The Anachronistic Testing-Date Bias of Machine Learning Strategies*

Federico Tiberga
London Business School
ftiberga@london.edu

September 29, 2026

Abstract

We examine whether the documented outperformance of machine learning models in cross-sectional stock return prediction reflects genuine economic alpha or is partly an artefact of an anachronistic testing-date bias: the systematic backtesting of models over periods before their public availability, when investors could not yet deploy the underlying forecasting technology. We assign each of ten machine learning (ML) models a public availability date and measure the bias using value-weighted long-short portfolios, net of transaction costs, over an out-of-sample period from February 2005 to December 2022. The pattern is concentrated rather than uniform. The clearest case is the two-layer neural network, which earns 91% of its cumulative log net wealth before its architecture became public, and whose Fama–French six-factor alpha falls from 0.91% per month ($t = 2.80$) to zero afterwards. The other neural networks decline more mildly, while two boosted-tree strategies show no strong alpha before their availability dates. Pooled estimates and continuous adoption indices built from research output, open-source code, and finance-sector hiring suggest a similar negative effect but are not statistically significant after accounting for cross-model dependence. The strongest evidence that public availability marks a shift in market behaviour comes from short selling: once a neural network model is made public, its short signal aligns more closely with subsequent movements in short interest, a pattern not observed for boosted-tree models. As a result, full-sample ML backtests can conflate implementable performance with profits earned before the relevant technology was available, especially for successful neural-network strategies.

Keywords: Machine Learning, Cross-Sectional Asset Pricing, Return Predictability, Transaction Costs, Publication Decay, Technology Diffusion, Market Efficiency

*Submitted to the AFA PhD Student Poster Session, 2027 Annual Meeting.

1 Introduction

Following the seminal work of Gu, Kelly, and Xiu (2020), a large body of research has analysed the ability of machine learning to predict cross-sectional stock returns (Gu, Kelly, and Xiu, 2021; Chen, Pelger, and Zhu, 2024; Azevedo, Hoegner, and Velikov, 2023). To evaluate performance, this literature commonly relies on backtests that assume full access to data, features, and architectures throughout the sample period. In doing so, it implicitly assesses an algorithm during years that predate its invention, its large-scale feasibility, or its public availability to investors. Martin and Nagel (2022) show that predictability should instead be judged relative to the information and forecasting technology available to investors in real time, since empirical tests would otherwise grant the researcher an ex-post advantage over historical investors. Building on this insight, we define and measure an anachronistic testing-date bias, which arises when the information set used to train and test a model embeds predictors or model architectures that were not yet economically or technologically feasible. Our focus is the forecasting technology rather than the signals it processes: the predictors in our data enter only after their publication, yet the model applied to them may still come from the future. Can one meaningfully evaluate how a modern model would have performed in a market that did not yet have access to it?

Existing work already recognises that the signal set is time-varying, since published predictors lose much of their power after disclosure (McLean and Pontiff, 2016; Azevedo et al., 2023). Even studies that account for the timing of signal availability, however, typically assume that the forecasting technology itself remains unchanged and universally accessible throughout the sample. This overlooks the reality that modern machine learning rests on advances in algorithms, software, and computing that emerged at identifiable points in time, so that backtesting a model before those advances attributes to investors a capability they did not have at the time. Closest to our question, Martin and Nagel (2022) formalise why an econometrician with hindsight can detect in-sample predictability that no contemporaneous investor could have exploited, but they do not study the availability of the predictive algorithms themselves: the fact that a neural network backtested in 2005 could not have been deployed, in its modern form, by any market participant at that date. We ask how much of the performance reported in machine-learning backtests accrues in periods before

the method was publicly usable, and whether that performance deteriorates once the method diffuses among investors.

We study this question for ten models estimated on a broad cross-section of US common stocks over the out-of-sample period from February 2005 to December 2022: three regularised linear models (Ridge, Lasso, and Elastic Net), three gradient-boosted tree ensembles (XGBoost, LightGBM, and CatBoost), and four feedforward neural networks of increasing depth. The feature set follows Azevedo et al. (2023) and combines the anomaly signals of Chen and Zimmermann (2022) with industry classifiers and the macroeconomic predictors of Welch and Goyal (2008), each admitted only after its publication date. We assign each model a public-availability date. For the seven nonlinear models, this date is data-driven: the earliest of the model's first machine-learning asset-pricing preprint and its first public implementation. For the neural networks, the rule applies to the last essential component of the tested architecture to become public: batch normalisation in February 2015 (Ioffe and Szegedy, 2015). The three linear models predate the sample, so they are always available to investors and serve as a control group, while the seven nonlinear models become available only partway through the out-of-sample period, which splits each one's record into a pre- and a post-availability span. We form value-weighted long-short decile portfolios and measure performance both gross and net of the high-frequency effective bid-ask spread costs of Chen and Velikov (2023), holding the stock universe, signal timing, validation design, portfolio rule, and cost model fixed across models, so that differences reflect the forecasts rather than the trading design. Throughout, a model's alpha is the abnormal return of its net long-short portfolio relative to the Fama and French (2018) six-factor model, which augments the Fama–French five-factor model with momentum. We complement the transparent binary split at each availability date with continuous monthly indices, built from research output, public code, and finance-sector hiring, that measure gradual diffusion, and with short-interest data that provide market-based evidence from investors' positions rather than a further transformation of the return series.

All ten portfolios earn positive gross returns. Net of costs, the four neural networks and the three linear models remain significant, whereas among the boosted trees, only Cat-

Boost is significant, and only at the 10% level. The clearest case of the bias is the two-layer network FFNN2, which earns 91% of its cumulative log net wealth before its architecture became public, against a 56% pre-availability share of the sample's months, and whose alpha falls from 0.91% per month ($t = 2.80$) to zero afterwards. FFNN4 and FFNN5 decline more mildly, and FFNN3 is flat. The boosted trees show no common pattern: XGBoost declines but not significantly, while LightGBM and CatBoost, which had little pre-availability net alpha for adoption to erode, improve after their dates. Pooled across the seven nonlinear models, the pre-availability wealth share (62%) exceeds the time share (58%), to be understood as the mean percentage of months in the test sample that occur before the public-availability dates, by only four points, and once we resample whole calendar months to respect the correlation among the four networks that share the February 2015 date, the pooled decline keeps its negative sign but is no longer statistically significant, since those four networks act almost as a single observation and leave the pooled test with little power. We therefore read the pooled coefficient as descriptive and place more weight on the model-level FFNN2 result, its comparison with the always-available Ridge portfolio, and the short-interest evidence, which is the strongest in the paper. After each network becomes public, the stocks it ranks as unattractive experience larger subsequent increases in short interest, significantly so for all four networks. On the other hand, the less profitable tree models do not experience such significant increases. Moreover, an event study shows short interest rising when a stock enters several model short legs simultaneously, with no preceding buildup.

These results have practical implications for how machine-learning backtests are read. A full-sample backtest averages over periods when no market participant could have implemented the strategy, and it implicitly assumes that the abnormal performance earned in those periods would persist once the method becomes public; our evidence suggests the opposite for the strategies that had genuine net alpha to lose. This pattern is consistent with the efficient-markets hypothesis operating with a delay, and it extends the publication-induced decay that McLean and Pontiff (2016) document for individual signals to the predictive technology itself. The claim is deliberately narrow. We do not argue that every machine-learning strategy contains an anachronistic component, as the tree results make clear, nor that short sellers run the exact architectures we study. We show that public-availability dates belong

in a real-time backtest alongside vintage data, signal-publication dates, and trading costs, so that a full-sample record and a technology-feasible record, which answer different questions, are both reported.

Our paper connects three strands of literature. The first studies machine learning for cross-sectional return prediction, where neural networks and boosted trees improve on linear benchmarks (Gu et al., 2020), with extensions to the predictor space, factor structure, and model complexity (Kelly, Pruitt, and Su, 2019; Kozak, Nagel, and Santosh, 2020; Freyberger, Neuhierl, and Weber, 2020; Gu et al., 2021; Chen et al., 2024; Kelly, Malamud, and Zhou, 2024), and a parallel literature showing that trading frictions absorb much of the profitability of complex, high-turnover strategies and that their performance weakens in more recent, more liquid samples (Novy-Marx and Velikov, 2016; DeMiguel, Martin-Utrera, Nogales, and Uppal, 2020; Chen and Velikov, 2023; Avramov, Cheng, and Metzker, 2023; Azevedo et al., 2023; Detzel, Novy-Marx, and Velikov, 2023; Li, DeMiguel, and Martin-Utrera, 2024). We add a timing condition to this evidence: a model's historical performance informs an investor's attainable return only once the model itself is available. The second strand studies publication decay and real-time information sets. Portfolio returns on published predictors fall out-of-sample and again after publication as arbitrage capital flows in (McLean and Pontiff, 2016), and a large literature debates how much of the anomaly record survives replication and multiple-testing corrections (Harvey, Liu, and Zhu, 2016; Green, Hand, and Zhang, 2017; Hou, Xue, and Zhang, 2020; Harvey and Liu, 2021; Jensen, Kelly, and Pedersen, 2023), with the open-source library of Chen and Zimmermann (2022) supplying our anomaly signals. We extend the real-time restriction of Martin and Nagel (2022) from the signal to the forecasting method, because disclosing a signal and making a technology usable are separate events. The third strand studies technology diffusion, which spreads gradually and at rates that differ across users and applications (Griliches, 1957; Comin and Hobijn, 2010). We follow its measurement tradition, accumulating research output into a knowledge stock (Hall, Jaffe, and Trajtenberg, 2001; Bloom, Jones, Van Reenen, and Webb, 2020), adapt the online-vacancy approach of Acemoglu, Autor, Hazell, and Restrepo (2022) to finance, and add public code as a direct measure of implementability. To our knowledge, we are the first to apply these diffusion measures to the predictive technologies of asset pricing and link them to strategy

returns.

The remainder of the paper is organised as follows. Section 2 presents the economic intuition and its empirical predictions. Sections 3 and 4 describe the out-of-sample design, the public-availability dates, and the adoption measures. Section 5 reports the return evidence, and Section 6 the short-interest evidence. The final section discusses interpretation and limits before concluding. The appendix collects the derivations, construction records, and supporting tests.

2 Technology Diffusion and Alpha Decay

The economic mechanism is simple. A predictive technology produces a trading signal, investors learn to use it, and their trades press prices against the signal. Implementation costs and limited arbitrage capital prevent complete elimination of the return. We formalise this idea by adapting the diffusion model of Azevedo et al. (2023). The full derivation appears in Appendix A.

Let α_m^0 be the alpha available from technology m before investors can adopt it, c_m its per-period implementation cost, and $a_{m,t}$ the share of potential arbitrageurs using it at date t. Adoption is zero before the public-availability date τ_m and rises gradually afterwards. Price impact and risk-bearing capacity are collected in $\kappa_m > 0$. Equilibrium alpha is

$$\alpha_{m,t} = c_m + \frac{\alpha_m^0 - c_m}{1 + \kappa_m a_{m,t}}. \tag{1}$$

If $\alpha_m^0 > c_m$, adoption reduces alpha. A larger initial net alpha leaves more return for arbitrageurs to erode. If the strategy never clears its implementation cost, adoption need not lower its return. The positive lower bound in Equation (1) reflects the limits to arbitrage in Shleifer and Vishny (1997) and Gromb and Vayanos (2010), as well as the costly information production in Grossman and Stiglitz (1980). A conventional full-sample backtest averages the pre-availability return with the lower return observed as adoption spreads. An investor can earn only the second component. The resulting gap is larger when the availability date

occurs later in the sample and when the technology has more initial net alpha. Appendix A.2 derives this decomposition and the comparative statics.

The framework yields three predictions. First, returns should not change at an arbitrary date for a method already available throughout the sample. The linear models and synthetic release dates provide these placebo tests. Second, factor-adjusted returns should fall as adoption rises, with the largest decline among models that had the most pre-availability net alpha. Binary availability dates provide a direct split, whereas continuous indices measure the slower diffusion process. Third, market positions should become more closely aligned with a model's signals once the technology is made public. Because Equation (1) treats adoption $a_{m,t}$ as latent, we do not derive this alignment from the equation but use it as a behavioural proxy for adoption, which we measure through subsequent changes in short interest.

