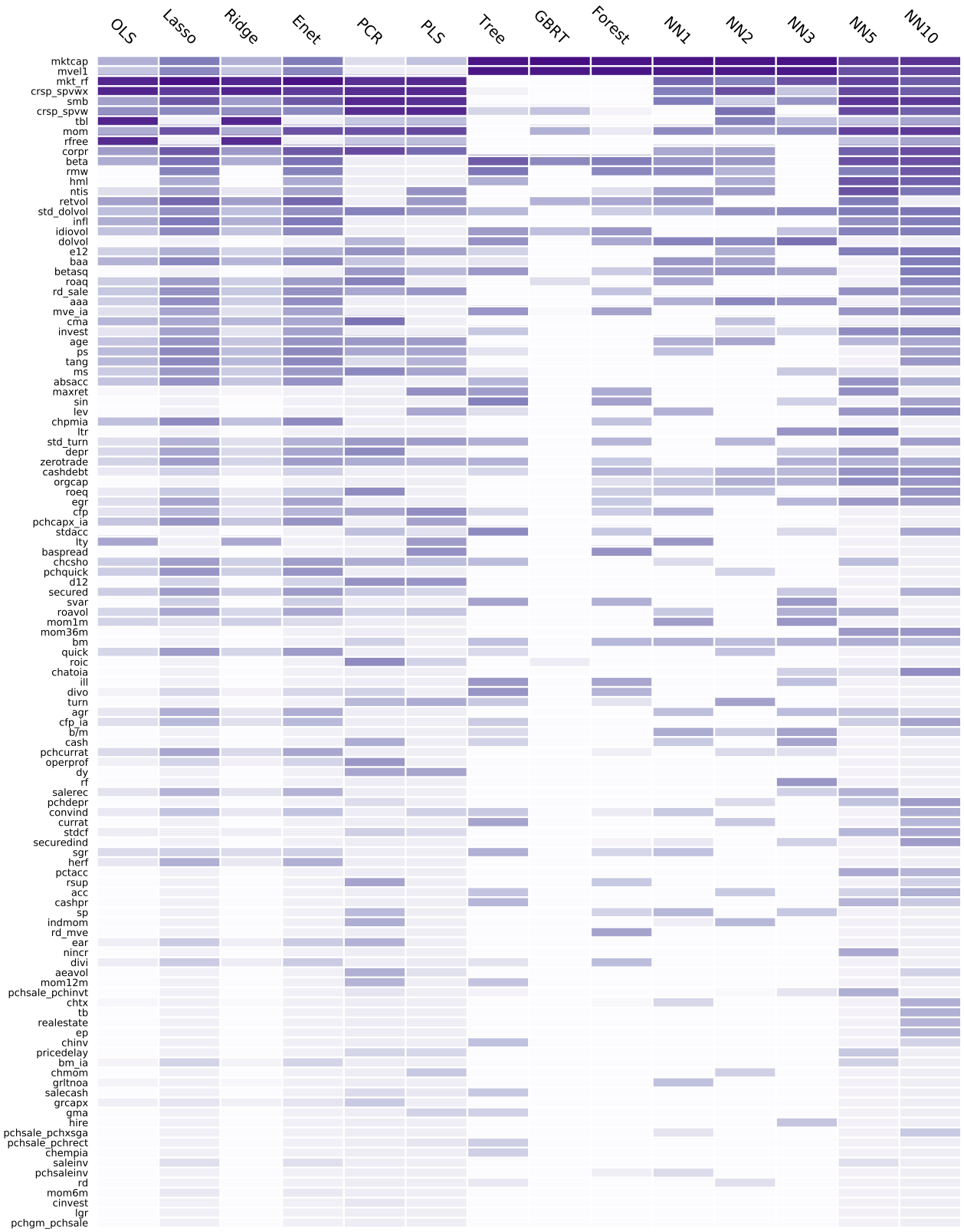


Figure A10: Percentage Decrease in Out-of-sample Spearman ρ Induced by the Masking of each Factor



A.3. Robustness Checks on Portfolio Performances

Table A7: Top Decile Long Strategies - Performance of ML Portfolios over 1980-2016

	1980-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	1.73	1.18	36	-19.0	-26.9	0.92	(8.10)	1.00	(6.38)	0.97	(6.82)	0.97	(5.41)
Lasso	1.98	1.11	74	-20.0	-33.0	1.09	(5.62)	1.21	(4.91)	1.16	(4.74)	1.19	(5.02)
Ridge	1.95	1.10	73	-20.0	-32.8	1.07	(5.73)	1.19	(4.98)	1.14	(4.89)	1.17	(5.07)
Enet	1.98	1.11	74	-20.0	-33.1	1.09	(5.63)	1.21	(4.92)	1.16	(4.77)	1.18	(5.01)
PCR	2.00	1.16	68	-22.0	-35.1	1.11	(5.89)	1.18	(5.42)	1.11	(5.58)	1.15	(4.74)
PLS	3.61	1.42	79	-30.4	-46.4	2.70	(5.52)	2.80	(5.28)	3.00	(4.65)	3.28	(5.07)
Tree	4.32	2.27	160	-17.2	-32.4	3.39	(5.06)	3.39	(5.13)	3.48	(5.44)	3.59	(5.60)
Forest	6.25	2.64	149	-18.0	-27.3	5.37	(4.14)	5.36	(4.15)	5.49	(4.31)	5.70	(4.45)
GBRT	4.52	1.97	137	-11.9	-19.4	3.66	(4.94)	3.57	(5.54)	3.93	(5.79)	3.79	(5.71)
NN1	5.88	2.92	94	-16.7	-23.9	4.99	(3.64)	5.01	(3.77)	5.15	(3.80)	5.19	(3.94)
NN2	7.23	3.10	100	-16.2	-23.6	6.43	(3.65)	6.48	(3.71)	6.55	(3.72)	6.74	(3.81)
NN3	7.66	3.04	107	-16.1	-24.7	6.83	(3.69)	6.86	(3.72)	7.06	(3.80)	7.15	(3.83)
NN5	3.81	1.85	110	-14.7	-26.0	2.90	(11.00)	2.88	(12.12)	3.13	(12.25)	2.96	(11.47)
NN10	4.30	1.87	126	-21.2	-33.3	3.38	(10.12)	3.41	(10.88)	3.62	(12.52)	3.50	(10.48)

Table A8: Top Decile Long Strategies - Performance of ML Portfolios over 2000-2016

	2000-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	1.16	0.95	28	-10.3	-25.4	0.82	(9.44)	0.91	(6.31)	0.89	(6.13)	0.88	(5.96)
Lasso	1.09	0.84	25	-11.2	-33.0	0.74	(5.50)	0.85	(3.63)	0.83	(3.51)	0.89	(3.77)
Ridge	1.09	0.84	25	-11.0	-32.8	0.74	(5.79)	0.85	(3.77)	0.83	(3.68)	0.89	(3.87)
Enet	1.09	0.84	25	-11.4	-33.1	0.73	(5.73)	0.84	(3.74)	0.82	(3.65)	0.88	(3.83)
PCR	1.24	0.86	52	-12.8	-35.1	0.87	(5.85)	0.94	(4.53)	0.91	(4.18)	0.87	(3.81)
PLS	3.52	1.28	75	-30.4	-46.4	3.30	(3.54)	3.31	(3.28)	3.45	(3.52)	3.99	(3.40)
Tree	4.44	2.22	176	-13.7	-32.4	4.08	(10.25)	3.95	(10.48)	4.00	(10.58)	4.16	(10.10)
Forest	7.58	2.90	176	-12.1	-27.3	7.31	(6.51)	7.12	(5.96)	7.18	(5.94)	7.45	(5.86)
GBRT	5.81	2.20	153	-11.9	-19.4	5.54	(11.57)	5.19	(12.31)	5.38	(11.78)	5.39	(12.16)
NN1	8.25	4.04	130	-9.4	-12.7	7.88	(10.54)	7.72	(9.48)	7.80	(9.20)	7.81	(8.85)
NN2	9.46	4.15	127	-5.1	-7.7	9.19	(5.19)	9.06	(4.82)	9.10	(4.80)	9.35	(4.63)
NN3	9.53	3.99	128	-9.5	-11.4	9.23	(4.83)	9.13	(4.57)	9.23	(4.52)	9.41	(4.30)
NN5	3.83	1.83	119	-12.1	-26.0	3.46	(9.69)	3.37	(9.91)	3.52	(9.91)	3.37	(8.84)
NN10	4.30	2.01	129	-12.6	-19.3	3.94	(6.79)	3.83	(7.14)	3.92	(7.56)	3.92	(7.47)

Table A9: Top/Bottom Decile Long/Short Strategies - Performance of ML Portfolios over 1980-2016

	1980-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	1.87	1.07	52	-21.8	-32.4	1.43	(3.79)	1.66	(3.74)	1.42	(4.06)	1.68	(3.34)
Lasso	1.36	0.63	80	-25.2	-88.0	0.99	(1.40)	1.16	(1.42)	1.11	(1.57)	1.55	(1.92)
Ridge	1.38	0.64	79	-25.3	-88.6	1.00	(1.40)	1.17	(1.42)	1.12	(1.59)	1.55	(1.92)
Enet	1.36	0.63	80	-25.1	-87.4	0.99	(1.41)	1.16	(1.42)	1.09	(1.56)	1.54	(1.92)
PCR	0.16	0.01	63	-25.0	-91.2	-0.04	(-0.07)	0.10	(0.16)	0.11	(0.20)	0.65	(1.04)
PLS	-1.43	-0.57	52	-32.5	-99.9	-0.70	(-1.99)	-0.63	(-1.77)	-0.94	(-2.51)	-0.86	(-2.45)
Tree	6.25	2.37	158	-20.9	-31.3	5.57	(4.48)	5.80	(4.62)	5.87	(4.65)	6.03	(4.62)
Forest	9.06	2.76	145	-30.3	-31.8	8.42	(4.21)	8.70	(4.32)	8.75	(4.43)	9.02	(4.31)
GBRT	7.84	1.31	169	-30.0	-77.0	8.13	(3.32)	8.06	(3.33)	8.52	(3.11)	8.60	(3.33)
NN1	9.39	3.04	94	-21.2	-31.5	8.64	(4.81)	8.94	(5.00)	9.13	(5.08)	9.35	(5.05)
NN2	10.30	3.13	95	-24.7	-33.9	9.68	(4.46)	9.94	(4.64)	9.81	(4.60)	10.28	(4.61)
NN3	10.55	2.98	103	-25.0	-33.9	9.86	(4.55)	9.97	(4.59)	10.32	(4.76)	10.46	(4.60)
NN5	3.88	1.41	125	-29.4	-37.0	3.14	(6.79)	3.24	(5.75)	3.43	(8.03)	3.43	(6.81)
NN10	3.70	1.44	129	-32.8	-36.4	3.00	(6.02)	3.13	(5.20)	3.24	(5.94)	3.21	(5.87)

Table A10: Top/Bottom Decile Long/Short Strategies - Performance of ML Portfolios over 2000-2016

