Université de Neuchâtel
Faculté des sciences économiques

THREE ESSAYS ON THE APPLICATION OF MACHINE LEARNING IN EQUITY MUTUAL FUNDS

PhD thesis submitted to the Faculty of Economics and Business

Institute of Financial Analysis

University of Neuchâtel

For the PhD degree in Finance

by

Moreno FRIGG

Approved by the dissertation committee:

Prof. Florian WEIGERT, University of Neuchâtel, thesis director

Prof. Carolina SALVA, University of Neuchâtel, jury president

Prof. Jürg FAUSCH, Lucerne University of Applied Sciences and Arts

Prof. Stefan RUENZI, University of Mannheim, Germany

Defended on 07 July 2026

Faculté des sciences économiques
Avenue du 1er-Mars 26
CH-2000 Neuchâtel

www.unine.ch/seco

unine
Université de Neuchâtel
Faculté des sciences économiques

IMPRIMATUR POUR LA THÈSE

Three Essays on the Application of Machine Learning in Equity Mutual Funds

Moreno FRIGG

UNIVERSITÉ DE NEUCHÂTEL
FACULTÉ DES SCIENCES ÉCONOMIQUES

La Faculté des sciences économiques,
sur le rapport des membres du jury

Prof. Florian WEIGERT, Université de Neuchâtel, directeur de thèse
Prof. Carolina SALVA, Université de Neuchâtel
Prof. Stefan RUENZI, University of Mannheim Business School, Germany
Prof. Jürg FAUSCH, Lucerne University of Applied Sciences and Arts (HSLU)

autorise l'impression de la présente thèse.

Neuchâtel, le 8 septembre 2026

La doyenne
Claudia Jonczyk Sédès

Faculté des sciences économiques
Avenue du 1er-Mars 26
CH-2000 Neuchâtel
Email: secretariat.seco@unine.ch
Tel: +41 32 718 15 00
www.unine.ch/seco

Acknowledgments

This dissertation is the culmination of years of work, yet it stands on far more than individual effort. It has been made possible by people who offered guidance when I needed direction, encouragement when I needed confidence, and belief when I needed both.

I am deeply grateful to my PhD supervisor, Florian Weigert, for giving me the opportunity to pursue my doctoral studies under his supervision and for the trust and confidence he placed in me. His expertise, thoughtful feedback, and academic guidance greatly strengthened my research. Both as a supervisor and co-author, his mentorship and steady guidance were greatly appreciated throughout this journey.

My particular gratitude goes to my mentor, co-author, and friend Jürg Fausch. He sparked my curiosity for academic research and played a central role in shaping my development both as a researcher and as a human being. His unwavering support, intellectual guidance, and commitment to our joint work have been invaluable.

A sincere thanks goes to Stefan Ruenzi, my co-author, whose deep knowledge of mutual funds and asset management greatly enriched our collaboration. Our joint paper benefited greatly from his expertise, and it has been truly inspiring to work with him. In addition, I would like to thank Karsten Döhnert for providing support whenever necessary and for the stimulating conversations about finance and life in general.

To my fellow doctoral students, especially Nadine Berchtold for her moral support during the toughest times, and Nikolay Pugachyov for our stimulating discussions on mutual fund research: thank you very much.

I would like to thank Joscha Rufibach and Laurent Matson, my best friends and faithful companions on my life's journey, for their unwavering support, encouragement, and friendship throughout the years. Without you, my life would not be nearly as joyful, meaningful, or filled with laughter as it is today.

Finally, my deepest gratitude goes to my family for their continuous support and encouragement throughout this journey. Their trust and belief in me helped me navigate both the challenges and the rewarding moments along the way. Most importantly, they taught me to follow my dreams and pursue what truly makes me happy, a mindset that has continuously inspired and guided me throughout my life.