3 Data, Models, and Portfolios

The stock universe contains US common shares listed on NYSE, AMEX, and Nasdaq. The underlying panel begins in March 1957 and contains about 3.5 million stock-month observations. Models predict month $t+1$ total returns, including delisting returns, with information available by the end of month t. The portfolio evaluation runs from February 2005 to December 2022. Appendix B describes the CRSP filters, delisting-return treatment, and monthly timing conventions.

We align our analysis with Azevedo et al. (2023) regarding features and ML algorithm training. Predictors come from the 325 anomalies compiled by Chen and Zimmermann (2022), the Fama–French 49 industry classification of Fama and French (1997), which enters as 48 dummy variables, and eight macroeconomic variables from Welch and Goyal (2008). An anomaly is added to the feature set only after its publication date. The set therefore expands from 192 predictors in 2005 to 381 firm, industry, and macroeconomic features by the end of the sample. This rule removes signal look-ahead from the portfolio tests and leaves model availability as the timing question of interest. Within each month, continuous

firm characteristics are mapped to percentile ranks between -1 and 1, and missing values are set to the cross-sectional median after ranking. Appendix B reports feature screening and the checks on missing data.

We estimate three model families. Ridge, Lasso, and Elastic Net form the regularised linear group. XGBoost, LightGBM, and CatBoost form the boosted-tree group. Four feed-forward neural networks, labelled FFNN2 through FFNN5, contain two to five hidden layers. Their hidden widths decline geometrically with depth. Each hidden layer uses batch normalisation followed by a leaky-ReLU activation. Training uses mean squared error, the Adam optimiser (Kingma and Ba, 2014), a small L1 penalty, and early stopping. We average the predictions of five independently initialised networks. The linear methods were public by the start of the portfolio sample. The trees and neural networks receive the dates described in Section 4.

The estimation design is annual and strictly forward-looking. To forecast returns in year y, the training sample expands from March 1957 through $y-7$, and the next six years, $y-6$ through $y-1$, form the validation sample. Hyperparameters are selected separately for each prediction year: the linear models use a validation grid search, and each boosted tree uses 40 trials of a tree-structured Parzen search (Akiba, Sano, Yanase, Ohta, and Koyama, 2019) with early stopping on the same validation set. We use the Parzen search to reduce computation, not to improve model selection: a grid search over the same space, available throughout the sample, would yield similar hyperparameters at a higher computational cost. Because the tuning method changes run time but not the selected model, it introduces no testing-date bias. The neural-network architecture and training constants are fixed across years, while validation loss determines the stopping point. Once a model is fitted, it produces monthly forecasts throughout year y, so that no observation from the prediction year enters estimation, validation, or feature selection. This design is common to all ten models, and Appendix B records the larger parameter grids and implementation details that affect computation but not the information set available for any prediction.

Each month, stocks are sorted into deciles using predicted next-month returns and NYSE breakpoints. The value-weighted long-short portfolio buys the top decile and shorts the bottom decile, then rebalances at the next monthly formation date. NYSE breakpoints prevent the large number of small Nasdaq firms from determining the cutoffs, while value weighting limits the influence of microcaps. Gross returns measure the value of forecasts before implementation costs are deducted. Net returns are calculated by subtracting trading costs derived from the high-frequency effective bid-ask spread estimates of Chen and Velikov (2023). For each leg, cost equals the effective spread times the absolute change from the drifted position to the new target position. We charge this cost on both the long and the short side. The same universe, real-time signal set, portfolio rule, and cost model are used for every technology, so differences across models come from their forecasts rather than from a different trading design.

4 Measuring Model Availability and Adoption

Public availability is not the invention date of a statistical idea. It is the first month in which an investor can identify and implement the method used in the backtest. For each nonlinear model, we take the earlier of two observable events: the first machine-learning asset-pricing preprint using the method and the first public implementation. The rule is deliberately based on public records. Private use may occur earlier, but it cannot be dated consistently across models. The linear methods predate the portfolio sample and remain in the analysis as an always-available comparison group. For the three boosted trees, working code appeared before the canonical papers, so we date XGBoost to February 2014, LightGBM to August 2016, and CatBoost to July 2017, each the month of the first public implementation and each preceding its canonical publication by one to two years. We use the code dates because an investor could have inspected and run the method at that point. Later paper dates would assign extra months to the pre-availability period and mechanically enlarge the anachronistic component, so the baseline dates are demanding rather than favourable to our hypothesis.

The neural-network date needs a narrower interpretation. Neural networks and return-prediction applications existed much earlier (White, 1988). We date the architec-

ture tested here, not the broad idea of neural-network investing. The backtest uses leaky-ReLU activations, batch normalisation, and Adam. Adam was published in December 2014 (Kingma and Ba, 2014). Batch normalisation, the last required component, appeared in February 2015 (Ioffe and Szegedy, 2015). We therefore assign February 2015 to FFNN2 through FFNN5. The most aggressive earlier alternative is January 2010, the date of the first generic neural-network repository in our code sample. Applying that date to all four networks reduces the average negative shift but does not reverse it. The later date is still conceptually preferable because a generic repository is not the architecture used in the test. The alternative establishes that the central FFNN2 result does not depend on treating 2010 through 2014 as technologically impossible. Appendix C reports the four network estimates under both dates and documents the source records.

4.1 Continuous Adoption Measures

A public date marks feasibility, but diffusion is gradual. We therefore construct monthly indices from research papers, public code, and finance-sector vacancies. For model family c, source k, and month t, the index is

$$A_{c,t}^{k} = \mathrm{z}\left(\log\left[1 + \sum_{s \le t} n_{c,s}^{k}\right]\right), \tag{2}$$

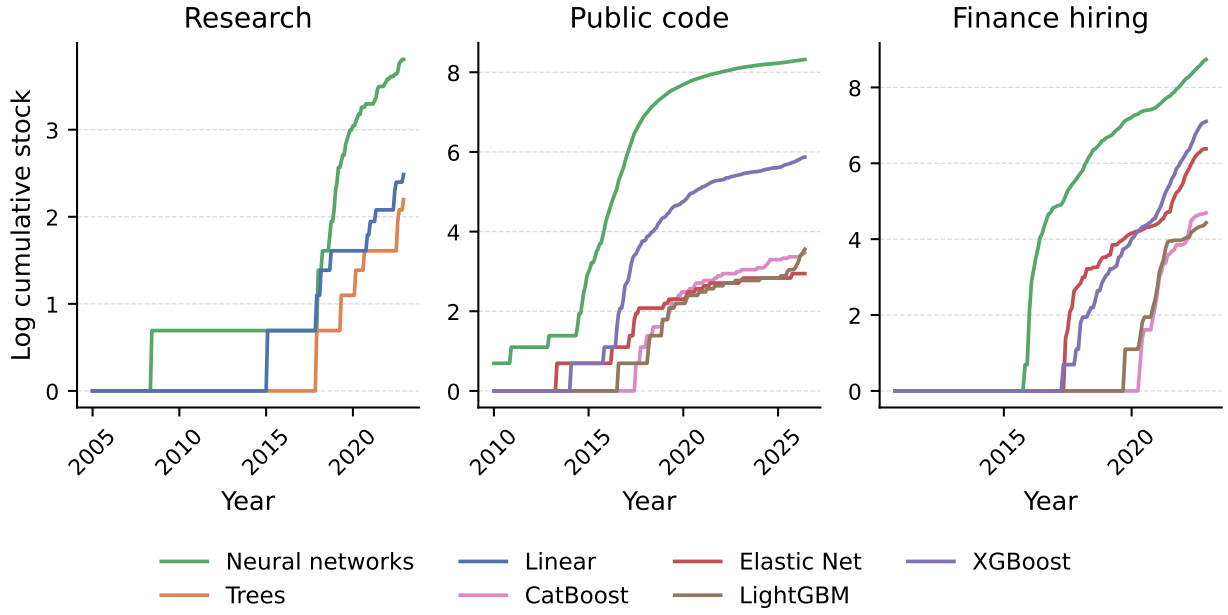
where $n_{c,s}^{k}$ counts the number of new papers, repositories, or relevant job postings. The cumulative stock records knowledge available by month t. The logarithm limits the influence of rapid growth late in the sample, and the standardisation makes coefficients comparable across sources.

The research index counts machine-learning asset-pricing papers classified by method. The code index counts public repositories linked to a library or algorithm. The hiring index counts finance vacancies that request a model-specific skill in a quantitative, portfolio, alpha, systematic, or trading role. These sources capture different margins. Papers measure disclosed knowledge, repositories measure practical access, and vacancies measure demand for the skill inside finance. Their usable windows also differ. Research covers the full portfolio

period, code begins in 2010, and hiring begins in 2016. Appendix C describes the Semantic Scholar, GitHub, and Revelio samples, including classification and duplicate handling. Figure 1 shows the three cumulative series before standardisation. Research and coding begin earlier, while the hiring series becomes informative only later in the sample. No single source determines the continuous-treatment result.

Figure 1: The Evolution of Model Adoption

The figure plots log cumulative research papers, public repositories, and finance-sector vacancies by model family. Regression indices are standardised within each source's usable window. Appendix C gives the source definitions and coverage.



5 Return Evidence

Table 1 reports all ten portfolios. Gross mean returns are positive throughout the model set. FFNN5 earns the largest gross return at 1.30% per month. Ridge and Elastic Net have the two highest gross Sharpe ratios, indicating that much of the return-prediction signal is captured by simpler methods. Trading costs lower mean returns by roughly 18 to 28 basis points per month. The four neural-network portfolios remain profitable after costs, with net returns from 0.78% to 1.05%. The three linear portfolios are also significant. Ridge earns

Table 1: Long-Short Portfolio Returns

This table reports mean monthly gross and net returns for each machine-learning long-short strategy over February 2005–December 2022. Portfolios are value-weighted, long the top predicted-return decile and short the bottom decile, using NYSE breakpoints on US common equities listed on NYSE, AMEX, and Nasdaq. Net returns subtract the high-frequency effective bid-ask spread cost estimates of Chen and Velikov (2023). SR denotes the annualised Sharpe ratio, computed as $\sqrt{12}$ times the ratio of the mean to the standard deviation of monthly returns, and TO% is the mean monthly two-way average-leg turnover. Stars are based on Newey–West HAC standard errors: $^{*}p < 0.10$, $^{**}p < 0.05$, and $^{***}p < 0.01$.

	Gross		Net		
Model	Ret%	SR	Ret%	SR	TO%
FFNN5	1.301***	0.89	1.049***	0.65	130.4
FFNN3	1.222***	0.83	0.953**	0.61	132.5
FFNN4	1.195***	0.78	0.943**	0.55	129.2
Ridge	1.062***	0.99	0.878***	0.78	139.0
CatBoost	1.057**	0.69	0.782*	0.44	132.9
FFNN2	1.028***	0.76	0.782**	0.51	129.7
Elastic Net	1.005***	0.95	0.813***	0.77	140.8
XGBoost	0.998*	0.47	0.755	0.32	119.8
Lasso	0.869***	0.73	0.602**	0.57	158.7
LightGBM	0.778	0.38	0.541	0.23	99.9

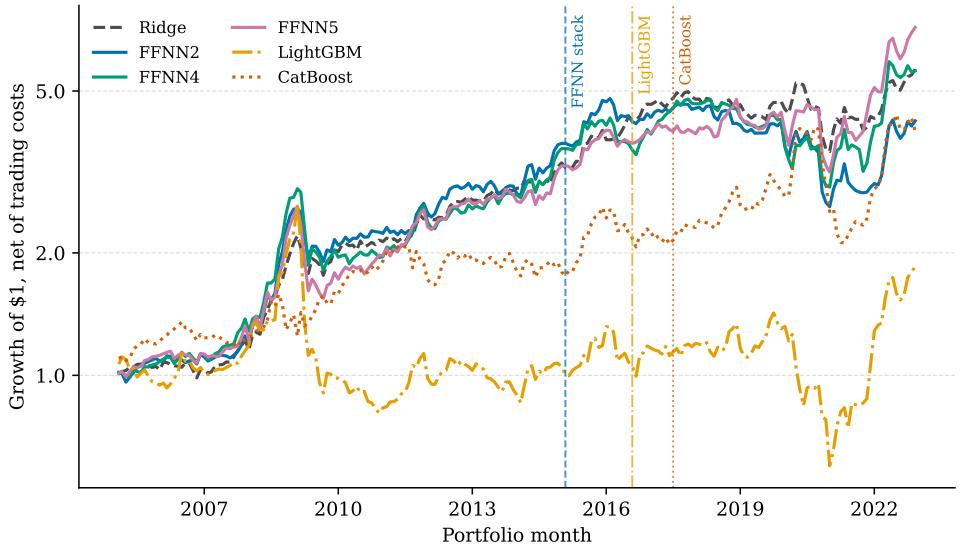
0.88% per month and has the highest net Sharpe ratio. XGBoost and LightGBM are not statistically different from zero after costs, while CatBoost is significant at the 10% level. These differences matter for the diffusion mechanism because a model without pre-availability net alpha has little for later adopters to erode.