	2000-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	1.07	0.80	43	-14.0	-32.4	1.04	(3.94)	1.30	(4.36)	1.21	(4.67)	1.33	(5.40)
Lasso	1.51	1.05	41	-10.3	-32.7	1.28	(7.02)	1.53	(3.92)	1.47	(3.94)	1.65	(4.38)
Ridge	1.53	1.07	41	-10.0	-32.7	1.30	(7.49)	1.54	(4.08)	1.48	(4.16)	1.66	(4.46)
Enet	1.49	1.03	41	-9.9	-32.7	1.27	(6.88)	1.52	(3.89)	1.44	(3.89)	1.63	(4.30)
PCR	0.25	0.17	49	-20.2	-74.3	0.18	(0.58)	0.47	(0.85)	0.47	(0.92)	0.72	(1.57)
PLS	-1.41	-0.44	43	-32.5	-98.4	-0.59	(-0.97)	-0.49	(-0.84)	-0.57	(-1.05)	-0.57	(-1.21)
Tree	5.31	1.90	171	-20.9	-31.3	5.36	(2.73)	5.27	(2.86)	5.42	(3.05)	5.78	(3.11)
Forest	9.34	2.69	170	-15.3	-25.3	9.54	(4.67)	9.40	(4.87)	9.55	(5.22)	9.91	(5.06)
GBRT	13.33	2.70	160	-21.3	-34.7	13.84	(12.48)	13.56	(13.74)	13.97	(15.81)	14.37	(12.66)
NN1	11.10	3.50	124	-7.0	-9.9	11.19	(9.98)	11.22	(9.70)	11.41	(10.70)	11.96	(11.01)
NN2	12.79	3.83	116	-9.5	-13.1	12.91	(16.87)	12.82	(17.72)	12.82	(18.20)	13.40	(15.83)
NN3	11.94	3.41	118	-15.6	-21.0	12.00	(17.75)	11.79	(19.88)	12.09	(23.64)	12.48	(19.46)
NN5	2.82	1.06	130	-21.1	-37.0	2.76	(3.50)	2.65	(3.51)	2.87	(4.02)	2.92	(4.22)
NN10	2.52	1.11	130	-22.9	-36.4	2.43	(3.23)	2.39	(3.07)	2.54	(3.46)	2.76	(3.82)

Table A11: Rank-weighted Long/Short Strategies - Performance of ML Portfolios over 1980-2016

	1980-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	1.11	0.67	43	-11.4	-44.6	0.79	(2.05)	1.00	(2.26)	0.79	(2.35)	1.08	(2.26)
Lasso	0.92	0.51	59	-18.7	-62.0	0.56	(1.16)	0.76	(1.37)	0.63	(1.33)	0.99	(1.67)
Ridge	0.93	0.52	58	-18.8	-61.6	0.57	(1.17)	0.76	(1.37)	0.64	(1.35)	0.99	(1.67)
Enet	0.92	0.51	58	-18.7	-62.3	0.56	(1.15)	0.76	(1.36)	0.63	(1.31)	0.98	(1.65)
PCR	0.40	0.11	40	-17.7	-79.1	0.02	(0.04)	0.14	(0.30)	0.15	(0.38)	0.56	(1.33)
PLS	-0.91	-0.58	31	-27.0	-99.0	-0.60	(-1.89)	-0.42	(-1.32)	-0.69	(-2.24)	-0.51	(-1.56)
Tree	4.22	2.26	93	-10.9	-23.1	3.47	(4.50)	3.61	(4.62)	3.63	(4.71)	3.67	(4.58)
Forest	4.82	2.27	77	-27.0	-35.4	4.01	(4.34)	4.21	(4.46)	4.15	(4.55)	4.29	(4.30)
GBRT	0.97	0.43	88	-19.0	-79.7	0.43	(0.79)	0.48	(0.84)	0.53	(0.90)	0.89	(1.37)
NN1	5.27	2.48	63	-20.5	-37.1	4.40	(6.32)	4.59	(6.07)	4.56	(6.32)	4.86	(6.49)
NN2	5.57	2.69	57	-18.9	-31.5	4.75	(8.35)	4.88	(8.35)	4.87	(8.24)	5.10	(7.67)
NN3	5.49	2.59	59	-19.8	-33.6	4.67	(7.06)	4.78	(6.99)	4.83	(7.27)	5.06	(6.83)
NN5	2.92	1.60	83	-18.1	-25.4	2.23	(4.82)	2.36	(4.26)	2.37	(4.65)	2.44	(4.76)
NN10	3.11	1.67	89	-20.4	-29.9	2.46	(5.04)	2.56	(4.51)	2.61	(5.05)	2.70	(4.82)

Table A12: Rank-weighted Long/Short Strategies - Performance of ML Portfolios over 2000-2016

	2000-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	0.37	0.27	32	-10.9	-38.6	0.40	(1.63)	0.60	(2.08)	0.54	(2.17)	0.73	(2.90)
Lasso	0.77	0.62	36	-11.6	-25.5	0.67	(4.20)	0.90	(2.59)	0.84	(2.59)	1.11	(3.42)
Ridge	0.78	0.64	36	-10.9	-24.9	0.68	(4.28)	0.90	(2.61)	0.84	(2.63)	1.11	(3.43)
Enet	0.76	0.62	36	-12.2	-25.0	0.67	(4.10)	0.89	(2.57)	0.83	(2.56)	1.10	(3.38)
PCR	0.35	0.24	33	-17.7	-51.5	0.17	(0.83)	0.38	(1.04)	0.37	(1.12)	0.64	(2.14)
PLS	-1.21	-0.53	25	-27.0	-96.6	-0.69	(-1.44)	-0.54	(-1.18)	-0.61	(-1.41)	-0.45	(-1.27)
Tree	2.90	1.74	106	-10.9	-16.5	2.72	(2.59)	2.73	(2.60)	2.81	(2.79)	2.99	(2.93)
Forest	3.40	1.83	93	-11.0	-32.8	3.23	(3.01)	3.27	(2.95)	3.33	(3.15)	3.59	(3.26)
GBRT	0.95	0.51	82	-19.0	-79.7	0.79	(0.68)	0.86	(0.69)	0.89	(0.73)	1.37	(1.15)
NN1	4.05	2.05	73	-20.5	-37.1	3.83	(6.29)	3.91	(5.72)	3.94	(5.80)	4.38	(6.88)
NN2	5.26	2.64	58	-15.6	-31.5	5.05	(16.47)	5.10	(17.23)	5.13	(17.11)	5.52	(15.74)
NN3	4.75	2.36	54	-19.8	-33.6	4.53	(18.58)	4.55	(17.53)	4.63	(17.30)	5.02	(15.90)
NN5	1.79	1.14	90	-18.1	-25.4	1.60	(2.48)	1.63	(2.27)	1.72	(2.45)	1.75	(2.97)
NN10	2.17	1.21	93	-20.4	-29.9	2.01	(2.56)	2.00	(2.47)	2.09	(2.71)	2.21	(2.99)

Table A13: Top Decile Long Strategies - Top 1000 Market Capitalizations

	1958-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	1.36	0.87	27	-18.1	-36.2	0.61	(6.43)	0.72	(5.56)	0.70	(6.09)	0.72	(4.74)
Lasso	1.55	0.95	77	-17.9	-35.9	0.80	(7.73)	0.95	(6.70)	0.93	(6.13)	0.96	(5.56)
Ridge	1.54	0.94	76	-17.9	-35.9	0.79	(7.99)	0.95	(6.87)	0.94	(6.29)	0.95	(5.71)
Enet	1.54	0.94	77	-17.9	-35.9	0.79	(8.00)	0.94	(6.83)	0.93	(6.28)	0.95	(5.66)
PCR	1.55	0.92	81	-19.7	-32.0	0.78	(4.12)	0.91	(4.37)	0.84	(4.37)	0.92	(3.96)
PLS	1.66	0.96	74	-17.3	-41.5	0.89	(7.37)	1.00	(6.54)	0.88	(5.18)	1.06	(6.30)
Tree	1.81	1.03	171	-18.2	-50.1	1.03	(6.30)	1.08	(6.11)	1.07	(7.84)	1.16	(6.53)
Forest	2.37	1.23	174	-13.5	-32.5	1.57	(5.90)	1.67	(5.02)	1.60	(6.93)	1.83	(5.07)
GBRT	2.27	1.29	129	-12.2	-34.1	1.49	(7.56)	1.57	(7.32)	1.69	(7.66)	1.68	(6.71)
NN1	2.02	1.24	58	-16.6	-33.2	1.24	(4.17)	1.36	(4.46)	1.34	(4.78)	1.37	(4.41)
NN2	2.11	1.20	79	-17.0	-38.1	1.34	(4.14)	1.48	(4.37)	1.40	(4.42)	1.60	(4.30)
NN3	2.18	1.27	86	-18.3	-69.5	1.40	(3.90)	1.49	(4.12)	1.53	(4.08)	1.50	(4.25)
NN5	2.17	1.18	83	-17.8	-33.2	1.36	(5.78)	1.44	(5.21)	1.51	(5.61)	1.59	(4.92)
NN10	2.42	1.24	96	-23.9	-40.7	1.59	(5.76)	1.70	(5.02)	1.83	(5.32)	1.91	(4.88)

Table A14: Top Decile Long Strategies - Top 100 Market Capitalizations

	1958-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	1.56	1.05	43	-18.5	-31.4	0.78	(9.21)	0.85	(7.90)	0.81	(8.63)	0.85	(7.12)
Lasso	1.99	1.19	93	-19.6	-32.3	1.18	(6.35)	1.23	(7.38)	1.15	(7.38)	1.22	(6.70)
Ridge	2.00	1.20	93	-19.5	-32.3	1.19	(6.36)	1.24	(7.34)	1.17	(7.42)	1.23	(6.72)
Enet	1.98	1.18	94	-19.6	-32.4	1.17	(6.45)	1.22	(7.44)	1.14	(7.42)	1.21	(6.77)
PCR	1.73	1.03	78	-21.5	-37.1	0.91	(9.77)	0.94	(9.33)	0.87	(9.82)	0.92	(7.78)
PLS	1.91	1.11	75	-17.4	-31.5	1.09	(8.75)	1.12	(7.63)	1.00	(5.43)	1.21	(7.14)
Tree	3.43	1.72	170	-16.6	-28.8	2.63	(3.05)	2.64	(3.03)	2.71	(3.22)	2.81	(3.14)
Forest	4.68	1.99	169	-17.9	-27.0	3.89	(3.12)	3.97	(3.03)	4.04	(3.16)	4.21	(3.13)
GBRT	3.16	1.66	137	-11.5	-27.1	2.34	(8.28)	2.33	(8.72)	2.51	(9.63)	2.46	(8.80)
NN1	5.49	2.39	87	-16.5	-29.9	4.78	(2.49)	4.85	(2.53)	5.09	(2.56)	4.91	(2.58)
NN2	6.61	2.36	107	-16.2	-30.5	5.98	(2.61)	6.13	(2.66)	6.27	(2.72)	6.31	(2.74)
NN3	6.76	2.49	116	-15.9	-30.6	6.08	(2.66)	6.17	(2.68)	6.48	(2.76)	6.27	(2.71)
NN5	2.99	1.60	93	-14.0	-26.7	2.15	(5.38)	2.17	(5.39)	2.34	(5.39)	2.25	(5.31)
NN10	3.44	1.74	114	-20.3	-29.6	2.57	(6.79)	2.61	(6.78)	2.74	(6.99)	2.73	(6.68)