Moreno Frigg
Kriens, April 2026

Abstract

The first chapter demonstrates improved out-of-sample predictability of future fund flows using state-of-the-art machine learning methods. Nonlinear machine learning models significantly outperform linear models in terms of out-of-sample R-squared. Using interpretable ML methods, we identify past flows and the Morningstar rating as the most important predictors for net-flows, while other past performance variables are of minor importance. We find that the importance of Morningstar ratings and expenses has increased over time. In addition, the interaction effect of past flows with the Morningstar rating has a substantial impact on future flows. Furthermore, our results demonstrate that machine learning-based fund flow predictions can be used to ex-ante differentiate between high and low-performing mutual funds. Finally, funds whose flow predictions can be improved the most using ML reveal the worst performance, consistent with the idea that liquidity management is particularly challenging for these funds.

The second chapter revisits the recent finding of DeMiguel, Gil-Bazo, Nogales, and Santos (2023) that machine learning (ML) methods can identify long-only portfolios of actively managed mutual funds with positive net annual alpha of up to 2.69%. Using their replication package, we show that these conclusions are driven by a coding error that introduces a forward-looking bias in the construction of portfolio returns. Removing the bias eliminates the long-only alpha. Based on an independent replication, we find that nonlinear ML methods consistently identify underperforming funds. However, this also holds for linear models. Consequently, long–short portfolios deliver robust and economically meaningful alphas, driven predominantly by the short leg. We further extend the analysis by expanding the sample period to the end of 2024, by considering forecast horizons up to 36 months and conducting univariate sorts across fund characteristics. These extensions do not materially enhance long-only predictability but offer empirical evidence of the relative advantages of ML at longer horizons. Overall, our results suggest that ML adds value primarily through the consistent avoidance of poorly performing funds rather than by identifying long-only outperforming ones. This has important implications for both the academic interpretation of ML performance and its practical application in active mutual fund selection.

The third chapter examines how machine learning design choices affect the predictability of equity mutual fund performance. Using 2,052 model specifications across seven design choice categories, I find little evidence of robust long-only alpha.

Only two specifications yield statistically significant outperformance, likely reflecting data snooping rather than genuine predictability. Nevertheless, design choices have economically meaningful effects on portfolio outcomes. Nonlinear methods, such as random forest and gradient boosting, generally outperform linear models, while longer training windows, particularly expanding schemes, yield more stable results. However, these effects are not uniform across models. This indicates that the impact of design choices is inherently model-specific. To quantify sensitivity to methodological decisions, I estimate non-standard errors (NSE). The NSE amounts to 0.83% annualized alpha, and the NSE-to-standard-error ratio of 0.86 indicates that specification uncertainty is comparable in magnitude to sampling variation. Overall, the results highlight the sensitivity of machine learning-based evidence in mutual fund research to design choices.

Keywords: Mutual funds, fund flow predictability, fund performance predictability, active management, replication, non-standard errors, design choices, machine learning, interpretable machine learning, big data.

Résumé

Le premier chapitre met en évidence une amélioration de la capacité prédictive hors échantillon des flux futurs des fonds grâce à l'utilisation de méthodes de machine learning de pointe. Les modèles non linéaires de machine learning surperforment nettement les modèles linéaires en termes de coefficient de détermination hors échantillon (R^2). À l'aide de méthodes de machine learning interprétables, nous identifions les flux passés et la notation Morningstar comme les principaux prédicteurs des flux nets, tandis que les autres variables relatives aux performances passées revêtent une importance moindre. Nous constatons que l'importance des notations Morningstar et des frais a augmenté au fil du temps. En outre, l'effet d'interaction entre les flux passés et la notation Morningstar exerce une influence substantielle sur les flux futurs. Par ailleurs, nos résultats montrent que les prédictions de flux des fonds fondées sur le machine learning peuvent être utilisées ex ante afin de différencier les fonds communs de placement à performance élevée de ceux à faible performance. Enfin, les fonds pour lesquels les prédictions de flux peuvent être le plus fortement améliorées grâce au machine learning affichent les performances les plus faibles, ce qui est cohérent avec l'idée que la gestion de la liquidité est particulièrement difficile pour ces fonds.