Figure 2 plots the cumulative net wealth for the three networks whose pre-availability alpha t-statistics exceed 2, the two late-dated tree models, and Ridge. FFNN2 accumulates most of its final wealth before February 2015 and changes little afterwards. FFNN4 and FFNN5 continue to grow, but at lower rates. LightGBM and CatBoost do not display a break towards weaker cumulative performance at their public dates. Ridge supplies a stable calendar-time benchmark.

We begin with the average relation between adoption and returns. We then show full-sample performance and split each model at its public-availability date. The final comparison subtracts the Ridge return from each nonlinear model's return. This sequence separates the

Figure 2: Cumulative Net Wealth for Selected Long-Short Portfolios

The figure plots the growth of one dollar invested in selected value-weighted long-short portfolios from February 2005 to December 2022, net of the effective-spread trading costs of Chen and Velikov (2023). Vertical lines mark the February 2015 feedforward-network date, the August 2016 LightGBM date, and the July 2017 CatBoost date. Ridge was available before the sample. The vertical axis is logarithmic.



pooled summary from the model-level evidence that drives the paper's interpretation.

5.1 Empirical Design

For model m, adoption enters the factor regression directly:

$$r^{LS}_{m,t+1} = \alpha_m + \lambda A_{m,t} + \beta_m^{\top} F_{t+1} + \varepsilon_{m,t+1}, \tag{3}$$

where $r^{LS}_{m,t+1}$ is the net long-short return and F_{t+1} contains the Fama and French (2018) six factors, the Fama–French five factors plus momentum. In pooled regressions, α_m is a model fixed effect, and factor loadings are model-specific. The binary version sets $A_{m,t}$ equal to one after model m becomes public. The continuous version uses one of the three indices in Section 4. The diffusion hypothesis predicts $\lambda < 0$. Raw-return specifications omit factor controls. The timing preserves the investor's information set. Availability and adoption are measured by the portfolio-formation month, while the dependent variable is the return earned over the next month. Every model contributes the same monthly portfolio series over its usable source window. We report Newey–West standard errors with automatic lag

selection. The pooled regression constrains λ to be common across models, so it gains observations at the cost of averaging technologies with different pre-availability profits. For each model, a separate break regression allows the intercept and every factor loading to change at availability. The reported pre- and post-availability alphas are therefore not forced to share risk exposures. A March 2015 synthetic date assigned to Elastic Net provides a direct placebo. Appendix D.6 repeats that test over dates from 2010 to 2018, and Appendix D.5 reports the complete loading estimates.

5.2 Pooled Adoption Estimates

Table 2 reports the pooled estimates. The binary post-availability coefficient is -0.03% per month in raw returns and -0.42% per month after the Fama–French six-factor adjustment. The factor-adjusted estimate has a Newey–West t-statistic of -1.65. It summarises an average decline across the seven nonlinear models, but it should not be read as seven independent experiments. The four neural networks share a February 2015 date, and their portfolio returns are correlated. The continuous indices point in the same direction. A one-standard-deviation increase in the research index is associated with a 0.17% lower monthly alpha ($t = -1.61$). The code estimate is -0.25% ($t = -2.04$), and the hiring estimate is -0.26% ($t = -1.85$). Raw returns show no stable pattern. That the code index is the most significant fits the diffusion mechanism. Public code is the most direct measure of implementability: once a method's implementation is openly available, an investor can run it without reconstructing it from a paper, so the code index most closely tracks the point at which a technique enters investors' toolkits. As more investors deploy the same method, they trade on the same signals, prices adjust against those signals, and the abnormal return erodes. Research output, by contrast, measures disclosed knowledge that still has to be implemented, which loosens its link to deployment and leaves its coefficient insignificant, while finance-sector hiring, though similar in magnitude, is estimated only from 2016 onward and so is less precisely identified. These results are consistent with gradual diffusion, although the later model-level estimates indicate that the relation is not common across all technologies.

Table 2: Adoption and Alpha Decay: Pooled Estimates

This table reports pooled estimates of λ from Equation 3, which regresses next-month value-weighted net long-short returns on the adoption measure $A_{m,t}$, with model fixed effects. The binary row measures adoption with the post-availability indicator $POST_{m,t}$ for the seven nonlinear models, so its coefficient is the average post-availability return shift in percent per month. The remaining rows measure adoption with a standardised, log cumulative index built from a complementary source (arXiv submission counts, GitHub repository counts, and finance-sector online-vacancy counts, Section 4), so their coefficients are in percent per month per one-standard-deviation increase in that model's own adoption. The "Raw" column includes no factor controls; the "Six-factor" column adds the Fama and French (2018) six factors, the Fama–French five factors plus momentum. N is the number of model-month observations. Inference uses Newey–West standard errors. The hiring index has informative variation only from 2016 onward. Stars denote $^{*}p < 0.10$, $^{**}p < 0.05$, and $^{***}p < 0.01$.

Adoption measure $A_{m,t}$	Raw λ	Six-factor λ	N	Sample
Binary availability ($POST$)	−0.033	−0.416*	1,498	2005–2022
Research (arXiv + SSRN)	+0.055	−0.172	2,140	2005–2022
Code (GitHub)	−0.024	−0.250**	1,550	2010–2022
Hiring (vacancy)	+0.208	−0.257*	990	2016–2022

We address cross-model dependence by resampling entire calendar months and treating each model observation from a sampled month as a single block. In the corresponding constant-loading binary regression, the all-model estimate is −0.37% per month with a bootstrap standard error of 0.42% and a two-sided p-value of 0.38. The estimate rises in magnitude to −0.59% for the four networks, with $p = 0.22$. It reaches −0.75% for FFNN2, FFNN4, and FFNN5, the networks with a pre-availability alpha t-statistic above two. Repeating that selection inside every bootstrap draw gives $p = 0.12$. None of these tests reaches conventional significance. We therefore treat the pooled estimates as descriptive and rely on the model-level decomposition and short-interest evidence for the paper's sharper claims. Appendix D.2 reports the full bootstrap table.

5.3 Performance Before and After Availability

Table 3 combines the economic and statistical decomposition. It reports each public date, the alpha on either side, the alpha change, the change relative to Ridge, and the pre-availability shares of time and cumulative log net wealth, with statistical significance marked by stars and the corresponding t-statistics given in the text. Placing time and wealth shares together

makes the anachronistic component visible in a single table. FFNN2 is the clearest case. Its pre-availability alpha is 0.91% per month ($t = 2.80$) and its post-availability alpha is zero. The estimated change is -0.91% ($t = -1.81$). The pre-availability period accounts for 91% of the strategy's cumulative log net wealth, even though it contains only 56% of the sample months. FFNN4 falls from 1.06% to 0.24%, and FFNN5 falls from 0.85% to 0.52%. FFNN3 is unchanged. The date sensitivity in Section 4 shows that moving the network date back to January 2010 leaves the FFNN2 change negative and of similar magnitude. The boosted trees do not share one pattern. XGBoost declines by 0.63% per month, but the effect is not statistically significant. LightGBM and CatBoost improve after availability. Both had weak pre-availability alpha, which matches Equation (1) but limits any broad claim about ML-wide decay. Across the seven nonlinear models, 62% of cumulative log net wealth is earned before public availability, compared with a 58% share of sample months. The four-point pooled gap is driven mainly by FFNN2 and FFNN4.

Table 3: Alpha Before and After Public Availability

This table combines the performance decomposition, model-specific break estimates, and the comparison with Ridge. Alphas are monthly percentage points from regressions of net long-short returns on the Fama and French (2018) six-factor model, the Fama–French five factors plus momentum. The alpha shift is post-availability alpha minus pre-availability alpha. The Ridge-relative shift is estimated after subtracting the always-available Ridge return. Time share is the fraction of sample months before the public date. Wealth share is the fraction of full-sample cumulative log net wealth earned in those months. The Elastic Net row uses a synthetic March 2015 date. Inference uses Newey–West standard errors. Stars denote $^{*}p < 0.10$, $^{**}p < 0.05$, and $^{***}p < 0.01$.

Model	Availability date	Pre alpha	Post alpha	Alpha shift	Ridge-relative shift	Pre time share	Pre wealth share
XGBoost	February 2014	0.929	0.295	−0.633	−0.385	50.2%	58.5%
LightGBM	August 2016	0.273	0.790*	+0.517	+1.105	64.2%	12.3%
CatBoost	July 2017	0.443	0.757	+0.313	+1.351	69.3%	53.7%
FFNN2	February 2015	0.909***	−0.003	−0.913*	−0.617*	55.8%	91.2%
FFNN3	February 2015	0.559*	0.560	+0.000	+0.358	55.8%	54.6%
FFNN4	February 2015	1.063***	0.237	−0.826	−0.271	55.8%	74.5%
FFNN5	February 2015	0.851***	0.517	−0.333	+0.118	55.8%	60.2%
Elastic Net placebo	March 2015 (synthetic)	0.733***	0.719*	−0.014	n.a.	56.7%	n.a.

Subtracting Ridge addresses the possibility that anomaly strategies weakened together late in the sample. The FFNN2 return falls by 0.62% per month after availability relative to Ridge ($t = -1.75$). FFNN4 also falls relative to Ridge, but by only 0.27% and without statistical significance. LightGBM and CatBoost improve relative to Ridge. The pooled Ridge-relative change is positive because those late-dated trees offset the neural-network declines. This comparison supports a model-specific FFNN2 interpretation rather than a common structural break across all ML strategies.

The Elastic Net placebo produces a -0.01% shift at the synthetic March 2015 date. Alternative dates also yield small, insignificant estimates. Model-by-model continuous slopes, reported in Appendix Table A.3, are most negative for FFNN2. The deeper networks are closer to the linear placebos. These results make FFNN2 the most consistent return case rather than evidence of a uniform effect across architectures.

5.4 Further Robustness

The appendix reports the remaining checks. Allowing factor loadings to move with continuous adoption leaves the research coefficient negative, while the code and hiring estimates become smaller. Local windows around the public dates give negative estimates in eleven of twelve specifications, although none is significant. Synthetic Elastic Net dates yield negative real-minus-placebo differences in eight of nine comparisons, with imprecise tests of differences. Together with the bootstrap, these checks show that the direction of the FFNN2 result is stable while the average effect across all models remains uncertain.

6 Market-Based Evidence from Short Interest

The return tests use dates drawn from public research and code. Short sellers' positions provide an outside measure of whether those dates coincide with a change in trading behaviour. The framework predicts closer alignment after a model becomes usable, especially for methods that produced pre-availability alpha.

6.1 Short-Interest Data and Timing

We obtain security-level short interest from the current and legacy Compustat short-interest files. We link securities to CRSP permnos through the CRSP/Compustat Merged Database and retain one observation per stock-month. Our main outcome is the number of shares sold short divided by CRSP shares outstanding:

$$SI_{i,t} = 100 \times \frac{\text{shares sold short}_{i,t}}{\text{shares outstanding}_{i,t}}. \tag{4}$$

The percentage scale makes the outcome comparable across firms and gives the event-study magnitudes a direct interpretation. A change of 0.10 means that short positions rise by one tenth of one percentage point of shares outstanding.