Table A15: Top Decile Long Strategies - Top 1000 Market Capitalizations

	1958-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	1.94	1.20	58	-15.8	-26.7	1.28	(5.43)	1.39	(4.77)	1.32	(5.67)	1.48	(4.45)
Lasso	1.45	0.73	99	-20.5	-49.7	1.07	(3.22)	1.23	(3.00)	1.22	(3.39)	1.53	(3.34)
Ridge	1.47	0.75	99	-20.6	-44.1	1.08	(3.27)	1.25	(3.08)	1.24	(3.49)	1.54	(3.39)
Enet	1.43	0.72	99	-20.4	-50.2	1.05	(3.11)	1.22	(2.93)	1.21	(3.32)	1.51	(3.26)
PCR	0.79	0.35	79	-20.2	-88.0	0.42	(1.06)	0.56	(1.20)	0.55	(1.31)	0.87	(1.64)
PLS	0.37	0.10	66	-31.6	-84.4	0.32	(0.82)	0.42	(0.85)	0.23	(0.44)	0.73	(1.35)
Tree	3.15	1.70	172	-14.1	-34.0	2.82	(3.33)	2.81	(3.34)	2.80	(3.40)	2.71	(3.53)
Forest	4.85	2.14	168	-23.5	-23.5	4.42	(3.13)	4.54	(3.18)	4.29	(3.24)	4.38	(3.19)
GBRT	3.13	0.89	170	-51.0	-74.5	3.56	(2.84)	3.56	(2.77)	4.09	(2.84)	4.11	(2.87)
NN1	5.14	1.81	79	-24.7	-54.2	5.07	(3.22)	5.43	(3.44)	5.10	(3.27)	4.81	(3.15)
NN2	5.49	1.78	97	-24.4	-65.1	5.59	(2.64)	5.93	(2.81)	5.61	(2.72)	5.31	(2.60)
NN3	6.18	1.81	106	-21.7	-72.6	6.41	(2.44)	6.74	(2.61)	6.54	(2.55)	6.23	(2.47)
NN5	2.37	1.03	111	-27.9	-38.8	1.79	(5.09)	1.88	(4.79)	2.01	(5.74)	1.87	(4.56)
NN10	2.23	1.02	119	-33.1	-40.1	1.65	(5.01)	1.86	(4.81)	1.86	(5.07)	1.89	(4.91)

Table A16: Top/Bottom Decile Long/Short Strategies - Top 100 Market Capitalizations

	1958-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	1.95	1.01	41	-24.2	-46.1	1.18	(6.51)	1.36	(5.50)	1.32	(6.15)	1.38	(4.84)
Lasso	1.74	0.87	88	-23.6	-44.9	1.16	(4.61)	1.37	(4.35)	1.42	(4.83)	1.54	(4.32)
Ridge	1.79	0.90	87	-24.6	-44.8	1.20	(4.87)	1.41	(4.53)	1.47	(5.15)	1.58	(4.43)
Enet	1.72	0.85	88	-24.6	-45.3	1.14	(4.58)	1.35	(4.31)	1.40	(4.82)	1.53	(4.27)
PCR	0.77	0.34	82	-28.9	-89.4	0.34	(0.69)	0.52	(0.98)	0.47	(0.95)	0.70	(1.19)
PLS	0.82	0.37	72	-23.0	-56.5	0.47	(1.41)	0.53	(1.27)	0.45	(0.97)	0.84	(1.84)
Tree	0.49	0.17	181	-29.2	-64.4	0.32	(1.67)	0.30	(1.66)	0.36	(1.92)	0.32	(1.73)
Forest	1.68	0.92	177	-19.8	-46.2	1.36	(3.24)	1.37	(3.11)	1.27	(4.15)	1.41	(3.35)
GBRT	1.38	0.55	165	-95.9	-99.7	2.02	(2.01)	1.85	(1.88)	2.57	(2.31)	1.90	(1.89)
NN1	2.04	1.03	66	-20.1	-69.9	1.56	(5.56)	1.74	(6.22)	1.51	(5.49)	1.53	(5.11)
NN2	1.92	0.92	91	-23.7	-65.0	1.59	(3.42)	1.76	(3.31)	1.50	(2.49)	1.50	(3.09)
NN3	2.17	1.05	94	-23.3	-50.8	1.80	(3.05)	2.00	(3.21)	1.81	(2.87)	1.82	(3.05)
NN5	1.98	0.88	104	-35.5	-46.3	1.44	(4.81)	1.51	(4.97)	1.47	(4.47)	1.54	(4.48)
NN10	1.83	0.86	108	-24.9	-45.2	1.33	(5.66)	1.48	(5.23)	1.48	(4.98)	1.42	(5.06)

Table A17: Rank-weighted Long/Short Strategies - Top 1000 Market Capitalizations

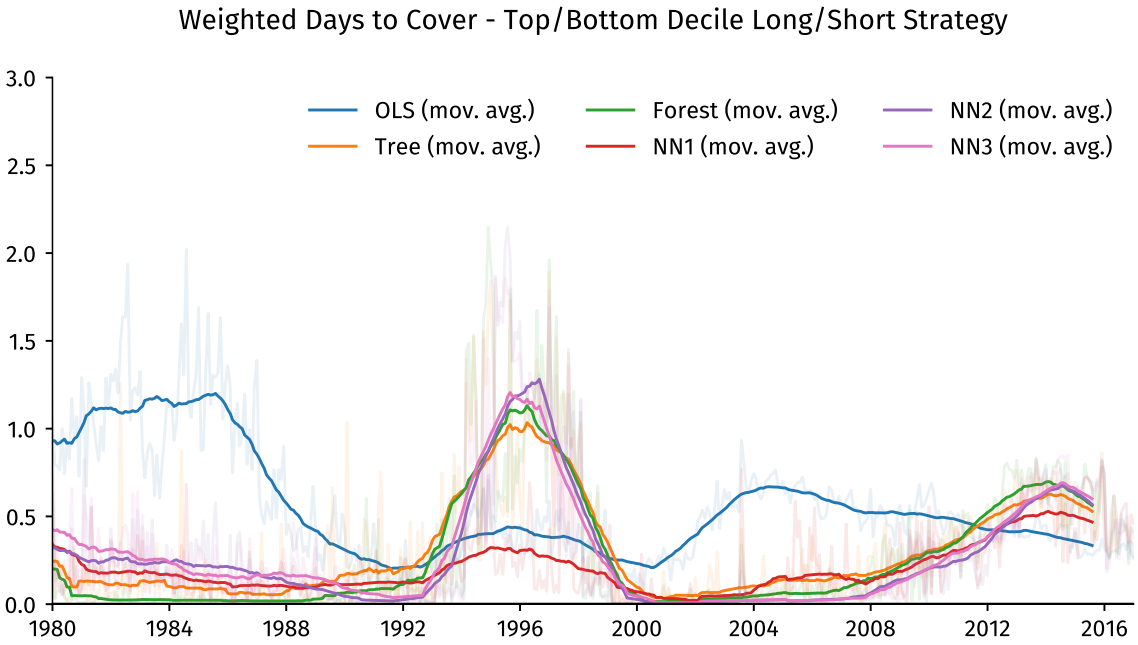
	1958-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	0.89	0.52	89	-12.4	-33.6	0.64	(3.82)	0.61	(4.19)	0.33	(2.51)	0.51	(3.37)
Lasso	0.96	0.56	96	-25.3	-58.2	0.71	(2.88)	0.79	(3.21)	0.53	(2.72)	0.89	(3.26)
Ridge	0.96	0.56	96	-25.2	-57.1	0.71	(2.88)	0.79	(3.19)	0.53	(2.71)	0.88	(3.23)
Enet	0.95	0.55	96	-25.3	-58.2	0.70	(2.89)	0.78	(3.20)	0.52	(2.70)	0.87	(3.23)
PCR	0.07	-0.19	69	-18.1	-84.7	-0.06	(-0.30)	0.05	(0.25)	-0.15	(-0.79)	0.18	(0.93)
PLS	-0.00	-0.20	57	-30.3	-86.0	-0.10	(-0.52)	-0.04	(-0.17)	-0.26	(-1.15)	0.07	(0.28)
Tree	5.02	2.15	121	-18.0	-81.2	4.82	(2.17)	4.83	(2.17)	4.94	(2.24)	5.05	(2.24)
Forest	6.16	2.47	115	-20.7	-62.4	6.03	(2.55)	6.04	(2.53)	6.16	(2.61)	6.26	(2.57)
GBRT	3.77	1.94	105	-17.3	-29.3	3.37	(5.63)	3.37	(6.11)	3.70	(6.56)	3.65	(6.06)
NN1	7.65	3.16	103	-6.0	-9.8	7.43	(3.31)	7.43	(3.29)	7.69	(3.37)	7.59	(3.39)
NN2	9.05	3.67	102	-6.8	-11.2	8.81	(4.08)	8.84	(4.03)	9.16	(4.25)	9.03	(4.11)
NN3	8.74	3.49	105	-5.0	-9.7	8.49	(3.88)	8.50	(3.83)	8.82	(4.08)	8.69	(3.88)
NN5	4.25	2.48	106	-8.9	-26.7	4.01	(6.97)	3.98	(6.99)	4.22	(7.29)	3.80	(7.08)
NN10	5.07	2.93	112	-5.5	-11.5	4.81	(9.29)	4.78	(9.58)	4.95	(10.64)	4.70	(9.96)

Table A18: Rank-weighted Long/Short Strategies - Top 1000 Market Capitalizations

	1958-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	-0.03	-0.18	78	-21.8	-91.9	0.13	(0.46)	0.26	(0.95)	-0.10	(-0.48)	0.22	(0.73)
Lasso	-0.00	-0.16	86	-28.9	-86.2	0.14	(0.46)	0.36	(1.07)	0.02	(0.08)	0.48	(1.27)
Ridge	0.01	-0.15	86	-27.9	-85.7	0.15	(0.49)	0.36	(1.08)	0.03	(0.11)	0.48	(1.27)
Enet	-0.01	-0.16	86	-29.8	-86.6	0.14	(0.46)	0.36	(1.06)	0.01	(0.05)	0.48	(1.25)
PCR	-0.37	-0.47	62	-27.9	-97.3	-0.32	(-1.41)	-0.19	(-0.82)	-0.39	(-1.97)	0.04	(0.15)
PLS	-0.94	-0.67	47	-32.4	-99.9	-0.66	(-2.33)	-0.57	(-2.09)	-0.82	(-3.04)	-0.54	(-2.01)
Tree	4.22	2.25	105	-11.9	-20.4	3.62	(3.25)	3.75	(3.41)	3.83	(3.67)	3.85	(3.54)
Forest	4.81	2.31	93	-19.8	-24.4	4.19	(3.39)	4.35	(3.53)	4.34	(3.69)	4.49	(3.71)
GBRT	1.21	0.57	100	-20.2	-70.9	0.76	(1.60)	0.88	(1.93)	0.96	(2.07)	1.21	(2.26)
NN1	5.61	2.41	89	-20.8	-34.5	5.16	(3.16)	5.36	(3.34)	5.56	(3.55)	5.68	(3.54)
NN2	5.56	2.35	88	-20.5	-39.9	5.15	(2.90)	5.27	(3.02)	5.49	(3.19)	5.50	(3.18)
NN3	5.44	2.30	93	-13.3	-43.1	5.04	(2.93)	5.14	(3.05)	5.35	(3.17)	5.43	(3.21)
NN5	3.57	1.98	103	-15.6	-16.3	3.22	(4.19)	3.32	(4.39)	3.53	(4.78)	3.26	(4.91)
NN10	4.08	2.20	108	-16.4	-20.1	3.75	(4.28)	3.85	(4.45)	4.01	(4.83)	3.89	(4.97)