Le deuxième chapitre réexamine le résultat récent de DeMiguel, Gil-Bazo, Nogales, and Santos (2023) selon lequel les méthodes de machine learning (ML) permettent d'identifier des portefeuilles long-only de fonds communs de placement gérés activement générant un alpha net annuel positif pouvant atteindre 2,69%. En utilisant leur package de réplication, nous montrons que ces conclusions sont dues à une erreur de programmation qui introduit un biais d'anticipation dans la construction des rendements des portefeuilles. La suppression de ce biais fait disparaître l'alpha des portefeuilles long-only. Sur la base d'une réplication indépendante, nous constatons que les méthodes ML non linéaires permettent d'identifier de manière systématique les fonds sous-performants. Toutefois, ce résultat vaut également pour les modèles linéaires. Par conséquent, les portefeuilles long-short génèrent des alphas robustes et économiquement significatifs, qui sont principalement attribuables à la position vendeuse. Nous étendons en outre l'analyse en prolongeant la période d'échantillonnage jusqu'à la fin de 2024, en considérant des horizons de prévision allant jusqu'à 36 mois et en réalisant des tris univariés selon les caractéristiques des fonds. Ces extensions n'améliorent pas sensiblement la capacité de prédiction pour les portefeuilles long-only, mais fournissent des éléments empiriques en faveur des

avantages relatifs des méthodes ML à des horizons plus longs. Dans l'ensemble, nos résultats suggèrent que les méthodes ML créent principalement de la valeur en permettant d'éviter de manière systématique les fonds sous-performants, plutôt qu'en identifiant des fonds long-only surperformants. Ces résultats ont des implications importantes tant pour l'interprétation académique de la performance des méthodes ML que pour leur application pratique à la sélection active de fonds communs de placement.

Le troisième chapitre examine comment les choix de conception des modèles de machine learning influencent la capacité à prédire la performance des fonds communs de placement en actions. En considérant 2 052 spécifications de modèles réparties sur sept catégories de choix de conception, je trouve peu d'éléments en faveur d'un alpha robuste pour les portefeuilles long-only. Seules deux spécifications produisent une surperformance statistiquement significative, ce qui reflète vraisemblablement un phénomène de data snooping plutôt qu'une véritable capacité de prédiction. Néanmoins, les choix de conception ont des effets économiquement significatifs sur les résultats des portefeuilles. Les méthodes non linéaires, telles que les forêts aléatoires et le gradient boosting, surpassent généralement les modèles linéaires, tandis que les fenêtres d'apprentissage plus longues, en particulier les schémas à fenêtre d'apprentissage croissante, produisent des résultats plus stables. Ces effets ne sont toutefois pas uniformes entre les différents modèles. Cela indique que l'impact des choix de conception est intrinsèquement spécifique au modèle considéré. Afin de quantifier la sensibilité aux choix méthodologiques, j'estime des erreurs standards non conventionnelles (non-standard errors, NSE). La NSE s'élève à 0,83% d'alpha annualisé, tandis que le ratio entre la NSE et l'erreur standard, égal à 0,86, indique que l'incertitude liée à la spécification est d'une ampleur comparable à la variabilité d'échantillonnage. Dans l'ensemble, les résultats mettent en évidence la sensibilité des conclusions fondées sur le machine learning dans la recherche sur les fonds communs de placement aux choix de conception des modèles.

Contents

I Machine Learning Mutual Fund Flows **1**

1 Introduction 1

2 Data 6

2.1 Sample data 6

2.2 Mutual fund characteristics 8

2.3 Macroeconomic information 11

2.4 Data preprocessing 13

3 Prediction framework and performance evaluation 14

3.1 Linear regression - baseline model 14

3.2 Machine learning algorithms 15

3.3 Cross-validation, hyperparameter tuning and performance evaluation 17

4 Empirical analysis 19

4.1 Out-of-sample predictability comparison 20

4.2 Characteristic importance and interactions 24

4.3 Predicted fund flows and performance 30

4.4 Fund flow prediction accuracy and performance 32

5 Conclusion 34

I.A Process of matching CRSP with Morningstar 59

I.B Machine learning methods 61

2.1 Elastic net 61

2.2 Decision tree 61

2.3 Random forest 62

2.4 Gradient boosting 63

2.5 Feedforward neural network 64

I.C Additional robustness check 66

I.D SHAP interaction values 67

I.E Missing data 69

I.F REFORMS checklist for ML-based science 70

II Does machine learning really help to select mutual funds with positive alpha? **73**