The timing of the test follows the portfolio experiment. A model signal formed with information available in month t is matched to the change in short interest from t to $t+1$. The signal therefore precedes the position change. We do not use short interest to train the return models or to set their availability dates. Missing short-interest observations remain missing rather than being imputed, so each monthly cross-section uses stocks covered in both the signal and position data. This design separates two ideas that could otherwise be conflated. A high short-interest level may reflect slow-moving firm characteristics, financing constraints, or earlier information. The next-month change asks whether short sellers move further against the stocks that a model ranks as unattractive. The availability split then asks whether that conditional alignment changes when the model becomes publicly usable.

6.2 Alignment with Model Short Signals

For each model and month, we rank stocks from the highest to the lowest model-implied short signal. We then estimate the cross-sectional regression

$$\Delta SI_{i,t+1} = a_t^m + \beta_t^m \,\mathrm{Rank}_t\big(-\widehat{r}_{i,t}^m\big) + X_{i,t}'\Gamma_t^m + \varepsilon_{i,t+1}^m, \tag{5}$$

where the controls include firm size, price, the bid-ask spread proxy, lagged short interest, lagged short-interest rank, and exchange fixed effects. A positive β_t^m means that stocks

viewed more negatively by the model experience a larger increase in short interest over the next month. Estimating a separate cross-section every month prevents changes in the number or composition of covered stocks from determining the time-series comparison.

For each model, we summarise the monthly slopes with

$$\widehat{\beta}_t^m = \mu^m + \delta^m \text{Post}_t^m + u_t^m, \tag{6}$$

where Post_t^m begins at the public-availability date set in Section 4. The coefficient δ^m is the post-minus-pre change reported in Table 4. Its standard error is Newey–West, which allows the monthly alignment slopes to be serially correlated. This two-step procedure preserves the Fama–MacBeth logic of the first-stage cross-sections while making the time break explicit.

Table 4 reports the results. Alignment strengthens for every neural network. The post-minus-pre change is 0.036 for FFNN2, 0.053 for FFNN3, 0.045 for FFNN4, and 0.062 for FFNN5. Their significance ranges from the 10% level for FFNN2 to the 1% level for FFNN5. None of the tree models shows a comparable change. XGBoost and LightGBM are close to zero, while CatBoost is negative. The economic size is easier to read against the pre-period slope. FFNN2 moves from 0.043 before availability to 0.079 afterwards. FFNN5 moves from 0.043 to 0.105. A one-unit change in rank spans the cross-section, so these coefficients compare the next-month short-interest change between the least and most negative model rankings, conditional on observable stock characteristics. The estimates show a shift in how short sellers position across stocks, not merely a rise in aggregate short interest.

Appendix Figure A.5 plots the monthly slopes. Appendix E reports alternative controls, the broader ML Reliance measure, and the weak reliance-return tests. Keeping those tests in the appendix avoids treating several related transformations of the same position data as independent evidence.

6.3 Common Short-Leg Entry

The alignment regression measures average co-movement. An event study sharpens the timing around a change in portfolio membership. An event occurs when a stock that belonged

Table 4: Pre- and Post-Adoption Model-Level Short-Seller Alignment

This table summarises time variation in the monthly model-level short-seller alignment slope from Equation 5. For each model we estimate the monthly cross-sectional alignment slope and then estimate the adoption-timing equation $\widehat{\beta}_t^m = \mu^m + \delta^m \text{Post}_t^m + u_t^m$. The break is each model's public availability date (Table A.1). The outcome is the one-month change in short interest, $\Delta SI\%$ (horizon $h = 1$). Columns are the seven nonlinear models: XGB, LGB, and CBT denote XGBoost, LightGBM, and CatBoost; FFNNk is the k-layer feedforward network. A positive Post−pre change means short sellers align more closely with the model's short signal after its public availability. Stars on Post−pre use Newey–West standard errors.

	XGB	LGB	CBT	FFNN2	FFNN3	FFNN4	FFNN5
Panel A: Public availability dates							
Pre-adoption mean	0.114	0.167	0.064	0.043	0.038	0.040	0.043
Post-adoption mean	0.135	0.169	0.032	0.079	0.091	0.085	0.105
Post − pre	0.021	0.002	−0.032	0.036*	0.053**	0.045**	0.062***

to no model's short leg in month $t-1$ enters the short legs of at least two models in month t. For event time $k = -3, \ldots, 6$, we estimate

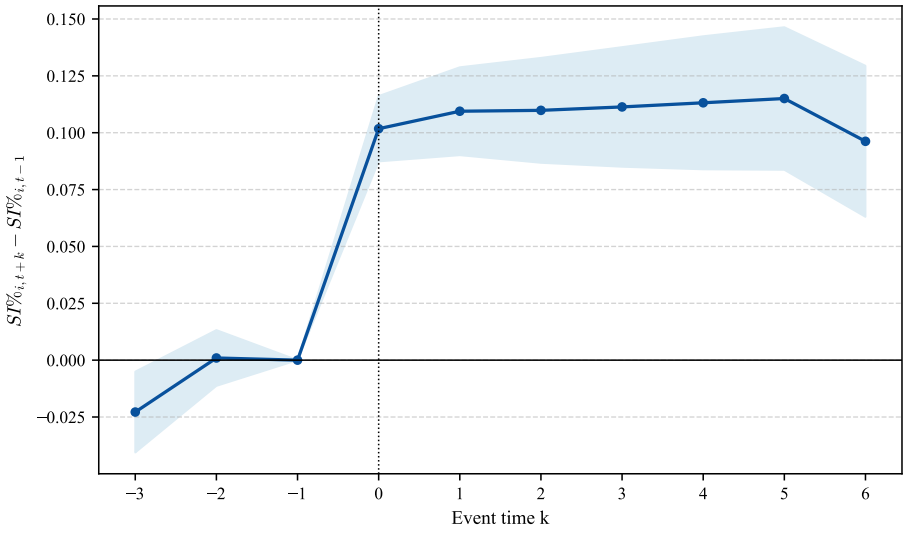
$$\widehat{\Delta SI}_k = \frac{1}{N_k} \sum_{(i,t)\in\mathcal{E}_k} \left(SI_{i,t+k} - SI_{i,t-1}\right), \tag{7}$$

where $\mathcal{E}_k$ contains events retained by the balanced stock-month window and with nonmissing short interest at horizon k. The balanced rule requires a stock observation in every event month. Missing short interest causes the usable event count to vary slightly across horizons. The pointwise confidence intervals use the cross-sectional standard error of the event-level changes.

Figure 3 shows no increase in the two months before entry. In the entry month, short interest rises by 0.10 percentage points of shares outstanding ($t = 13.9$). The increase remains close to 0.10 points six months later. The absence of a preceding build-up makes it less likely that the models merely repackage short-interest changes already under way. The event-time path and the availability split answer different timing questions. The first shows what happens when several signals identify the same stock as unattractive. The second shows whether short sellers align more closely with a model after its public release.

Figure 3: Short Interest Around Common Short-Leg Entry

The figure plots the average change in short interest, $SI\%_{i,t+k} - SI\%_{i,t-1}$, around common short-leg entry events. An event occurs when a stock that belonged to no model's short leg in month $t-1$ enters the short legs of at least two models in month t. Event time runs from three months before to six months after entry. The sample uses balanced stock-month windows, although missing short interest causes small changes in usable N across horizons. The shaded band is the pointwise 95% confidence interval based on cross-sectional standard errors across events. The dotted vertical line marks the entry month.



6.4 What the Market Evidence Identifies

The short-interest results are strongest as evidence that the public dates mark a change in market behaviour. The family split is informative. Alignment strengthens for the networks, which have the clearest pre-availability profits, but not for the tree portfolios, which have weak or improving post-availability returns. That match is difficult to obtain from a generic rise in short selling or from one common end-of-sample trend.

The tests do not identify the code or architecture used by a short seller. Investors may trade on related nonlinear signals, shared data, or research that overlaps with the models. The common-entry event also pools model families and does not isolate a public-availability shock. Stocks can enter more than once and event windows can overlap, so its confidence band is best read as timing evidence rather than as a count of independent shocks. Its value lies in the absence of a pre-trend and the persistence after entry. We therefore read the alignment regression as the main market-based test and the event study as a visual account

of how signal alignment appears in positions.

7 Discussion and Conclusion

A machine-learning backtest can use predictors that were public at the time and still rely on forecasting technology that was not. We measure this timing gap for ten long-short strategies from 2005 to 2022. The return evidence is not uniform across ML methods. It is strongest for the profitable feedforward networks, especially FFNN2, whose alpha falls from 0.91% per month before the tested architecture becomes public to zero afterwards. LightGBM and CatBoost improve after availability, and the pooled pre-availability wealth share exceeds the corresponding time share by only four percentage points. These results support a focused conclusion: some full-sample ML backtests contain returns earned before the tested technology was publicly usable, but the size of that component depends on the model. This heterogeneity is itself part of the result. The boosted-tree estimates move in both directions, while the always-available linear methods remain strong over the full sample. The framework predicts decay only when a technology begins with enough net alpha to attract costly arbitrage, and the data do not show that every complex method clears that threshold.

Short sellers' positions give the most direct evidence that public availability marks an economic change. Neural-network short signals align more closely with subsequent short-interest changes after the networks become public. The tree models show no similar shift. Short interest also rises when stocks enter several model short legs at once, with no preceding increase. The evidence is consistent with investors trading on overlapping ML information after diffusion. It cannot establish that they run the exact architectures in our tests.

The February 2015 network date belongs to the tested stack. It is not a claim that neural-network investing began in 2015. Earlier return applications and generic software are well documented. Public records also cannot reveal proprietary use before disclosure, so any availability date is an observable proxy for the edge of a private adoption process. Moving the network date to the first generic repository in January 2010 attenuates the average network decline and leaves the FFNN2 change negative. That exercise does not settle the date, but

it shows which conclusion survives a much earlier lower bound. The market tests establish a change in market alignment, not direct model adoption and not a complete causal channel from publication to alpha decay. The return evidence has a related limit: seven nonlinear models provide a small cross-section, four networks share one date, and their returns covary. The month-block bootstrap makes the resulting uncertainty visible and is why we do not use the pooled coefficient as a decisive test.

The practical prescription is reporting discipline. Public-availability dates should be treated as part of a real-time backtest, alongside vintage data, signal-publication dates, and trading costs. A full-sample ML backtest measures the historical fit of a fixed modern technology, while a technology-feasible backtest measures performance after that technology becomes publicly usable. Both quantities are useful, but they answer different questions. Pre-availability results remain informative about a model's statistical capacity, but they should not be read as returns available to historical investors or carried unchanged into forward-looking expectations. Reporting the availability decision, a reasonable earlier alternative, and performance on each side of the date lets readers separate a model's statistical capacity from an investment record a historical investor could have earned.

A Theory Derivations

This appendix derives the framework in Section 2. The derivation clarifies the role of public availability, implementation costs, and limited arbitrage capital.

A.1 Diffusion and Equilibrium Alpha

Adoption of technology m begins at its public-availability date τ_m and then follows an S-shaped path:

$$a_{m,t} = \begin{cases} 0, & t < \tau_m, \\ \dfrac{\bar{a}}{1 + \exp\{-\rho(t - \tau_m - h)\}}, & t \geq \tau_m. \end{cases} \tag{A.1}$$

The parameter ρ controls diffusion speed, h is the half-life, and $\bar{a} < 1$ reflects limits on arbitrage capital. An adopter with risk aversion γ chooses a position in a strategy with variance σ_m^2 and implementation cost c_m. Mean-variance demand is

$$x_{m,t} = \frac{\max\{\alpha_{m,t} - c_m, 0\}}{\gamma \sigma_m^2}. \tag{A.2}$$

Aggregate demand is $a_{m,t} x_{m,t}$. If its price impact is ϕ_m, equilibrium alpha satisfies $\alpha_{m,t} = \alpha_m^0 - \phi_m a_{m,t} x_{m,t}$. For $\alpha_m^0 > c_m$, substitution gives

$$\begin{aligned} (\alpha_{m,t} - c_m)(1 + \kappa_m a_{m,t}) &= \alpha_m^0 - c_m, \\ \kappa_m &= \frac{\phi_m}{\gamma \sigma_m^2}. \end{aligned} \tag{A.3}$$

Solving the first line yields Equation (1). If $\alpha_m^0 \leq c_m$, no adopter trades and alpha remains at α_m^0. The response to adoption is

$$\frac{\partial \alpha_{m,t}}{\partial a_{m,t}} = -\frac{\kappa_m(\alpha_m^0 - c_m)}{(1 + \kappa_m a_{m,t})^2}. \tag{A.4}$$

Alpha falls only when the strategy clears its implementation cost. The decline is larger when initial net alpha is larger. Since adoption is bounded by $\bar{a}$, alpha also remains above the cost floor.