A.4. Short Interest around Machine Learning Portfolios

Figure A11: Weighted Days to Cover in Selected ML Portfolios



Weighted Days to Cover - Top/Bottom Decile Long/Short Strategy

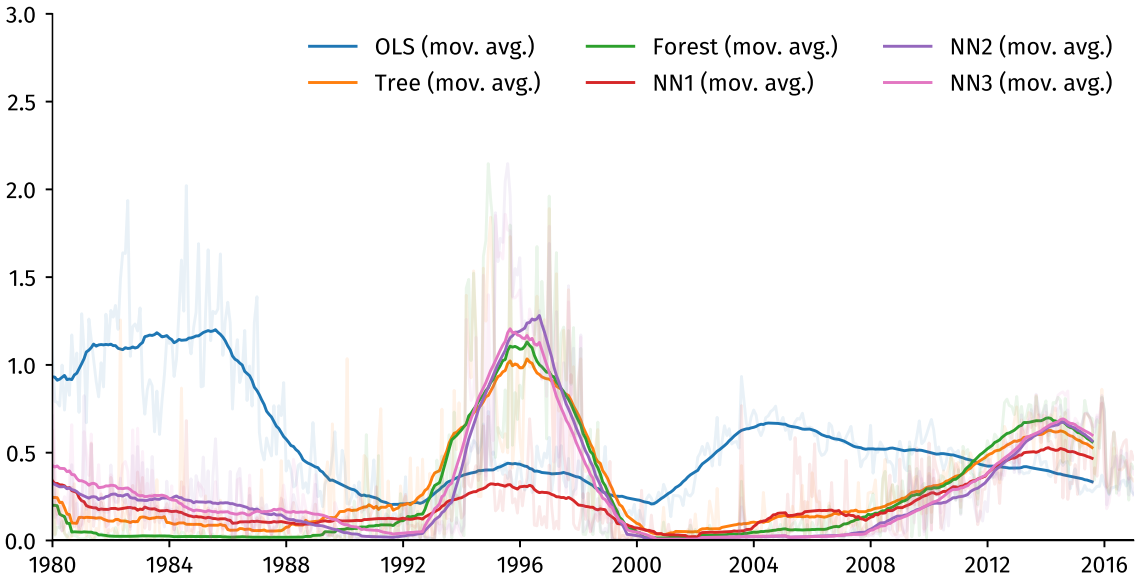
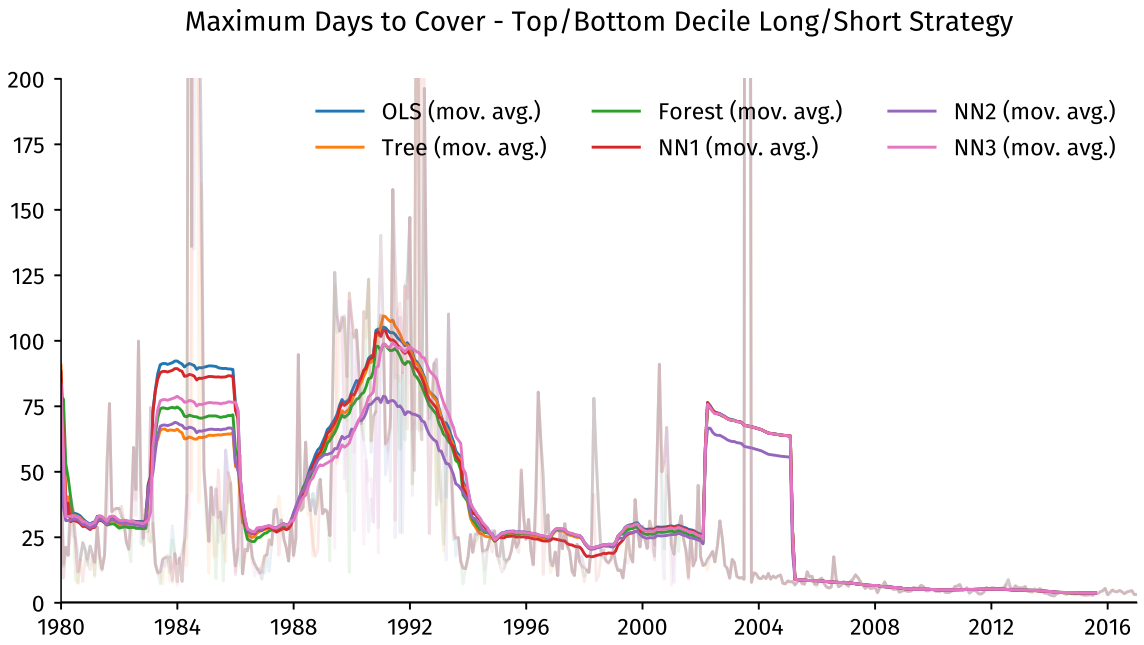


Figure A12: Maximum Days to Cover in Selected ML Portfolios



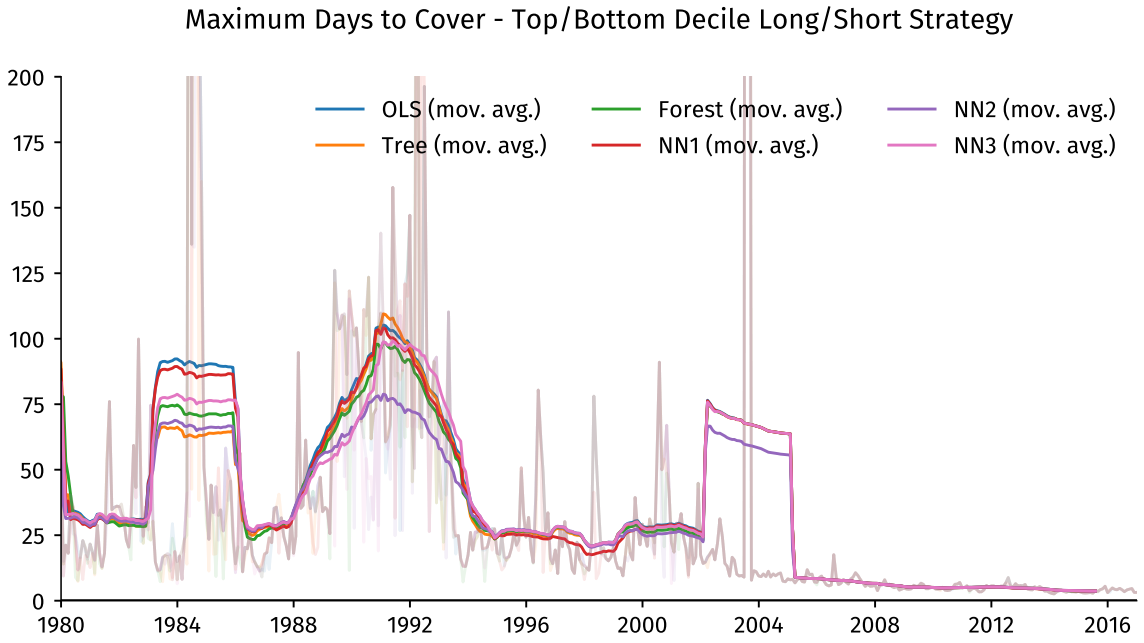


Figure A13: Weighted Trading Ratio in Selected ML Portfolios

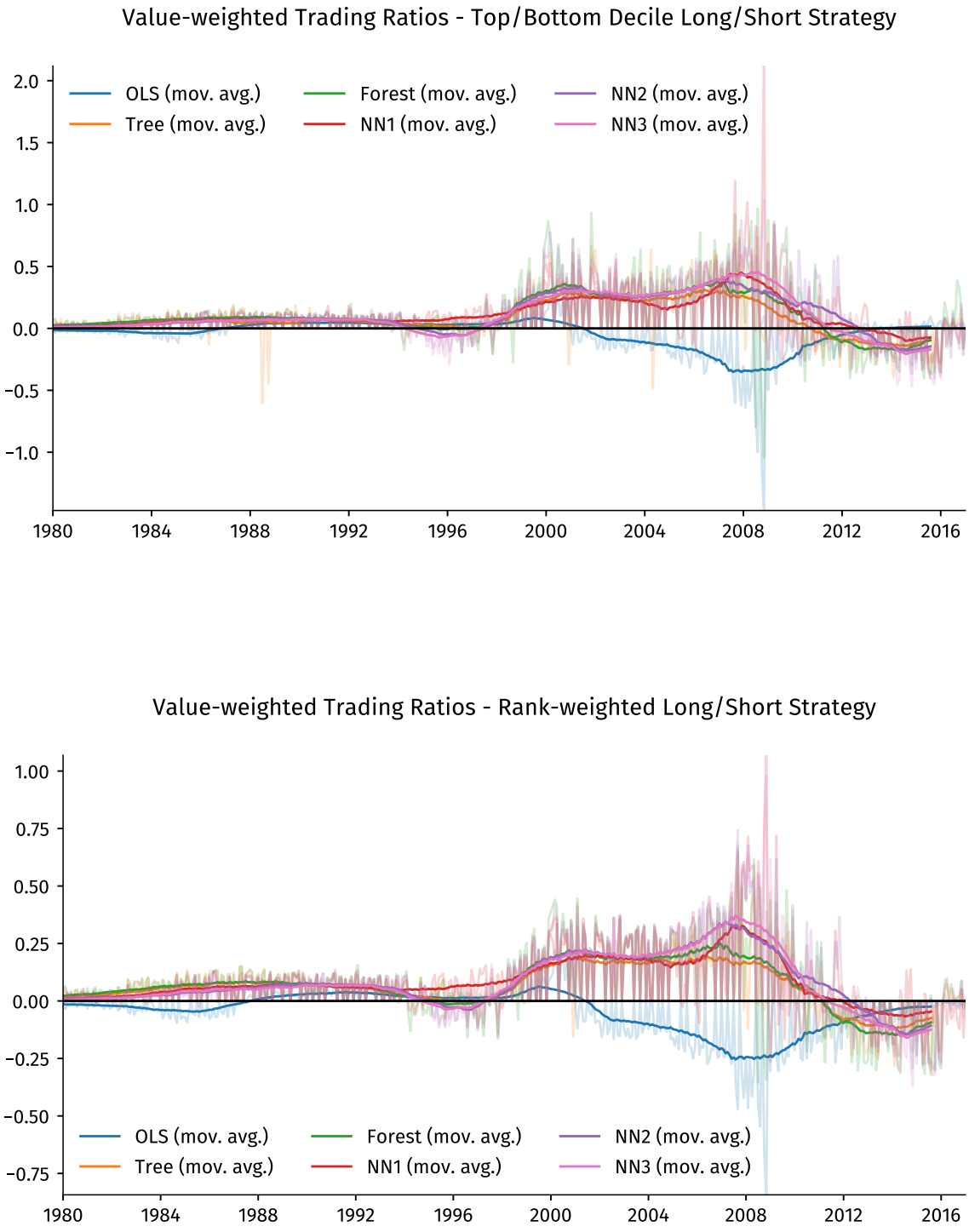
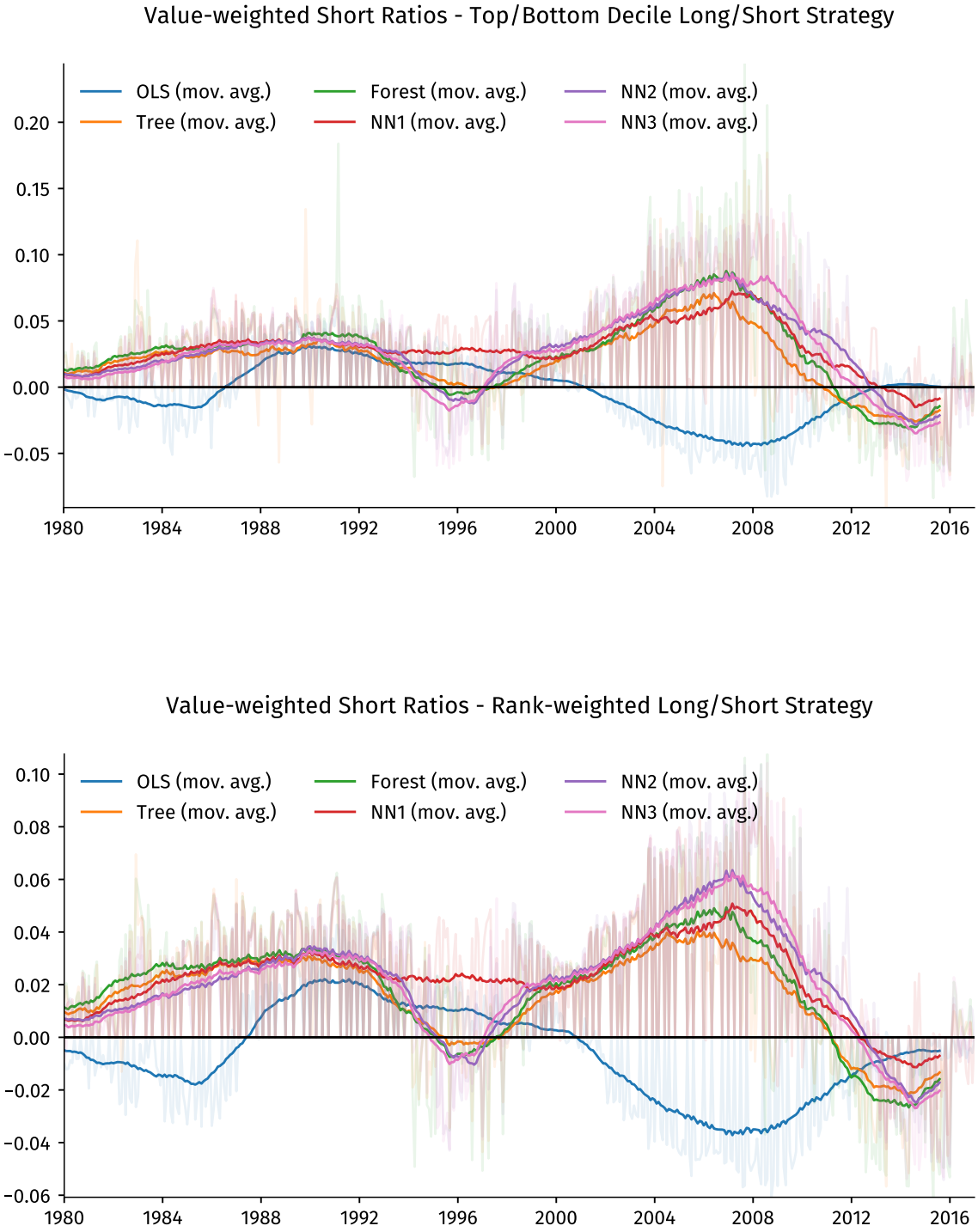