1 Introduction 73

2 Reproduction of DeMiguel et al. (2023) . . . 75
3 Replication of DeMiguel et al. (2023) . . . 79
3.1 Replication design, methodology and data . . . 79
3.2 Empirical results . . . 83
4 Extensions and additional insights . . . 91
4.1 Expansion of sample period until 2024 . . . 91
4.2 Forecasting and investment horizon . . . 91
4.3 Univariate sorts . . . 93
5 Conclusion . . . 94
II.A Sample construction . . . 109
II.B Reproduction and Replication using alternative factor models . . . 113
II.C Replication based on intentionally retaining the survivorship bias present in DeMiguel et al. (2023) . . . 116
II.D Replication of table 4 to 7 in DeMiguel et al. (2023) . . . 117

III Machine Learning Design Choices and Equity Mutual Fund Performance **121**
1 Introduction . . . 121
2 Data, fund characteristics and prediction framework . . . 124
2.1 Sample data . . . 125
2.2 Mutual fund characteristics . . . 125
2.3 Data preprocessing and prediction framework . . . 126
2.4 Design choices, performance implications and non-standard errors . . . 127
3 Machine learning design choices . . . 129
3.1 Algorithm (N=4) . . . 130
3.2 Imputation method (N=3) . . . 130
3.3 Training window (N=4) . . . 131
3.4 Training frequency (N=3) . . . 132
3.5 Hyperparameter search strategy (N=4) . . . 132
3.6 Hyperparameter tuning procedure (N=2) . . . 133
3.7 Hyperparameter frequency adjustment (N=3) . . . 134
4 Empirical analysis . . . 134
4.1 Design choice impact on portfolio alpha . . . 134
4.2 Non-standard errors . . . 137
5 Conclusion . . . 139

List of Tables

I.1 Number of share classes per year with monthly average, median and quantile flows . . . 36
I.2 Predictor variables by category . . . 37
I.3 Summary statistics of predictor variables . . . 38
I.4 Out-of-sample OLS model comparison based on R^2_{OOS} . . . 40
I.5 Out-of-sample machine learning model comparison based on R^2_{OOS} . . . 41
I.6 Out-of-sample machine learning model comparison based on R^2_{OOS} using a 12-, 36-, and 60-month rolling-window on the training data . . . 42
I.7 Forecast comparison based on the modified DMW statistic 43
I.8 Out-of-sample model comparison based on R^2_{OOS} in different market conditions . . . 44
I.9 Mean of absolute SHAP values for random forest . . . 45
I.10 Mean of absolute SHAP values for random forest - Top and bottom decile of the predicted fund flow distribution and for institutional and retail share classes . . . 46
I.11 Mean absolute SHAP interaction values for random forest and gradient boosting . . . 51
I.12 Monthly out-of-sample alpha (in %) of the top and bottom decile (1^{st} percentile) portfolios . . . 54
I.13 Monthly out-of-sample alpha (in %) based on prediction accuracy sorted quintile portfolios . . . 56
I.14 Fund flow prediction accuracy and performance . . . 57
I.A1 Detailed matching procedure between CRSP and Morningstar share classes . . . 59
I.C1 Out-of-sample machine learning model comparison based on $R^2_{OOS,Null}$ . . . 66
I.D1 Robustness analysis of SHAP interaction values based on subsets of varying size for random forest . . . 68
I.E1 Percentage of missing values per predictor variable . . . 69
I.F1 REFORMS Checklist for ML-based Science . . . 70