A.2 Backtest and Implementable Alpha

Consider a sample from 0 to T with an availability date τ_m inside it. Define implementable alpha as the average from τ_m to T. The full-sample average can then be written as

$$\begin{aligned} \bar{\alpha}_m &= \frac{1}{T}\left[\tau_m \alpha_m^0 + \int_{\tau_m}^{T} \alpha_{m,t}\, dt\right] \\ &= \frac{\tau_m}{T}\alpha_m^0 + \frac{T-\tau_m}{T}\alpha_m^{\text{impl}}, \end{aligned} \tag{A.5}$$

$$\bar{\alpha}_m - \alpha_m^{\text{impl}} = \frac{\tau_m}{T}(\alpha_m^0 - \alpha_m^{\text{impl}}) > 0.$$

The last inequality holds when the technology has positive erodible alpha. The gap widens with the pre-availability share of the sample and with the difference between initial and post-availability alpha.

B Data and Implementation Details

B.1 Stock Universe and Delisting Returns

The CRSP sample uses exchange codes 1, 2, and 3 and share codes 10 and 11. It therefore contains common shares listed on NYSE, AMEX, and Nasdaq. The return target includes delisting returns. When a performance-related delisting return is missing, we replace it with -35% for NYSE and AMEX stocks (Shumway, 1997) and -55% for Nasdaq stocks (Shumway and Warther, 1999), addressing the delisting bias documented in these papers.

B.2 Predictor Processing and Model Fitting

The anomaly data come from the `openassetpricing` implementation of Chen and Zimmermann (2022). Within each month, each continuous feature is mapped to a percentile rank between -1 and 1. Missing observations are set to zero, which is the cross-sectional midpoint after ranking. Industry enters through 48 indicators, with one of the 49 Fama–French industries omitted. The macroeconomic features are the dividend-price ratio, earnings-price ratio, book-to-market ratio, net equity expansion, Treasury-bill rate, term spread, default spread, and stock variance (Welch and Goyal, 2008).

Each anomaly becomes eligible only after its publication date. We keep all eligible signals rather than selecting the subset with the strongest historical univariate performance. This choice follows the high-dimensional forecasting approach of Kelly et al. (2024) and avoids ranking features with information from the later test sample. Tree hyperparameters are updated annually with the tree-structured Parzen estimator (Akiba et al., 2019). A grid search could recover the same configurations at greater computational cost, so the tuning procedure is not used to date the tree methods. The neural networks use the architecture components listed in Section 3.

The annual split is fixed before any model is estimated. For prediction year y, the training set contains every observation through $y-7$, the validation set contains years $y-6$ through $y-1$, and the test set contains year y. The training window expands by one year at

each re-estimation. The six-year validation block remains fixed in length. Linear penalties are selected by grid search on that block. Each tree model uses 40 tree-structured Parzen trials and early stopping, then produces monthly forecasts for the untouched test year.

FFNN2 through FFNN5 use two to five hidden layers. Hidden widths follow a geometric decline from the input dimension. Every hidden layer applies batch normalisation and a leaky-ReLU activation. Each network minimises mean squared error with Adam at an initial learning rate of 0.01, an L1 penalty of 10^{-5}, batches of at most 10,000 observations, and at most 200 epochs. The learning rate falls when validation loss stops improving, and training ends after 20 epochs without improvement. Reported forecasts average five independently initialised networks. These choices are fixed rather than re-tuned on each test year.

B.3 Trading Costs

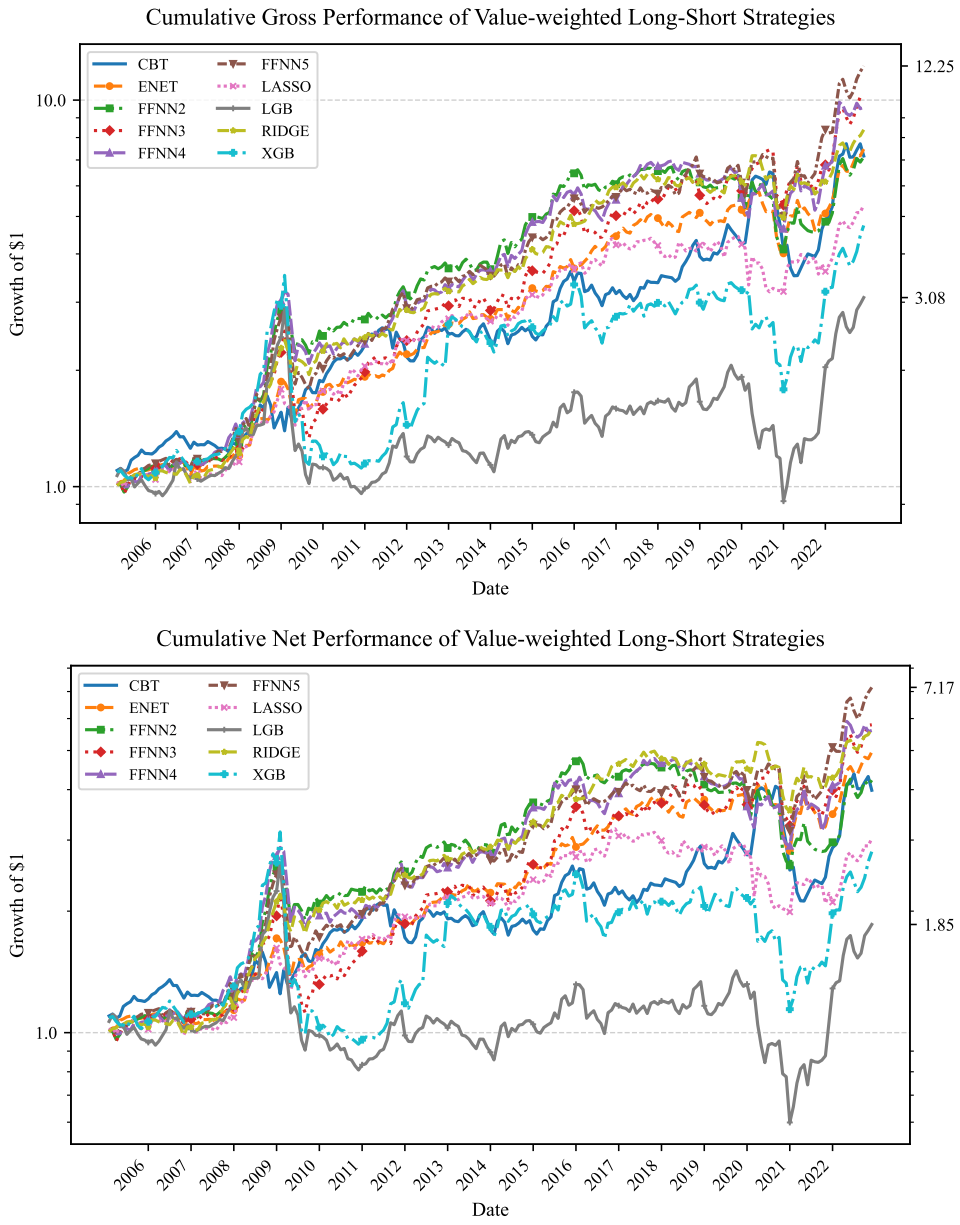
For each portfolio leg, the cost at rebalancing is

$$\begin{aligned} TC_t^{\text{leg}} &= \sum_i \left| w_{i,t}^{\text{target}} - w_{i,t}^{\text{owned}} \right| bas_{i,t}, \\ w_{i,t}^{\text{owned}} &\propto (1 + r_{i,t}) w_{i,t-1}^{\text{target}}, \end{aligned} \tag{A.6}$$

where $bas_{i,t}$ is the effective-spread estimate of Chen and Velikov (2023). The owned weight includes return-driven drift between formation dates. Net long-short returns subtract the costs of both legs. Figure A.1 shows cumulative gross and net wealth for all ten strategies. The figure complements the monthly summaries in Table 1.

Figure A.1: Cumulative Gross and Net Portfolio Wealth

The figure plots cumulative wealth for the ten value-weighted long-short portfolios from February 2005 to December 2022. The upper panel uses gross returns. The lower panel subtracts effective-spread trading costs. Both vertical axes are logarithmic.



C Adoption Sources and Date Construction

The three adoption series use a common stock construction but measure different stages of diffusion. This section describes what enters each series and how the public dates are selected.

C.1 Research, Code, and Hiring Records

The research index combines arXiv with Semantic Scholar records for SSRN papers and journal articles. A record enters only when its title or abstract identifies both machine learning and asset pricing. Duplicates are matched by DOI and title, and the earliest public date is retained. Titles and abstracts also assign papers to linear models, boosted trees, and neural networks. A paper may count towards more than one family when it studies several methods. The final corpus contains about 1,900 papers, including the early neural-network return study of White (1988).

The code index counts public GitHub repositories associated with a library or algorithm. Repository metadata assign each project to a model family, and the monthly stock counts projects in existence by that date. Public code is the most direct measure of practical implementability. The source has two limits. Coverage is thin in the early years of GitHub, and deleted or private repositories do not appear in the current snapshot. The first observed repository is therefore a conservative availability marker rather than a complete history of code use.

The hiring index uses Revelio Labs vacancies at US finance-sector employers, identified by NAICS code 52 and linked to listed firms. We classify postings with a transparent keyword dictionary rather than a learned classifier or any large language model, following the online-vacancy approach of Acemoglu et al. (2022). Revelio supplies the raw posting text, namely the title, description, and standardised skills, and we search it for model-identifying terms: for example, "xgboost" or "extreme gradient boosting" for XGBoost, "lightgbm" for LightGBM, "catboost" for CatBoost, deep-learning frameworks such as "pytorch," "tensorflow," "lstm," or "transformer" for the neural networks, and "elastic net," "lasso," or "ridge regression" for the linear models. A posting counts toward a model only when such a term

Table A.1: Model Dates and Dating Decisions

The table reports the dates used to construct the binary availability indicators. Linear models predate the out-of-sample portfolio period and serve as always-available comparisons.

Model	Public date	Reference	Dating basis
Ridge	January 1970	Hoerl and Kennard (1970)	Original publication
Lasso	January 1996	Tibshirani (1996)	Original publication
Elastic Net	March 2005	Zou and Hastie (2005)	Original publication
XGBoost	February 2014	Chen and Guestrin (2016)	First public implementation
LightGBM	August 2016	Ke et al. (2017)	First public implementation
CatBoost	July 2017	Prokhorenkova et al. (2018)	First public implementation
FFNN2 to FFNN5	February 2015	Ioffe and Szegedy (2015)	Final component of the tested stack

co-occurs with a finance or trading term, such as "alpha," "signal," "systematic," "quant," "portfolio," "factor model," or "trading," which keeps unrelated demand, such as a machine-learning role in fraud detection, out of the index. Revelio coverage becomes useful in the mid-2010s, so this index begins in 2016. It measures employer demand after a method reaches the finance labour market rather than the first moment the method could be implemented.

C.2 Availability Decisions

Table A.1 records the dates. The tree dates come from their first public implementations, which precede the canonical papers. Linear methods retain their original publication dates because they predate GitHub and the research corpus used for the monthly index. The feedforward-network date refers to the tested stack. Generic neural networks, return-prediction applications, and repositories existed earlier. The backtest uses leaky-ReLU activations, batch normalisation, and Adam. Batch normalisation was the final required component to become public, in February 2015 (Ioffe and Szegedy, 2015). Adam appeared in December 2014 (Kingma and Ba, 2014). Dating the networks from the first generic repository in January 2010 weakens the average estimated decline but does not reverse it. This alternative is a lower bound because it treats a generic implementation as equivalent to the architecture used in the portfolio test.