Figure A14: Uninterpolated Data : Weighted Short Ratio in Selected ML Portfolios



A.5. Short Interest along Machine Learning Predictions

Figure A15: Cross-Section Analysis of Short Ratios along Machine Learning Predictions

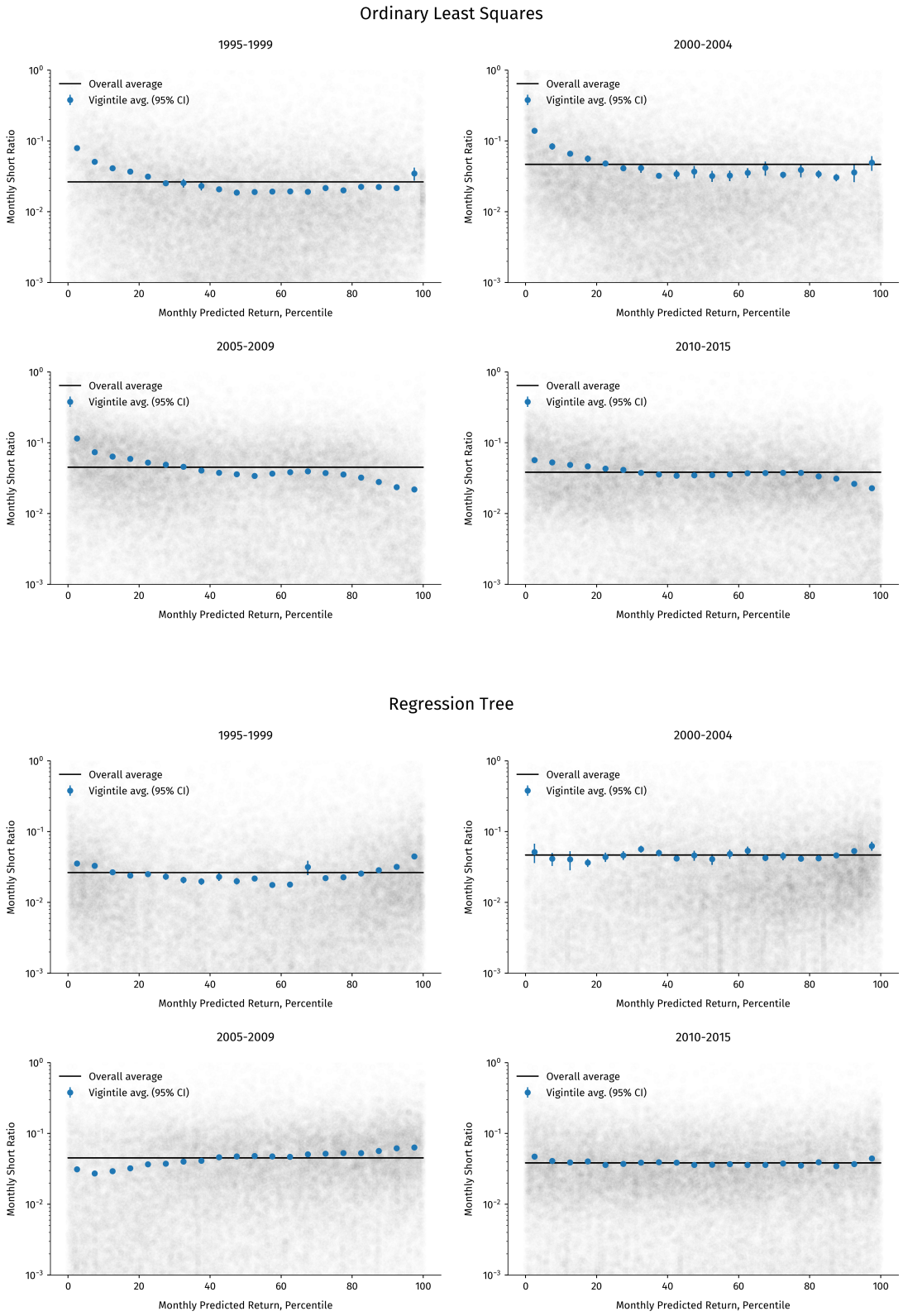


Figure A16: Cross-Section Analysis of Short Ratios along Machine Learning Predictions

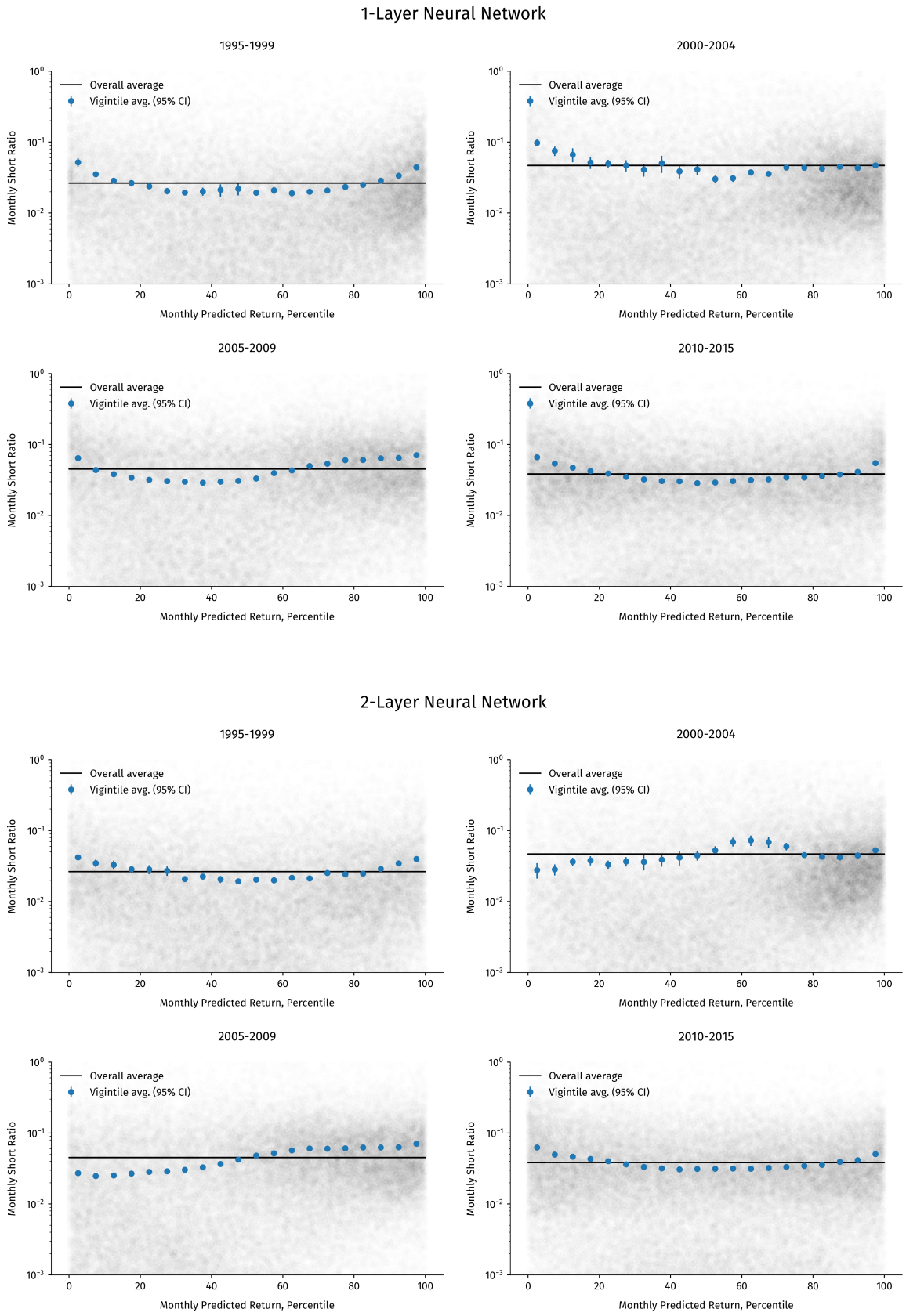


Figure A17: Time Series Analysis of Short Ratios around Machine Learning Predictions