II.1 Monthly out-of-sample alpha (in %) - Reproduction . . . 96

II.2 Share class characteristics: Descriptive statistics 98
II.3 Monthly out-of-sample alpha (in %) - Replication 100
II.4 Monthly out-of-sample alpha (in %) - Extended data set through 2024 104
II.5 Monthly out-of-sample alpha (in %) for a forecasting horizon of 36 months 105
II.6 Monthly out-of-sample alpha (in %) of univariate sorts - 12 months 106
II.7 Monthly out-of-sample alpha (in %) of univariate sorts - 36 months 107
II.A1 Sample selection 111
II.B1 Monthly out-of-sample alpha (in %) - Reproduction 114
II.B2 Monthly out-of-sample alpha (in %) - Replication 115
II.C1 Monthly out-of-sample alpha (in %) - Replication with survivorship bias 116
II.D1 Out-of-sample alpha with respect to OLS 117
II.D2 Out-of-sample mean excess return and risk 118
II.D3 Out-of-sample alpha of double-sorted portfolios 119
II.D4 Out-of-sample alpha of fund portfolios under different market conditions 120

III.1 Share class characteristics: Descriptive statistics 141
III.2 Hyperparameter search spaces 143
III.3 Monthly out-of-sample alpha (in %) of best and worst configuration per algorithm 144
III.4 NSE-to-SE ratios across research design choices 147

List of Figures

I.1 Correlation matrix between fund characteristics 39
I.2 Evolution of characteristic importance for random forest over time 47
I.3 Directional impact of the characteristics for random forest 48
I.4 Directional impact of the characteristics for random forest over time 49
I.5 Functional form of past fund flows 50
I.6 Nonlinear impact of the Morningstar rating 52
I.7 Interaction effect of past fund flows and Morningstar rating 53

II.1 Cumulative portfolio alpha of DeMiguel et al. (2023) and the reproduction 97
II.2 Correlation matrix between the target variable and fund characteristics 99
II.3 Cumulative portfolio alpha of the replication 101
II.4 Characteristic importance 102
II.5 Capital misallocation and machine learning, DeMiguel et al. (2023) vs. Replication 103

III.1 Design choices 142
III.2 Distribution of annualized alphas across alternative design choices 145
III.3 Distribution of alphas across alternative design choices by algorithm 146

Chapter I

Machine Learning Mutual Fund Flows

(with Jürg Fausch, Stefan Ruenzi and Florian Weigert)

1. Introduction

Open-end mutual funds offer liquidity to investors by allowing them to redeem fund shares at their daily net asset value. This feature can create problems for fund managers if flows are unpredictable and volatile, as it makes the liquidity management of the fund difficult. Prior literature has shown that large unexpected flows can negatively influence fund performance (Edelen, 1999) and eventually even lead to distorted asset prices (Coval and Stafford, 2007). Thus, it is of considerable importance to precisely predict future flows.

In this paper, we study the investment decisions of mutual fund investors, reflected in fund flows, and reveal new evidence about the predictability of fund flows based on mutual fund characteristics and other variables. According to the literature, numerous variables impact flows, and these relationships are often nonlinear (Sirri and Tufano, 1998; Chevalier and Ellison, 1997). This setting creates an ideal scenario for applying machine learning (ML) methods. These models are not only able to capture the impact of nonlinearities and interactions between a large set of fund characteristics but also mitigate the risk of in-sample model overfitting when meaningfully calibrated. In the same fashion as Gu, Kelly, and Xiu (2020), DeMiguel et al. (2023), Kaniel et al. (2023), Bali et al. (2023) and Fausch et al. (2025a) in the context of return predictions, we apply a linear OLS regression (baseline model) and ML methods to obtain improved fund flow predictions. Our analysis using state-of-the-art ML techniques also allows us to identify nonlinear and hitherto unknown interaction effects that have an impact on predicted flows, thus also contributing to a better understanding of mutual fund investors' decision making. Finally, building on the hypothesis that flow pressure can lead to return effects on the underlying stocks a fund holds and eventually to performance persistence, we aim to differentiate between high and low performing mutual funds (out-of-sample) based on machine learning implied monthly fund flow predictions.