Table A.2: Neural-Network Date Sensitivity

The table re-estimates the model-specific break regression after moving the date for every feedforward network from February 2015 to January 2010, the first generic neural-network repository in the code sample. Entries are monthly percentage-point alpha shifts. The baseline has 120 pre-date and 95 post-date months. The alternative has 59 pre-date and 156 post-date months. Stars are based on Newey–West standard errors and denote $^{*}p < 0.10$, $^{**}p < 0.05$, and $^{***}p < 0.01$.

Model	February 2015 baseline	January 2010 alternative
FFNN2	−0.913*	−0.894*
FFNN3	+0.000	+0.225
FFNN4	−0.826	−0.695
FFNN5	−0.333	−0.330
Equal-weighted average	−0.518	−0.424

D Additional Return Evidence

The main text reports the pooled estimates and the combined before-and-after table. This section shows the model-level continuous estimates, alternative factor-loading assumptions, benchmark differences, local windows, and placebo dates.

D.1 Continuous Adoption by Model

Table A.3 estimates Equation (3) separately for each portfolio. FFNN2 is negative on all three sources. LightGBM and CatBoost have flat or positive slopes, while the linear models are statistically indistinguishable from zero. The Ridge slopes are similar in magnitude to those of FFNN4 and FFNN5, so the placebo separation is clearest for FFNN2.

D.2 Month-Block Bootstrap

The analytic pooled standard errors treat the return panel parametrically. The four networks share one release date and move together, so we also resample entire calendar months. Each bootstrap draw samples 215 months with replacement and carries all model observations in a selected month into the draw. We then re-estimate the constant-loading binary regression. This preserves the observed cross-model dependence within each month.

Table A.3: Continuous Adoption and Alpha Decay by Model

Each cell reports the factor-adjusted estimate of λ from a separate model regression. Coefficients are monthly percentage points for a one-standard-deviation increase in the adoption index. Stars are based on Newey–West standard errors and denote $^{*}p < 0.10$, $^{**}p < 0.05$, and $^{***}p < 0.01$.

Model	Research	Public code	Finance hiring
Elastic Net	−0.141	−0.237	−0.259
Lasso	−0.300	−0.311	−0.150
Ridge	−0.323	−0.309	−0.366
XGBoost	−0.328	−0.447	+0.046
LightGBM	+0.092	+0.086	+0.123
CatBoost	+0.078	+0.165	−0.328
FFNN2	−0.419	−0.537*	−0.496
FFNN3	−0.029	−0.259	−0.353
FFNN4	−0.229	−0.318	−0.240
FFNN5	−0.092	−0.208	−0.131

Table A.4: Month-Block Bootstrap of the Pooled Availability Shift

The table reports constant-loading binary-availability regressions. Whole calendar months are sampled with replacement in 5,000 draws. The final row reselects models whose pre-availability alpha meets the five-percent significance threshold inside each draw.

Model pool	Shift (%)	Bootstrap SE (%)
All seven nonlinear models	−0.371	0.422
FFNN2 through FFNN5	−0.594	0.484
FFNN2, FFNN4, and FFNN5, fixed	−0.747	0.487
Pre-alpha selection repeated in draw	−0.747	0.479

Table A.4 reports 5,000 draws. The all-model coefficient is −0.37% with a bootstrap standard error of 0.42% and $p = 0.38$. Restricting the pool to the four networks raises the magnitude to −0.59%, but $p = 0.22$. Selecting FFNN2, FFNN4, and FFNN5 from the full sample gives −0.75% and $p = 0.13$. The final row repeats the pre-alpha selection inside each bootstrap draw, which gives $p = 0.12$. The dependence-robust exercise supports the paper's decision to treat the pooled result as descriptive.

Table A.5: Adoption and Alpha Decay with Changing Factor Loadings

The first estimate repeats the baseline factor-adjusted coefficient. The second allows every factor loading to move with adoption. Coefficients are monthly percentage points. Stars are based on Newey–West standard errors and denote $^{*}p < 0.10$ and $^{**}p < 0.05$.

Adoption measure	Fixed loadings	Changing loadings	N	Sample
Binary availability	-0.416^{*}	-0.336	1,498	2005 to 2022
Research	-0.172	-0.174^{*}	2,140	2005 to 2022
Public code	-0.250^{**}	-0.113	1,550	2010 to 2022
Finance hiring	-0.257^{*}	-0.207	990	2016 to 2022

D.3 Changing Factor Loadings

The baseline specification keeps factor loadings fixed. We also estimate

$$\begin{aligned} r^{LS}_{m,t+1} = \alpha_m + \lambda_m A_{m,t} + \beta_m^\top F_{t+1} \\ + \Delta\beta_m^\top (A_{m,t} F_{t+1}) + \varepsilon_{m,t+1}. \end{aligned} \tag{A.7}$$

The interaction term permits systematic risk exposures to change with adoption. Table A.5 shows that the binary and research estimates remain similar. Code and hiring coefficients become smaller and lose significance.

D.4 Returns Relative to Ridge

To absorb influences common to all long-short portfolios, we difference each nonlinear model's return from the always-available Ridge return:

$$r^{LS}_{m,t+1} - r^{LS}_{\text{Ridge},t+1} = \alpha_m + \delta_m \text{Post}_{m,t} + \beta_m^\top F_{t+1} + \varepsilon_{m,t+1}. \tag{A.8}$$

Table A.6 reports the results. FFNN2 falls by 0.62% per month after availability relative to Ridge. The estimate has a t-statistic of -1.75. The pooled estimate is positive because the two late-dated tree models improve.

D.5 Complete Break Estimates

Table A.7 gives the intercepts, factor loadings, and post-availability changes from Equation (A.7). It shows which shifts in portfolio exposure accompany the alpha estimates in

Table A.6: Post-Availability Return Shifts Relative to Ridge

The dependent variable is the model's net long-short return minus the Ridge return. The factor-adjusted column includes the Fama and French (2018) six factors. Coefficients are monthly percentage points. Stars are based on Newey–West standard errors and denote $^{*}p < 0.10$.

Model	Raw shift	Factor-adjusted shift	N
XGBoost	+0.037	−0.385	215
LightGBM	+1.401	+1.105	215
CatBoost	+1.341	+1.351	215
FFNN2	−0.517	−0.617*	215
FFNN3	+0.486	+0.358	215
FFNN4	−0.151	−0.271	215
FFNN5	+0.283	+0.118	215
Pooled, model fixed effects	+0.382	+0.185	1,505

Table 3.

Table A.7: Complete Model-Specific Parameter Estimates

This table reports all coefficients from Equation (A.7). Each cell is a coefficient estimate with significance stars based on Newey–West HAC standard errors. Intercepts and POST coefficients are monthly percentage points; factor loadings and loading shifts are reported in beta units. Columns are the seven nonlinear models (CBT, LGB, and XGB denote CatBoost, LightGBM, and XGBoost; FFNNk is the k-layer network) and PLACEBO, the Elastic Net assigned a synthetic March 2015 date. The real-date Elastic Net is omitted because its public-availability date leaves too few pre-availability observations for a stable model-specific break regression.

Coefficient	CBT	FFNN2	FFNN3	FFNN4	FFNN5	LGB	XGB	PLACEBO
α_{pre}	0.443	0.909^{***}	0.559^{*}	1.063^{***}	0.851^{***}	0.273	0.929	0.733^{***}
$\Delta\alpha$	0.313	-0.913^{*}	0.000	−0.826	−0.333	0.517	−0.633	−0.014
β_{MKT}	0.211^{**}	−0.043	0.020	−0.125	−0.087	-0.358^{***}	−0.266	0.054
β_{SMB}	−0.113	0.118	0.342	0.020	0.051	-0.700^{***}	−0.449	−0.117
β_{HML}	0.812^{***}	0.080	−0.160	0.120	−0.161	0.409	0.357	-0.286^{*}
β_{RMW}	0.278	0.758^{***}	0.635^{**}	0.521^{**}	0.613^{***}	0.926^{***}	0.619	−0.094
β_{CMA}	−0.512	0.519^{**}	0.768^{**}	0.303	0.794^{***}	0.231	0.372	0.635^{***}
β_{MOM}	0.306^{***}	0.441^{***}	0.540^{***}	0.587^{***}	0.588^{***}	0.674^{***}	0.827^{***}	0.231^{***}
$\Delta\beta_{\text{MKT}}$	-0.361^{*}	−0.061	−0.194	0.059	−0.104	0.036	−0.016	-0.225^{*}
$\Delta\beta_{\text{SMB}}$	−0.303	−0.206	−0.273	0.157	−0.132	-0.659^{**}	−0.477	0.142
$\Delta\beta_{\text{HML}}$	-0.802^{*}	0.537	0.610	0.809^{**}	0.650^{*}	0.729^{**}	0.595	0.656^{*}
$\Delta\beta_{\text{RMW}}$	0.548	−0.147	0.447	0.456	0.614^{**}	0.036	0.465	0.577^{**}
$\Delta\beta_{\text{CMA}}$	0.455	−0.739	−0.871	−0.697	−0.797	−0.505	−0.842	-0.755^{*}
$\Delta\beta_{\text{MOM}}$	0.014	0.145	0.166	−0.008	−0.039	-0.444^{*}	-0.427^{**}	−0.012

Table A.8: Local Regressions Around Public Availability

The table estimates the pooled alpha change in symmetric windows of T years around each model's date. The full specification allows factor loadings to change. The restricted specification holds them fixed. Standard errors are Newey–West, and stars denote $^{*}p < 0.10$, $^{**}p < 0.05$, and $^{***}p < 0.01$; no window estimate reaches significance.

	Changing loadings			Fixed loadings		
T	Alpha shift (%)	N	Models	Alpha shift (%)	N	Models
7	−0.437	1,156	7	−0.640	1,156	7
6	−0.028	1,008	7	−0.327	1,008	7
5	−0.114	847	7	−0.340	847	7
4	−0.259	679	7	−0.382	679	7
3	−0.516	511	7	−0.345	511	7
2	−0.244	343	7	+0.035	343	7

Table A.9: Alternative Elastic Net Placebo Dates

Each row assigns a synthetic date to Elastic Net. The real-minus-placebo column tests whether the change for the treated models differs from the placebo change. Coefficients are monthly percentage points. Stars denote $^{*}p < 0.10$, $^{**}p < 0.05$, and $^{***}p < 0.01$; the Newey–West real-minus-placebo difference tests are not significant at conventional levels.

Placebo date	Placebo change	Real minus placebo	Implied real change	N treated
March 2010	−0.224	−0.066	−0.290	1,505
March 2011	+0.046	−0.336	−0.290	1,505
March 2012	−0.033	−0.258	−0.290	1,505
March 2013	−0.142	−0.148	−0.290	1,505
March 2014	−0.028	−0.262	−0.290	1,505
March 2015	−0.014	−0.276	−0.290	1,505
March 2016	−0.029	−0.261	−0.290	1,505
March 2017	−0.361	+0.071	−0.290	1,505
March 2018	−0.211	−0.079	−0.290	1,505

D.6 Local Windows and Synthetic Dates

Table A.8 narrows the sample to symmetric windows around each model's date. Eleven of twelve estimates are negative. Precision falls as the windows become shorter. Table A.9 assigns synthetic dates to Elastic Net. The placebo changes are small and insignificant, and the real-minus-placebo estimate is negative in eight of nine comparisons.

D.7 Model-Level Return Figures

Figure A.2 shows the break estimates and their confidence intervals. Figure A.3 compares each change with pre-availability alpha.

Figure A.2: Model-Specific Alpha Shifts

Dots are post-availability alpha changes from Equation (A.7). Thick and thin lines are 90% and 95% Newey–West confidence intervals. The Elastic Net placebo uses March 2015.