Figure A18: Time Series Analysis of Short Ratios around Machine Learning Predictions

Figure A19: Cross-Section Analysis of Days to Cover along Machine Learning Predictions

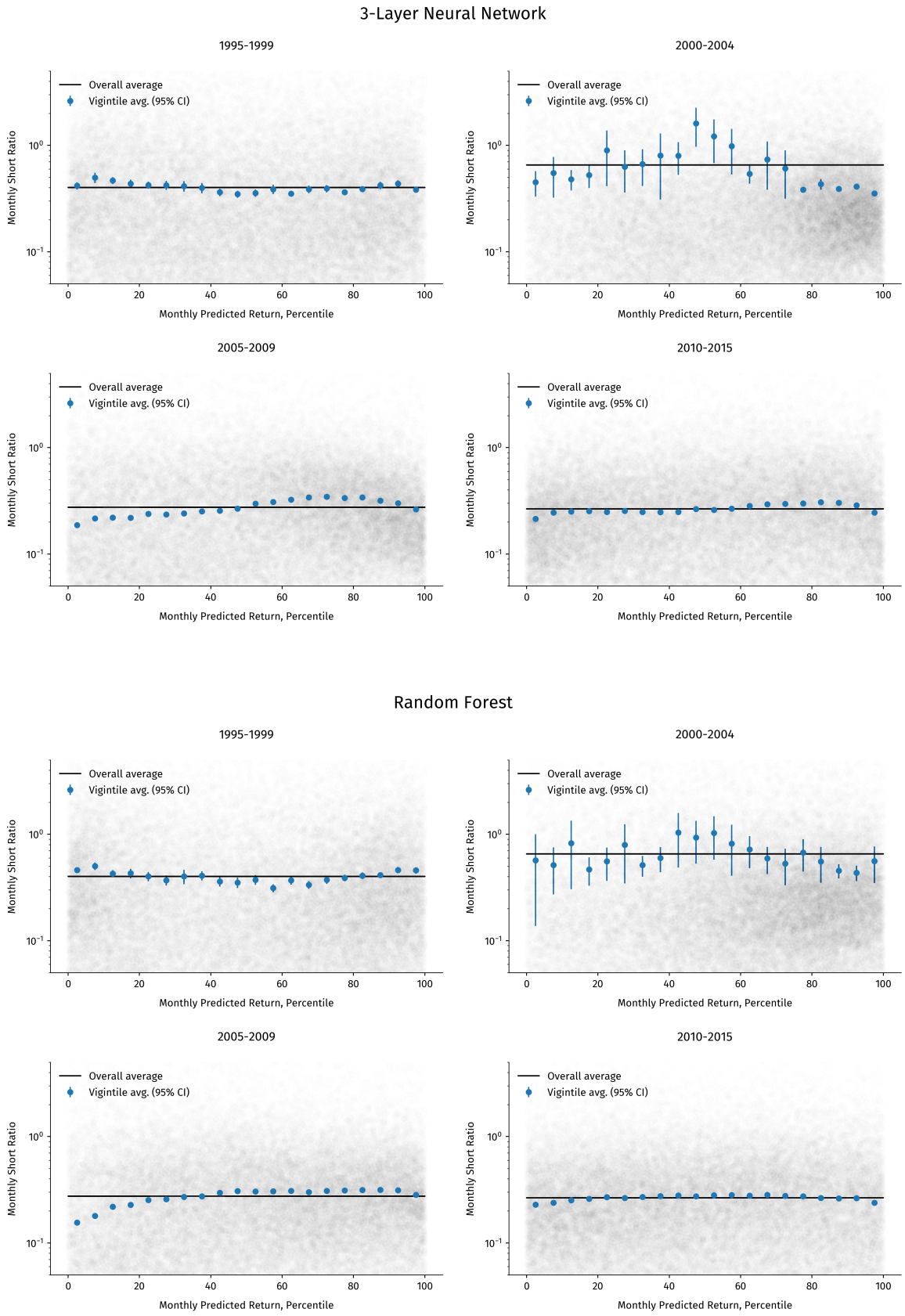


Figure A20: Time Series Analysis of Days to Cover around Machine Learning Predictions

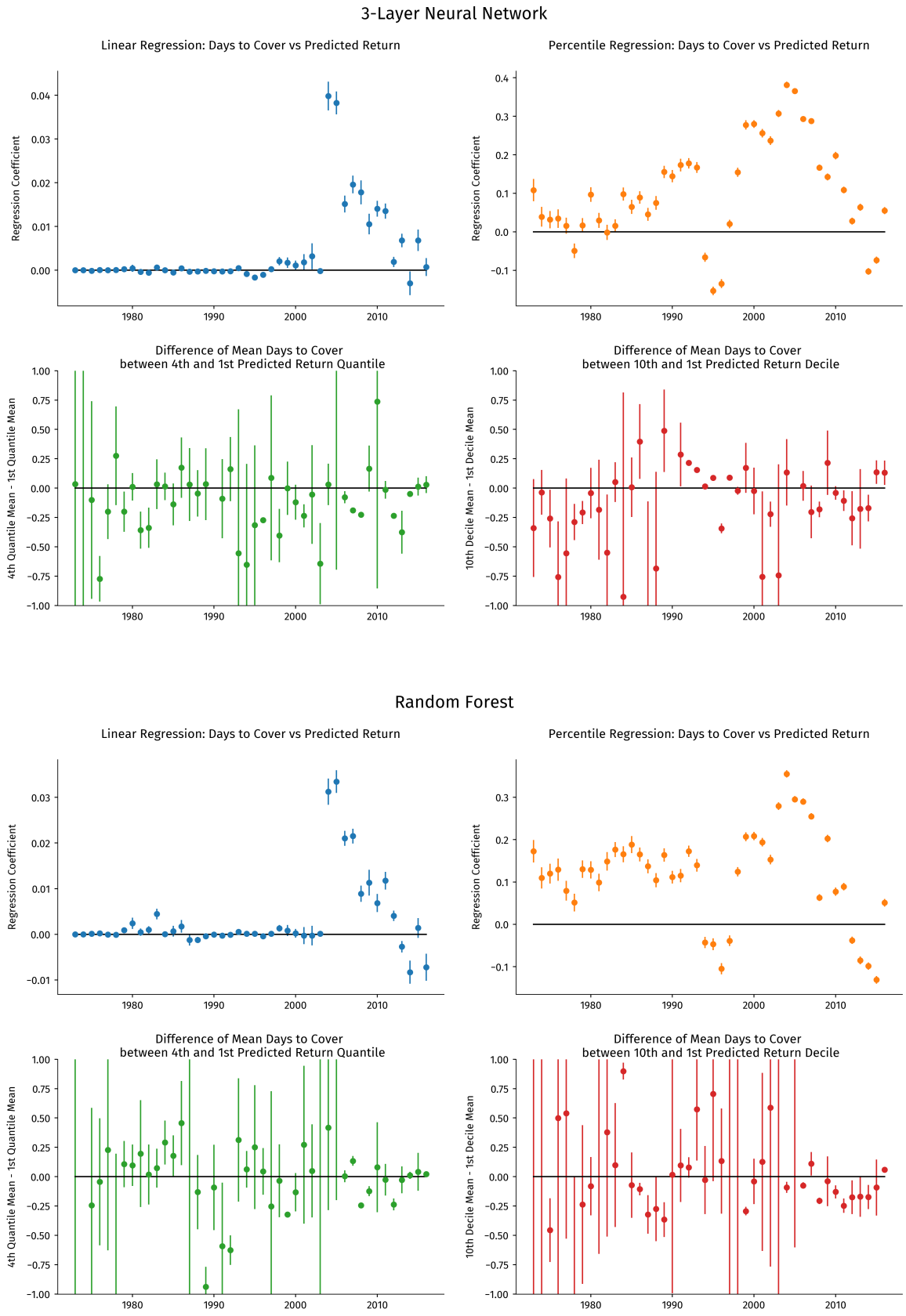
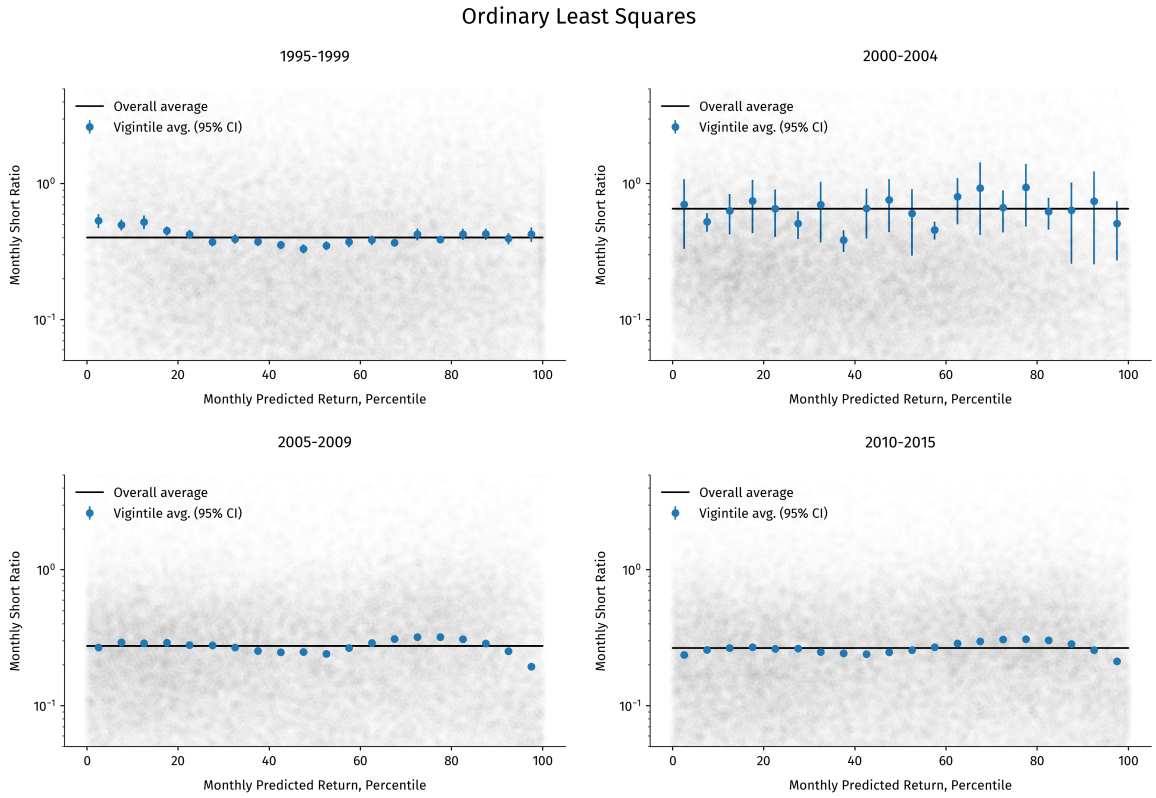


Figure A21: Cross-Section and Time Series Analysis of Days to Cover around OLS Predictions