We start our empirical analyses by comparing the predictive power of simple OLS regressions and different ML models for future equity fund flows in the US over the period from January 2000 to December 2023. To this end, we use the out-of-sample R^2 statistic to benchmark our predictions against a fund's simple historical mean flows. To assess whether some models deliver significantly better predictions than other models we conduct pairwise comparisons of the predictive accuracy of different forecasting models by utilizing the model-free modified Diebold and Mariano (1995) and West (1996) test statistic.

For our empirical exercise, we use a semi-structured systematic approach to first identify 55 variables that might have an influence on flows based on the prior literature. These variables are mainly linked to past performance and other fund characteristics. In addition, we use nine variables that proxy for aggregate financial market and macroeconomic conditions that might have an influence on aggregate flows into equity funds. To provide a fair comparison between OLS and ML methods, the specification of the OLS model is guided by the literature on mutual fund flows and accounts for the well-known nonlinear impact of past performance on fund flows (Sirri and Tufano, 1998; Chevalier and Ellison, 1997). We find that linear models like OLS and elastic net achieve a positive out-of-sample R^2 of about 17% to 18% percent for the full sample using a historical mean flow benchmark. Nonlinear ML-models like random forest and gradient boosting improve this measure by about 4 percentage points, i.e., by more than a fifth, while neural networks also lead to an improvement, albeit a smaller one. These results remain robust across different economic conditions and market environments, indicating that the predictive accuracy of the forecasting models holds steady even during economic downturns, which are often marked by increased outflows from risky asset classes (e.g., Jank, 2012; Pástor and Vorsatz, 2020). Among all statistical models, the forecasts obtained by random forest are the most precise and significantly better (at the 1% level) than the predictions obtained from any other forecasting method based on pairwise forecast accuracy tests using modified Diebold and Mariano (1995) and West (1996) tests.

While early ML prediction approaches were often criticized as black boxes with hard-to-interpret-results, significant progress in recent years now allows for meaningful model interpretation. We build upon these results to quantify the relative importance of fund characteristics and their interactions for the prediction of fund flows. For that purpose, we estimate model-agnostic SHAP values (Lundberg and Lee, 2017; Lundberg et al., 2020) to determine the contribution of each predictor

to the respective fund flow forecasts.[1] Based on the best performing model, random forest, our results indicate that past flows over the last month are the most influential variable predicting flows, followed by the average flows over the past six and 12 months, the Morningstar rating, and total net assets (TNA). The importance of past flows is consistent with stable fund characteristics being important flow drivers.[2] We also document that various measures of past performance (e.g., alpha, value added, market-adjusted returns) are less important in predicting future fund flows, which is consistent with Ben-David et al. (2022) who show that fund returns and more sophisticated performance measures have little impact on flows once Morningstar ratings are included in OLS regressions. Interestingly, this is the case for both retail and institutional investor fund share classes. While most other variables are of similar importance for both types of share classes, expense ratios are more important for institutional share class flow predictions.

Looking at the strength of the impact of these predictors over time, we find that flow-related predictors are consistently ranked as the most important predictors over the whole sample period, while the Morningstar rating is not as important in the earliest years of the sample as it becomes in later periods. This pattern is consistent with the increased popularity of these ratings over time. While most other variables do not show a clear time trend, we do find that expense ratios have become more influential in predicting fund flows. Our results also indicate that in the bottom decile of the predicted flow distribution costs seem to play a more important role, i.e., outflows are more cost-sensitive than inflows.

SHAP values also allow us to determine the directional impact of each variable. We find that our most important predictor variable, past monthly flows, usually affects future flow predictions positively, but in some cases also negatively. This finding suggests a nonlinear relationship on fund flow forecasts. In contrast, the directional impact of average flows over the past six and 12 months is typically positively related to predicted flows. The Morningstar rating usually has a positive impact on flow predictions. All of these patterns are quite stable over time. Furthermore, while expense ratios were of marginal relevance at the beginning of the sample period with no clear directional impact, they became clearly negatively related to

[1] SHAP values break down a model's prediction for a single data point, showing how much each variable contributes to the final outcome by comparing it to an average baseline prediction. Essentially, they explain which variables affect the prediction positively or negatively and by how much relative to the baseline forecast, hence providing an interpretable "fair share" of each variable's impact.