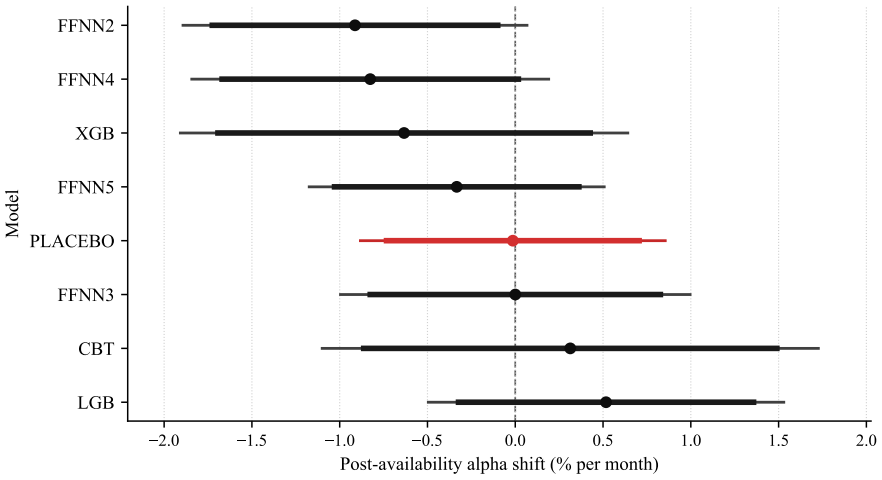


Figure A.3: Alpha Change and Pre-Availability Alpha

The figure plots each model's alpha change against its pre-availability alpha. Both estimates come from the same regression. The fitted relation is descriptive because the vertical-axis measure contains the horizontal-axis estimate.

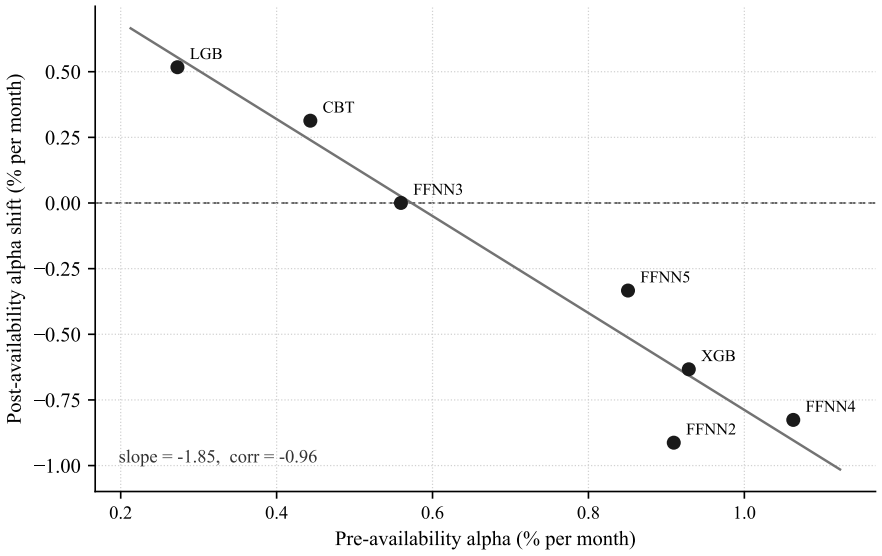


Table A.10: ML Reliance Summary Statistics

ML Reliance is the incremental R^2 from Equation (A.11). Values are multiplied by 10^4. Each model has 214 monthly observations.

	ENET	XGB	LGB	CBT	FFNN2	FFNN3	FFNN4	FFNN5
Mean	2.81	3.54	3.54	3.84	3.98	4.37	3.77	3.96
Standard deviation	4.69	4.27	4.72	5.42	6.28	5.81	5.30	6.03
Pre-availability mean	0.10	3.81	3.66	3.46	3.89	3.95	3.68	3.82
Post-availability mean	2.84	3.26	3.31	4.73	4.08	4.92	3.89	4.14
Post minus pre	2.74	−0.54	−0.35	1.28	0.18	0.97	0.21	0.33

E Additional Short-Interest Evidence

The main text keeps the alignment regression and the common-entry event study. This section gives the broader reliance measure, its calendar-time tests, and supporting figures.

E.1 Constructing ML Reliance

For model m and month t, we first regress the next-month change in short interest on public firm information:

$$\Delta SI_{i,t+1} = \alpha_t + X'_{i,t}\Gamma + \varepsilon_{i,t+1}. \tag{A.9}$$

We then add model-implied shortness, $s^m_{i,t} = \text{Rank}_t(-\widehat{r}^m_{i,t})$:

$$\Delta SI_{i,t+1} = \alpha_t + X'_{i,t}\Gamma + \theta^m_t s^m_{i,t} + u_{i,t+1}. \tag{A.10}$$

The controls include market equity, price, the bid-ask spread proxy, lagged short interest, lagged short-interest rank, realised volatility, turnover, illiquidity, exchange indicators, industry indicators, and published anomaly signals. ML Reliance is the change in explanatory power:

$$\text{MLReliance}_{m,t} = R^{2,\text{aug}}_{m,t} - R^{2,\text{base}}_t. \tag{A.11}$$

E.2 Calendar-Time Reliance

We test for a change in the level of ML Reliance using

$$\text{MLReliance}_{m,t} = \alpha_m + \delta \text{Post}^m_t + e_{m,t}. \tag{A.12}$$

Table A.11: Public Availability and ML Reliance

The dependent variable is monthly ML Reliance. The post coefficient is multiplied by 10^4. Both specifications include model fixed effects. The second adds a linear calendar trend. Stars are based on Newey–West standard errors and denote $^{*}p < 0.10$, $^{**}p < 0.05$, and $^{***}p < 0.01$.

	Model fixed effects	Model fixed effects and trend
Post coefficient	0.281	−0.889
Model-months	1,498	1,498
Models	7	7
R^2	0.003	0.007

The estimate is positive without a trend and negative with a linear calendar trend. Neither is statistically significant. Monthly incremental R^2 is noisy, so the level test is less informative than the alignment slopes in the main text.

E.3 Supporting Alignment Figures

Figure A.5 plots the monthly alignment slopes behind Table 4. Figure A.6 shows that model short legs contain more high-short-interest stocks than an unconditional sort would imply.

E.4 Reliance and Subsequent Returns

The final test asks whether higher current reliance predicts lower next-month portfolio returns:

$$r^{LS}_{m,t+1} = \alpha_m + \lambda \text{MLReliance}_{m,t} + \beta_m^{\top} F_{t+1} + \eta_{m,t+1}. \tag{A.13}$$

The estimate in Table A.12 is negative but imprecise. A related classification test labels a model-month as adoption-positive when the model signal adds explanatory power at the 1% level and exceeds the model's pre-availability 90th percentile. Table A.13 reports those rates. They rise after availability for most models, but not for FFNN2.

Figure A.4: Calendar-Time ML Reliance

Light lines are monthly ML Reliance estimates. Dark lines are twelve-month centred moving averages. Public dates are omitted so the paths can be viewed independently of the chosen breakpoints.

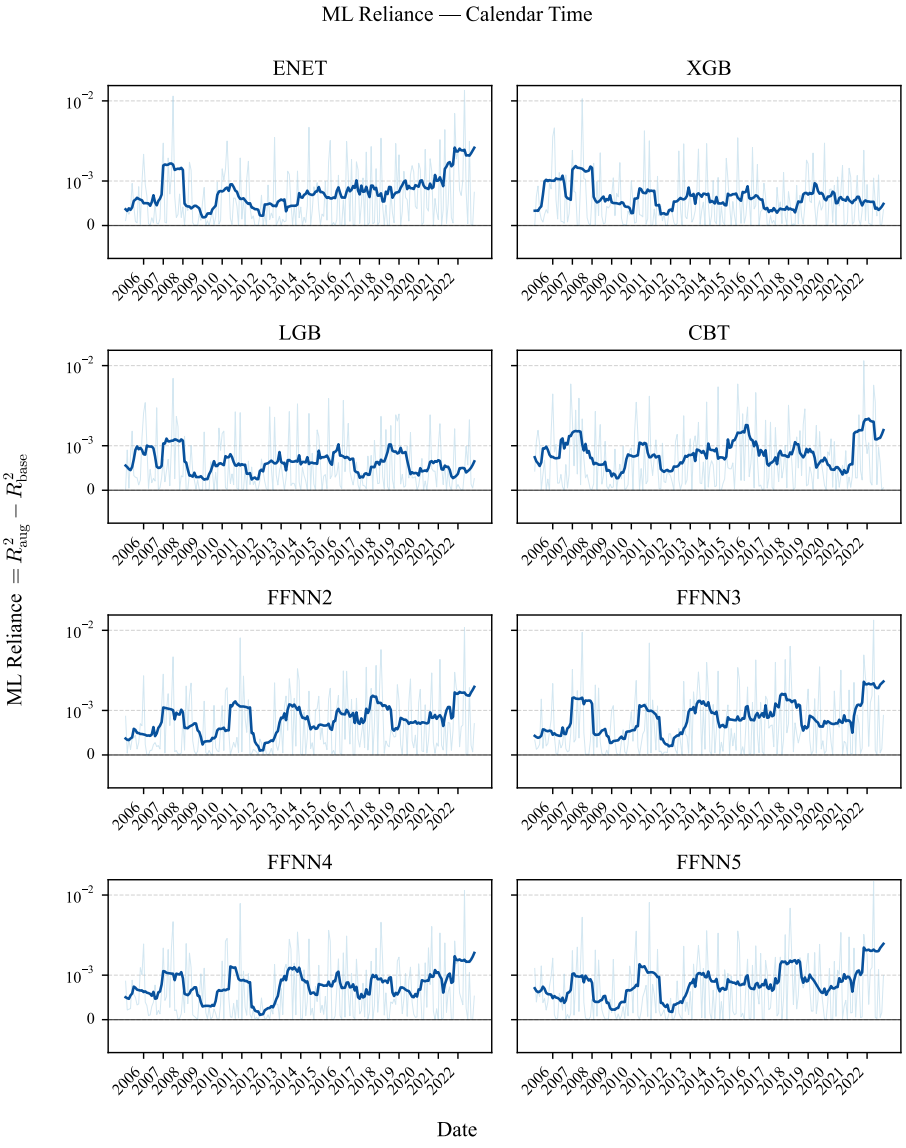


Figure A.5: Time-Varying Short-Seller Alignment

The figure plots monthly slopes from Equation (5). Positive values mean that stocks ranked as less attractive by a model experience larger subsequent increases in short interest. Dark lines are twelve-month centred moving averages. Vertical lines mark public availability.