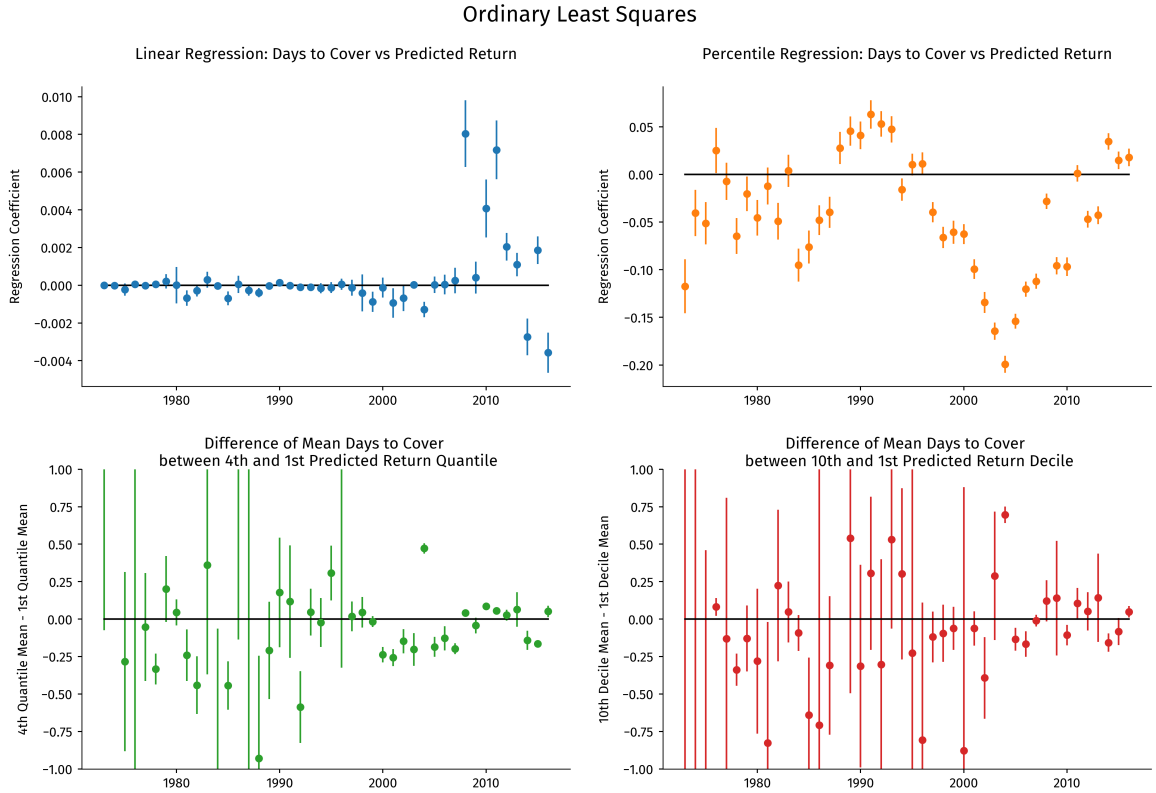
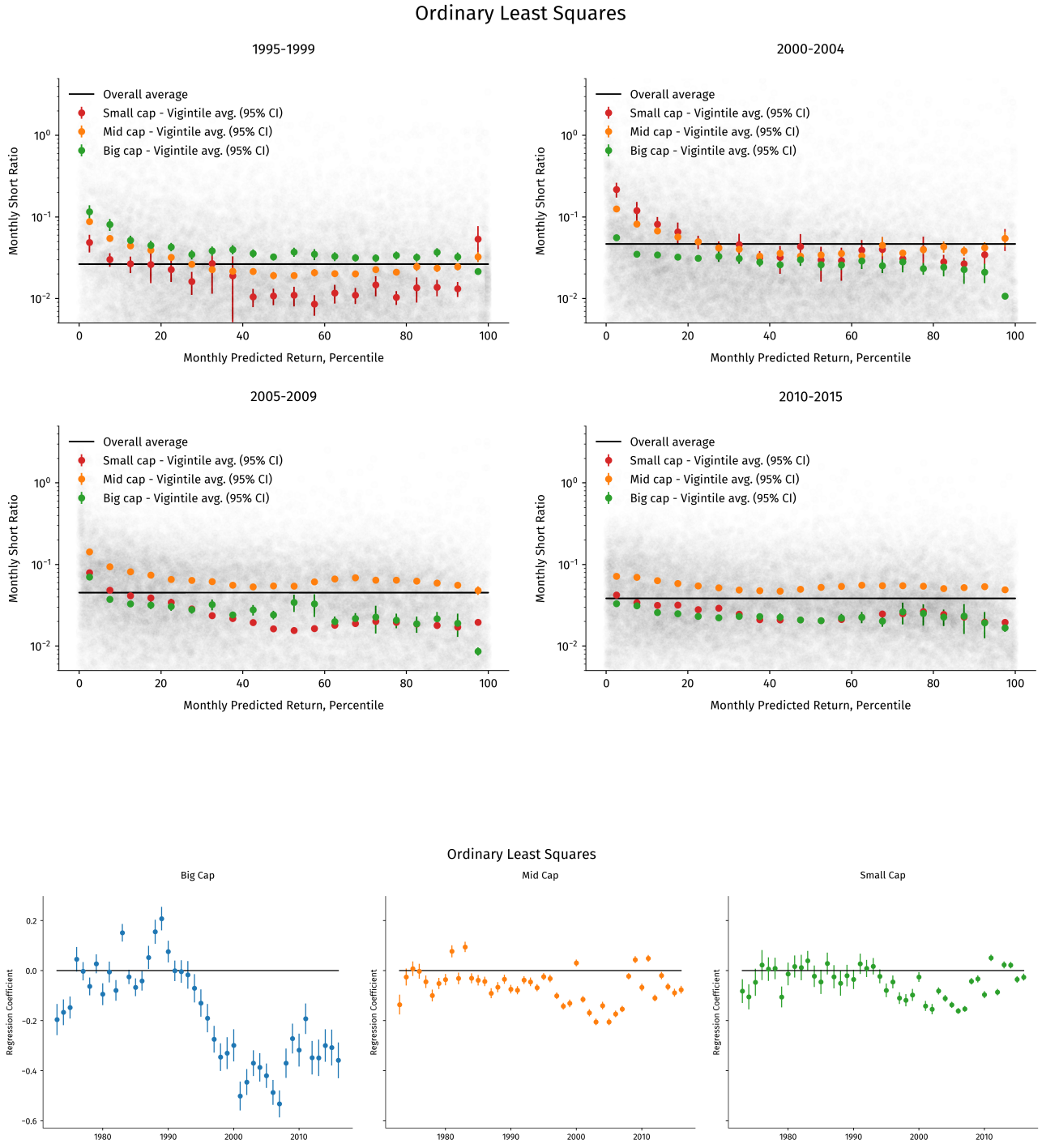


Figure A22: Short Interest around OLS Predictions by Market Capitalization



A.6. Regression Results on Post-Publication Decline

Table A19: Post-Publication Decline of Returns for Top/Bottom Decile Strategies

	Lasso - 3Y-FF3-Alpha			Enet - 3Y-FF3-Alpha		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Publication Dummy	-2.14**	-2.14***	-2.60*	-0.34	-1.09*	-0.38
	(0.92)	(0.64)	(1.36)	(0.57)	(0.57)	(0.83)
Obs	600	600	600	444	444	444
Adj. R^2	0.01	0.02	0.01	0.01	0.01	0.01

	Forest - 3Y-FF3-Alpha			GBRT - 3Y-FF3-Alpha		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Publication Dummy	4.04**	3.99*	6.84***	0.47	-1.57	-2.88*
	(1.75)	(2.07)	(1.59)	(1.85)	(1.43)	(1.69)
Obs	564	564	564	480	480	480
Adj. R^2	0.13	0.13	0.18	0.02	0.03	0.05

	NN1 - 3Y-FF3-Alpha			NN2 - 3Y-FF3-Alpha			NN3 - 3Y-FF3-Alpha		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Pub. Dummy	-0.16	-0.16	-2.18	2.65	2.65	-4.95*	2.65	2.65	-4.95*
	(1.41)	(0.84)	(1.48)	(2.85)	(2.46)	(2.79)	(2.85)	(2.46)	(2.79)
Obs	600	600	600	600	600	600	600	600	600
Adj. R^2	0.14	0.15	0.15	0.09	0.10	0.16	0.09	0.10	0.16

Regressions of monthly returns from different strategies on a post-publication dummy (equal to 1 after the publication year and 0 before) and controlling time trends. Time trends are linear, quadratic or quintic (fifth-order polynomial). We include returns up to 15 years before and after publication. Standard errors are computed using a Newey-West estimator allowing up to 10 years of lag. Parentheses are standard errors. * : p<.1, ** : p<.05, *** : p<.01.

Table A20: Post-Publication Decline of Returns for Rank-weighted Strategies

	Lasso - Returns			Enet - Returns		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Publication Dummy	3.66***	3.66***	1.01	-3.62***	-2.42**	-0.07
	(0.94)	(0.93)	(0.81)	(1.31)	(1.04)	(0.67)
Obs	600	600	600	444	444	444
Adj. R^2	0.04	0.04	0.06	0.06	0.06	0.08

	Forest - Returns			GBRT - Returns		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Publication Dummy	3.18	4.32***	3.57***	-2.73**	-0.57	-0.76
	(2.62)	(0.85)	(1.10)	(1.39)	(1.36)	(1.85)
Obs	564	564	564	480	480	480
Adj. R^2	0.01	0.09	0.10	0.01	0.03	0.06

	NN1 - Returns			NN2 - Returns			NN3 - Returns		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Pub. Dummy	3.77	3.77*	-2.38	2.29	2.29**	-1.17	2.29	2.29**	-1.17
	(2.37)	(2.06)	(1.50)	(1.53)	(1.15)	(1.36)	(1.53)	(1.15)	(1.36)
Obs	600	600	600	600	600	600	600	600	600
Adj. R^2	0.08	0.10	0.14	0.14	0.16	0.17	0.14	0.16	0.17

Regressions of monthly returns from different strategies on a post-publication dummy (equal to 1 after the publication year and 0 before) and controlling time trends. Time trends are linear, quadratic or quintic (fifth-order polynomial). We include returns up to 15 years before and after publication. Standard errors are computed using a Newey-West estimator allowing up to 10 years of lag. Parentheses are standard errors. * : p<.1, ** : p<.05, *** : p<.01.

Table A21: Post-Publication Decline of 3-Year FF3-Alphas for Top/Bottom Decile Strategies

	Lasso - 3Y-FF3-Alpha			Enet - 3Y-FF3-Alpha		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Publication Dummy	-2.14**	-2.14***	-2.60*	-0.34	-1.09*	-0.38
	(0.92)	(0.64)	(1.36)	(0.57)	(0.57)	(0.83)
Obs	600	600	600	444	444	444
Adj. R^2	0.01	0.02	0.01	0.01	0.01	0.01

	Forest - 3Y-FF3-Alpha			GBRT - 3Y-FF3-Alpha		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Publication Dummy	4.04**	3.99*	6.84***	0.47	-1.57	-2.88*
	(1.75)	(2.07)	(1.59)	(1.85)	(1.43)	(1.69)
Obs	564	564	564	480	480	480
Adj. R^2	0.13	0.13	0.18	0.02	0.03	0.05

	NN1 - 3Y-FF3-Alpha			NN2 - 3Y-FF3-Alpha			NN3 - 3Y-FF3-Alpha		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Pub. Dummy	-0.16	-0.16	-2.18	2.65	2.65	-4.95*	2.65	2.65	-4.95*
	(1.41)	(0.84)	(1.48)	(2.85)	(2.46)	(2.79)	(2.85)	(2.46)	(2.79)
Obs	600	600	600	600	600	600	600	600	600
Adj. R^2	0.14	0.15	0.15	0.09	0.10	0.16	0.09	0.10	0.16

Regressions of monthly returns from different strategies on a post-publication dummy (equal to 1 after the publication year and 0 before) and controlling time trends. Time trends are linear, quadratic or quintic (fifth-order polynomial). We include 3-year FF3-alphas up to 15 years before and after publication. Standard errors are computed using a Newey-West estimator allowing up to 10 years of lag. Parentheses are standard errors. * : p<.1, ** : p<.05, *** : p<.01.