[2] In traditional OLS-based mutual fund flow studies stable fund characteristics are often captured by including fund fixed effects in flow regressions. However, in our predictive machine learning framework, it is unclear how to account for such fixed-effects equivalently.

future flow predictions from 2006 onward. This finding suggests an increased cost-awareness of fund investors, potentially driven by the competitive pressure from low-cost index funds (Cremers et al., 2016).

Motivated by the nonlinear impact of past performance on flows documented in earlier studies (see, e.g., Sirri and Tufano, 1998), we also assess the functional form of the relationship between the Morningstar rating and predicted flows. We document that both, past flows and the Morningstar rating have a convex impact on predicted flows.

SHAP values can also be used to analyze the importance of the impact of interactions between variables.[3] We analyze the importance of all possible interactions between our predictor variables and find the interaction of past flows with the Morningstar rating to have the strongest impact on flow forecasts among all non-entirely flow-related interactions.[4] Analyzing the functional form for the impact of this relationship, we document that high past inflows and a high Morningstar rating imply a positive impact on predicted flows, while high past inflows and a low Morningstar rating are associated with a negative impact, suggesting that new money (proxied for by recent past flows) is more attentive to performance information contained in ratings.

In the last step of our analysis, we investigate whether the predictability of fund flows through ML models can help in differentiating high- from low-performing funds out-of-sample and find this to be the case: Long-short decile portfolios based on predicted flows from the ML models with the highest forecasting accuracy, random forest and gradient boosting, generate an alpha of 2.40% and 2.52% per year, respectively, which is statistically significant at the 5% and 1% levels. This outperformance is very similar to the results obtained by Kaniel et al. (2023), that are based on computationally more intense feedforward neural networks and a richer information set to directly predict the fund alpha distribution rather than flows. Consistent with our findings, they identify fund flows as one of the most important predictors of future risk-adjusted fund performance. The predictive power of flow predictions for future performance is consistent with a smart money effect (see, e.g., Gruber, 1996; Zheng, 1999).

We further analyze the performance implications of the higher forecasting accuracy offered by ML models relative to more basic flow prediction methods. We find

[3]SHAP interaction values show how pairs of variables jointly impact a model's prediction. This approach not only reveals how each variable influences the prediction on its own but also how the combination of two variables shifts the outcome differently than if they acted independently.

[4]The strongest interactions are found between lagged monthly flows and lagged 6-month and lagged 12-month flows, respectively.

that funds whose flow forecasts based on the simple historical mean are particularly bad relative to those from random forest also exhibit worse overall performance. This effect may stem from the fact that, during our sample period, these funds likely lacked access to modern ML-based flow forecasting methods, making it especially challenging for them to predict flows accurately and manage liquidity effectively. Controlling for other variables known to affect future performance, our results show potential performance differences ranging from 0.31% to 0.48% per year.