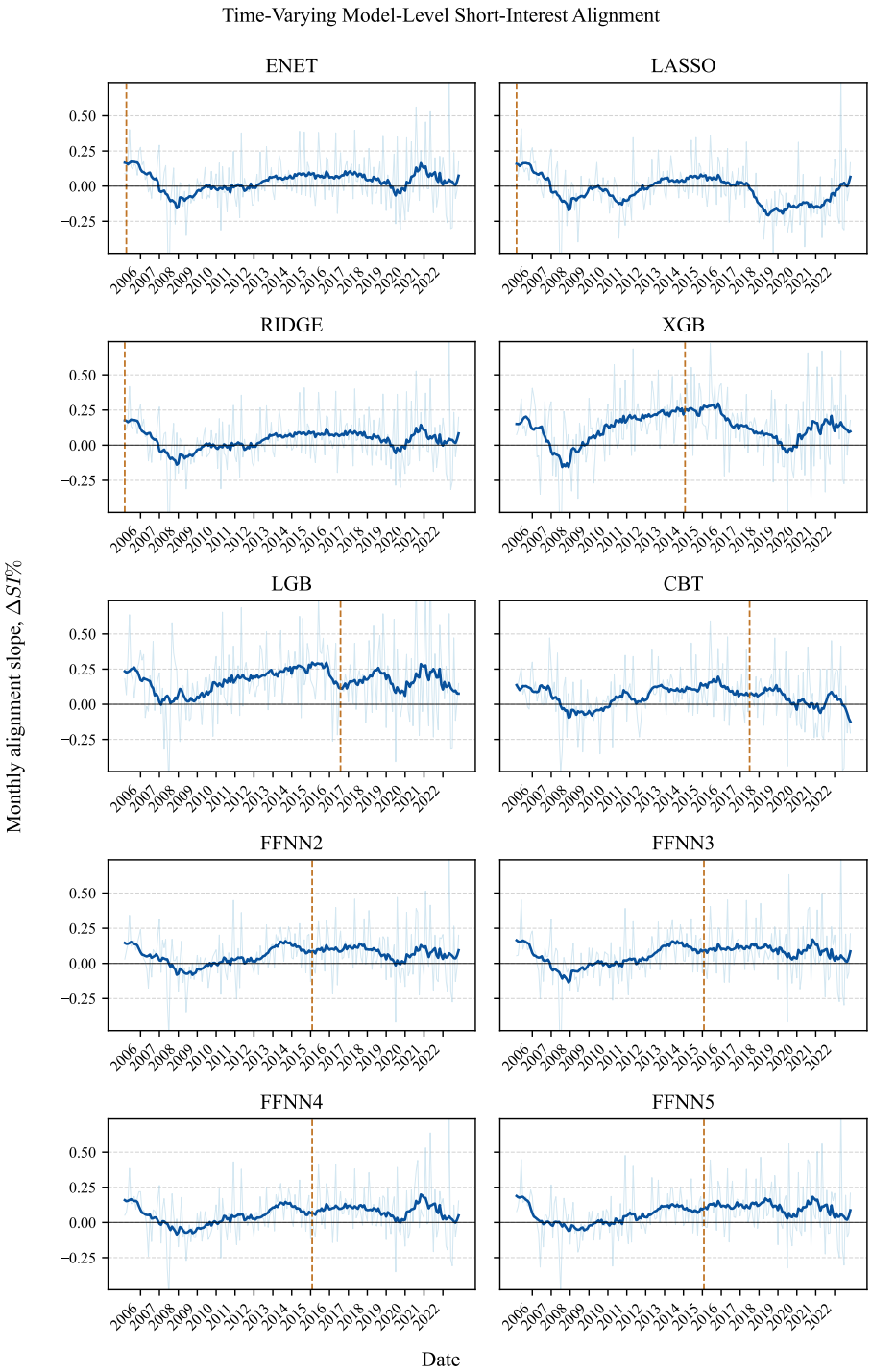


Figure A.6: Model Short Legs and High Short Interest

The figure reports the share of each model's short leg that lies in the top short-interest tercile or quintile. Horizontal lines mark the unconditional shares of 33.3% and 20.0%.

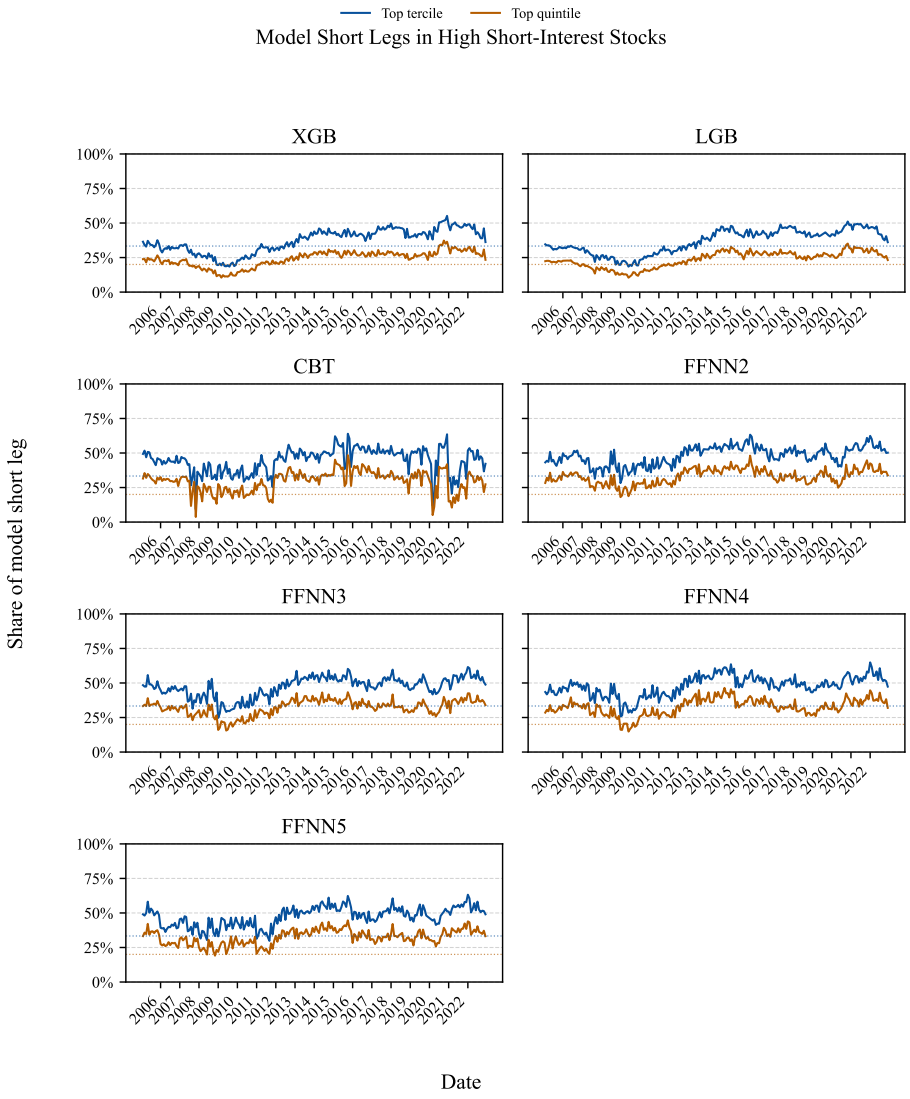


Table A.12: ML Reliance and Future Portfolio Returns

The table estimates Equation (A.13) with model fixed effects and model-specific Fama and French (2018) six-factor loadings. The outcome is next-month net long-short return. Stars are based on Newey–West standard errors and denote $^{*}p < 0.10$, $^{**}p < 0.05$, and $^{***}p < 0.01$.

	λ	Model-months	Models	R^2
Net long-short return	−0.572	1,498	7	0.422

Table A.13: Short-Side Adoption Classification

Entries are percentages of model-months classified as adoption-positive. The adjusted post rate subtracts the pre-availability rate and is bounded below at zero.

	All	XGB	LGB	CBT	FFNN2	FFNN3	FFNN4	FFNN5
Pre-availability rate	2.5	1.9	2.2	2.0	5.0	0.8	3.3	2.5
Post-availability rate	4.8	1.9	2.6	6.2	3.2	8.5	4.3	7.4
Post minus pre	2.3	0.0	0.5	4.1	−1.8	7.7	0.9	4.9
Adjusted post rate	2.3	0.0	0.5	4.1	0.0	7.7	0.9	4.9

References

Acemoglu, D., D. Autor, J. Hazell, and P. Restrepo, 2022, Artificial intelligence and jobs: Evidence from online vacancies, *Journal of Labor Economics* 40, S293–S340.

Akiba, T., S. Sano, T. Yanase, T. Ohta, and M. Koyama, 2019, Optuna: A next-generation hyperparameter optimization framework, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Avramov, D., S. Cheng, and L. Metzker, 2023, Machine learning vs. economic restrictions: Evidence from stock return predictability, *Management Science* 69, 2587–2619.

Azevedo, V., C. Hoegner, and M. Velikov, 2023, The expected returns on machine-learning strategies, working paper.

Bloom, N., C. I. Jones, J. Van Reenen, and M. Webb, 2020, Are ideas getting harder to find?, *American Economic Review* 110, 1104–1144.

Chen, A. Y., and M. Velikov, 2023, Zeroing in on the expected returns of anomalies, *Journal of Financial and Quantitative Analysis* 58, 968–1004.

Chen, A. Y., and T. Zimmermann, 2022, Open source cross-sectional asset pricing, *Critical Finance Review* 11, 207–264.

Chen, L., M. Pelger, and J. Zhu, 2024, Deep learning in asset pricing, *Management Science* 70, 714–750.

Chen, T., and C. Guestrin, 2016, Xgboost: A scalable tree boosting system, *arXiv preprint arXiv:1603.02754* .

Comin, D., and B. Hobijn, 2010, An exploration of technology diffusion, *American Economic Review* 100, 2031–2059.

DeMiguel, V., A. Martin-Utrera, F. J. Nogales, and R. Uppal, 2020, A transaction-cost perspective on the multitude of firm characteristics, *Review of Financial Studies* 33, 2180–2222.

Detzel, A., R. Novy-Marx, and M. Velikov, 2023, Model comparison with transaction costs, *The Journal of Finance* 78, 1743–1775.

Fama, E. F., and K. R. French, 1997, Industry costs of equity, *Journal of Financial Economics* 43, 153–193.

Fama, E. F., and K. R. French, 2018, Choosing factors, *Journal of Financial Economics* 128, 234–252.

Freyberger, J., A. Neuhierl, and M. Weber, 2020, Dissecting characteristics nonparametrically, *The Review of Financial Studies* 33, 2326–2377.

Green, J., J. R. M. Hand, and X. F. Zhang, 2017, The characteristics that provide independent information about average U.S. monthly stock returns, *The Review of Financial Studies* 30, 4389–4436.

Griliches, Z., 1957, Hybrid corn: An exploration in the economics of technological change, *Econometrica* 25, 501–522.

Gromb, D., and D. Vayanos, 2010, Limits of arbitrage: The state of the theory, *Annual Review of Financial Economics* 2, 251–275.

Grossman, S. J., and J. E. Stiglitz, 1980, On the impossibility of informationally efficient markets, *American Economic Review* 70, 393–408.

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

Gu, S., B. Kelly, and D. Xiu, 2021, Autoencoder asset pricing models, *Journal of Econometrics* 222, 429–450.

Hall, B. H., A. B. Jaffe, and M. Trajtenberg, 2001, The NBER patent citations data file: Lessons, insights and methodological tools, Working Paper 8498, National Bureau of Economic Research.

Harvey, C. R., and Y. Liu, 2021, Lucky factors, *Journal of Financial Economics* 141, 413–435.

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

Hoerl, Arthur E., and Robert W. Kennard, 1970, Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* 12, 55–67.

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

Ioffe, S., and C. Szegedy, 2015, Batch normalization: Accelerating deep network training by reducing internal covariate shift, *arXiv preprint arXiv:1502.03167* .

Jensen, T. I., B. T. Kelly, and L. H. Pedersen, 2023, Is there a replication crisis in finance?, *The Journal of Finance* 78, 2465–2518.

Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, 2017, Lightgbm: A highly efficient gradient boosting decision tree, in *Advances in Neural Information Processing Systems*.

Kelly, B., S. Malamud, and K. Zhou, 2024, The virtue of complexity in return prediction, *The Journal of Finance* 79.

Kelly, B. T., S. Pruitt, and Y. Su, 2019, Characteristics are covariances: A unified model of risk and return, *Journal of Financial Economics* 134, 501–524.

Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* .

Kozak, S., S. Nagel, and S. Santosh, 2020, Shrinking the cross section, *Journal of Financial Economics* 135, 271–292.

Li, S., V. DeMiguel, and A. Martin-Utrera, 2024, Comparing factor models with price-impact costs, *Journal of Financial Economics* 162, 103949.

Martin, I. W. R., and S. Nagel, 2022, Market efficiency in the age of big data, *Journal of Financial Economics* 145, 154–177.

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

Novy-Marx, R., and M. Velikov, 2016, A taxonomy of anomalies and their trading costs, *The Review of Financial Studies* 29, 104–147.

Prokhorenkova, L., G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, 2018, Catboost: Unbiased boosting with categorical features, in *Advances in Neural Information Processing Systems*.

Shleifer, A., and R. W. Vishny, 1997, The limits of arbitrage, *The Journal of Finance* 52, 35–55.

Shumway, T., 1997, The delisting bias in crsp data, *The Journal of Finance* 52, 327–340.

Shumway, T., and V. A. Warther, 1999, The delisting bias in CRSP's Nasdaq data and its implications for the size effect, *The Journal of Finance* 54, 2361–2379.

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

Welch, I., and A. Goyal, 2008, A comprehensive look at the empirical performance of equity premium prediction, *The Review of Financial Studies* 21, 1455–1508.

White, H., 1988, Economic prediction using neural networks: The case of IBM daily stock returns, in *IEEE 1988 International Conference on Neural Networks*, 451–458.

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, 301–320.