Table A22: Post-Publication Decline of 3-Year FF3-Alphas for Rank-weighted Strategies

	Lasso - 3Y-FF3-Alpha			Enet - 3Y-FF3-Alpha		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Publication Dummy	1.76*	1.76**	-0.54	-3.90***	-3.07***	-2.69***
	(1.00)	(0.83)	(0.48)	(0.47)	(0.39)	(0.59)
Obs	600	600	600	444	444	444
Adj. R^2	0.34	0.41	0.74	0.80	0.84	0.86

	Forest - 3Y-FF3-Alpha			GBRT - 3Y-FF3-Alpha		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Publication Dummy	4.79**	5.55***	3.11***	0.69	2.67***	3.27***
	(1.91)	(0.98)	(1.17)	(0.83)	(0.57)	(1.13)
Obs	564	564	564	480	480	480
Adj. R^2	0.31	0.64	0.70	0.07	0.29	0.49

	NN1 - 3Y-FF3-Alpha			NN2 - 3Y-FF3-Alpha			NN3 - 3Y-FF3-Alpha		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Pub. Dummy	1.79	1.79	-2.29*	1.21	1.21	-2.22	1.21	1.21	-2.22
	(1.46)	(1.40)	(1.25)	(1.13)	(1.07)	(1.50)	(1.13)	(1.07)	(1.50)
Obs	600	600	600	600	600	600	600	600	600
Adj. R^2	0.52	0.57	0.77	0.68	0.70	0.76	0.68	0.70	0.76

Regressions of monthly returns from different strategies on a post-publication dummy (equal to 1 after the publication year and 0 before) and controlling time trends. Time trends are linear, quadratic or quintic (fifth-order polynomial). We include 3-year FF3-alphas up to 15 years before and after publication. Standard errors are computed using a Newey-West estimator allowing up to 10 years of lag. Parentheses are standard errors. * : p<.1, ** : p<.05, *** : p<.01.

B. Bibliography

References

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

Arnott, R., C. R. Harvey, and H. Markowitz (2019). A backtesting protocol in the era of machine learning. The Journal of Financial Data Science 1(1), 64–74.

Asquith, P., P. A. Pathak, and J. R. Ritter (2005). Short interest, institutional ownership, and stock returns. Journal of Financial Economics 78(2), 243–276.

Banz, R. W. (1981). The relationship between return and market value of common stocks. Journal of financial economics 9(1), 3–18.

Barr, J. R., E. A. Ellis, A. Kassab, C. L. Redfearn, N. N. Srinivasan, and K. B. Voris (2017). Home price index: a machine learning methodology. International Journal of Semantic Computing 11(01), 111–133.

Basu, S. (1977). Investment performance of common stocks in relation to their price-earnings ratios: A test of the efficient market hypothesis. The journal of Finance 32(3), 663–682.

Belson, W. A. (1959). Matching and prediction on the principle of biological classification. Journal of the Royal Statistical Society: Series C (Applied Statistics) 8(2), 65–75.

Bianchi, D., M. Büchner, and A. Tamoni (2019). Bond risk premia with machine learning. USC-INET Research Paper (19-11).

Brogaard, J. and A. Zareei (2018). Machine learning and the stock market. Available at SSRN 3233119.

Campbell, J. Y. and J. H. Cochrane (1999). By force of habit: A consumption-based explanation of aggregate stock market behavior. Journal of political Economy 107(2), 205–251.

Carhart, M. M. (1997). On persistence in mutual fund performance. The Journal of finance 52(1), 57–82.

Chen, L., M. Pelger, and J. Zhu (2019). Deep learning in asset pricing. Available at SSRN 3350138.

Chen, Y., Z. Da, and D. Huang (2019). Arbitrage trading: The long and the short of it. The Review of Financial Studies 32(4), 1608–1646.

Chordia, T., A. Subrahmanyam, and Q. Tong (2014). Have capital market anomalies attenuated in the recent era of high liquidity and trading activity? Journal of Accounting and Economics 58(1), 41–58.

Culkin, R. and S. R. Das (2017). Machine learning in finance: the case of deep learning for option pricing. Journal of Investment Management 15(4), 92–100.

Currie, J., H. Kleven, and E. Zwiers (2020). Technology and big data are changing economics: mining text to track methods. In AEA Papers and Proceedings, Volume 110, pp. 42–48.

Dimitriadou, A., P. Gogas, T. Papadimitriou, and V. Plakandaras (2019). Oil market efficiency under a machine learning perspective. Forecasting 1(1), 157–168.

Dimson, E. and P. Marsh (1999). Murphy's law and market anomalies. The Journal of Portfolio Management 25(2), 53–69.

Fama, E. F. (1991). Efficient capital markets: Ii. The journal of finance 46(5), 1575–1617.

Fama, E. F. and K. R. French (1993). Common risk factors in the returns on stocks and bonds. Journal of.

Fama, E. F. and K. R. French (2015). A five-factor asset pricing model. Journal of financial economics 116(1), 1–22.

Feng, G., S. Giglio, and D. Xiu (2020). Taming the factor zoo: A test of new factors. The Journal of Finance 75(3), 1327–1370.

Feng, G., N. G. Polson, and J. Xu (2018). Deep learning in asset pricing. arXiv preprint arXiv:1805.01104.

Fischer, T. and C. Krauss (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research 270(2), 654–669.

Friedman, J. H. (2002). Stochastic gradient boosting. Computational statistics & data analysis 38(4), 367–378.

Gentzkow, M., B. Kelly, and M. Taddy (2019). Text as data. Journal of Economic Literature 57(3), 535–74.

Green, J., J. R. Hand, and X. F. Zhang (2013). The supraview of return predictive signals. Review of Accounting Studies 18(3), 692–730.

Green, J., J. R. Hand, and X. F. Zhang (2017). The characteristics that provide independent information about average us monthly stock returns. The Review of Financial Studies 30(12), 4389–4436.

Gromb, D. and D. Vayanos (2010). Limits of arbitrage. Annual Review of Financial Economics 2(1), 251–275.

Gu, S., B. Kelly, and D. Xiu (2020). Empirical asset pricing via machine learning. The Review of Financial Studies 33(5), 2223–2273.

Hanson, S. G. and A. Sunderam (2014). The growth and limits of arbitrage: Evidence from short interest. The Review of Financial Studies 27(4), 1238–1286.

Harvey, C. R., Y. Liu, and H. Zhu (2016). ... and the cross-section of expected returns. The Review of Financial Studies 29(1), 5–68.

Ho, T. K. (1995). Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition, Volume 1, pp. 278–282. IEEE.

Hochreiter, S. and J. Schmidhuber (1997). Long short-term memory. Neural computation 9(8), 1735–1780.

Hoerl, A. E. (1962). Application of ridge analysis to regression problems.

Hou, K., C. Xue, and L. Zhang (2020). Replicating anomalies. The Review of Financial Studies 33(5), 2019–2133.

Hsu, M.-W., S. Lessmann, M.-C. Sung, T. Ma, and J. E. Johnson (2016). Bridging the divide in financial market forecasting: machine learners vs. financial economists. Expert Systems with Applications 61, 215–234.

Jacobs, H. (2015). What explains the dynamics of 100 anomalies? Journal of Banking & Finance 57, 65–85.

Jacobs, H. and S. Müller (2020). Anomalies across the globe: Once public, no longer existent? Journal of Financial Economics 135(1), 213–230.

Jensen, M. C. (1968). The performance of mutual funds in the period 1945–1964. The Journal of finance 23(2), 389–416.

Ke, Z. T., B. T. Kelly, and D. Xiu (2019). Predicting returns with text data. Technical report, National Bureau of Economic Research.

Krauss, C., X. A. Do, and N. Huck (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500. European Journal of Operational Research 259(2), 689–702.

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel (1989). Backpropagation applied to handwritten zip code recognition. Neural computation 1(4), 541–551.

Legendre, A. M. (1805). Nouvelles méthodes pour la détermination des orbites des comètes. F. Didot.

Loh, W.-Y. (2014). Fifty years of classification and regression trees. International Statistical Review 82(3), 329–348.

Lou, D. and C. Polk (2012). Comomentum: Inferring arbitrage activity from return correlations. In AFA 2013 San Diego Meetings Paper.

Lucas, R. E. (1978). Asset prices in an exchange economy. Econometrica: Journal of the Econometric Society, 1429–1445.

Malkiel, B. G. and E. F. Fama (1970). Efficient capital markets: A review of theory and empirical work. The journal of Finance 25(2), 383–417.

McClelland, J. L., D. E. Rumelhart, P. R. Group, et al. (1986). Parallel distributed processing. Explorations in the Microstructure of Cognition 2, 216–271.

McCulloch, W. S. and W. Pitts (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics 5(4), 115–133.

McLean, R. D. (2010). Idiosyncratic risk, long-term reversal, and momentum. Journal of Financial and Quantitative Analysis 45(4), 883–906.

McLean, R. D. and J. Pontiff (2016). Does academic research destroy stock return predictability? The Journal of Finance 71(1), 5–32.

Pearson, K. (1901). On lines of closes fit to system of points in space, london, e dinb. Dublin Philos. Mag. J. Sci 2, 559–572.

Pontiff, J. (2006). Costly arbitrage and the myth of idiosyncratic risk. Journal of Accounting and Economics 42(1-2), 35–52.

Rojas, C. G. and M. Herman (2018). Foreign exchange forecasting via machine learning.

Ross, S. (1976). The arbitrage theory of capital asset pricing. Journal of Economic Theory 13(3), 341–360.

Santosa, F. and W. W. Symes (1986). Linear inversion of band-limited reflection seismograms. SIAM Journal on Scientific and Statistical Computing 7(4), 1307–1330.

Stambaugh, R. F., J. Yu, and Y. Yuan (2015). Arbitrage asymmetry and the idiosyncratic volatility puzzle. The Journal of Finance 70(5), 1903–1948.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) 58(1), 267–288.

Tikhonov, A. N. (1943). On the stability of inverse problems. In Dokl. Akad. Nauk SSSR, Volume 39, pp. 195–198.

Timmermann, A. and C. W. Granger (2004). Efficient market hypothesis and forecasting. International Journal of forecasting 20(1), 15–27.

Weigand, A. (2019). Machine learning in empirical asset pricing. Financial Markets and Portfolio Management 33(1), 93–104.

Werbos, P. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences. Ph. D. Dissertation, Harvard University.

Wold, H. (1966). Estimation of principal components and related models by iterative least squares. Multivariate analysis, 391–420.

Wu, W., J. Chen, L. Xu, Q. He, and M. L. Tindall (2019). A statistical learning approach for stock selection in the chinese stock market. Financial Innovation 5(1), 20.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. Journal of the royal statistical society: series B (statistical methodology) 67(2), 301–320.