Overall, our paper contributes to three streams of the literature. First, we contribute to the broad literature on the flow determinants of mutual funds. Since the early work of Chevalier and Ellison (1997) and Sirri and Tufano (1998), many papers have analyzed the impact of various performance measures, fund and fund manager characteristics on flows. We extend this literature by providing evidence that combining information from many individual fund characteristics, allowing for nonlinearities and by uncovering new evidence on the impact of hithertho unnoticed interaction effects, helps to significantly improve mutual fund flow predictions. Using interpretable ML methods, we are also able to contribute to a better understanding of the intricate impact of the interplay between past ratings and past flows on flow predictions. Second, we also contribute to a growing literature of applying ML techniques in finance.[5] So far, the majority of papers apply ML models to predict stock returns (see, among others, Gu, Kelly, and Xiu, 2020; Chen, Pelger, and Zhu, 2024; Leippold, Wang, and Zhou, 2022; Neuhierl et al., 2026). Bianchi, Büchner, and Tamoni (2021) and Bali et al. (2022) forecast bond returns, while the predictability of delta-hedged option returns is examined by Bali et al. (2023). Medeiros et al. (2021) and Hauzenberger, Huber, and Klieber (2023) use ML techniques to forecast inflation and Goulet Coulombe et al. (2022) provide evidence that nonlinearities in features, exploited by ML models, help to predict macroeconomic variables. Finally, in the mutual fund literature, ML techniques have mainly been applied to predict fund performance (Li and Rossi, 2021; DeMiguel et al., 2023; Kaniel et al., 2023; Fausch et al., 2025a). We complement this literature by showing how ML techniques can be used to improve flow forecasts. Finally, we contribute to the literature on the impact of flows and flow volatility on future performance. In an early contribution, Edelen (1999) shows that large unexpected flows can negatively affect fund performance. In more recent work, Kim (2020) and Dou, Kogan, and Wu (2024) argue that fund managers have an incentive to hedge their flow risk by underweighting stocks with returns that covary highly with aggregate fund flows (fund flow beta). Aragon and Kim (2023) demonstrate that

[5]For a review on this topic, see Giglio, Kelly, and Xiu (2022) as well as Kelly and Xiu (2023).

stocks are at higher risk of fire sales when held by mutual funds that experience outflows during periods of aggregate industry redemptions. We contribute to this literature by providing evidence consistent with the idea that funds whose flows are relatively difficult to predict using simple historical means as compared to the best ML methods have a greater potential for performance improvement. We also make a minor methodological contribution by implementing the recently proposed reliability protocol for ML applications (Kapoor et al., 2024), addressing concerns about validity and reproducibility, for the first time in a financial research context. The remainder of the paper is structured as follows. Section 2 describes the data used in the empirical analysis. In Section 3, we provide information about the statistical methods used for prediction and discuss the performance evaluation metrics to evaluate our forecasts. Section 4 presents the main empirical results of the paper. We look at the out-of-sample predictive ability and discuss which predictors matter most. We also analyze the predictive ability of fund flow forecasts to differentiate ex-ante between high and low performing mutual funds. Finally, we conclude in Section 5.

2. Data

Our primary data sources are the Center for Research in Security Prices (CRSP) Survivor-Bias-Free US Mutual Fund database and MorningstarDirect which virtually cover the full population of U.S. mutual funds.

2.1. Sample data

Many funds have different share classes with different management fees, expense ratios or loads (Khorana, Servaes, and Tufano, 2008). To account for this economically relevant heterogeneity, our analyses are performed at the share class level. Our sample includes all institutional and retail share classes that focus on the US equity market. Because data coverage on monthly fund TNAs prior to 1991 is scarce and incomplete (see, e.g., Dou, Kogan, and Wu, 2024) our sample spans from January 1991 to December 2023. We restrict our analyses to share classes of actively managed funds, excluding ETFs and passive funds. More specifically, for all share classes available before 2003 we follow the approach suggested by Gil-Bazo and Ruiz-Verdú (2009) and exclude all share classes that contain specific keywords indicating passively managed funds. Beginning in 2003, we use the corresponding identifier provided by CRSP to select these share classes. Second, we require that

each included share class invests more than 70 percent of their total net assets (TNA) into equities. Third, to avoid incubation bias, we only consider share classes reaching at least 36 monthly observations since inception and exceed a threshold of 15 million in TNA after this incubation period, which is a commonly used cutoff in the literature (see, among others, Elton, Gruber, and Blake, 2001; Evans, 2010; Doshi, Elkamhi, and Simutin, 2015; Kaniel et al., 2023).

To extend the set of characteristics we rely on fund and fund manager data provided by Morningstar. According to Massa, Reuter, and Zitzewitz (2010), Morningstar is a more important source of information for investors than CRSP and provides us with the Morningstar rating which seems to be an important flow determinant (Del Guercio and Tkac, 2008; Khorana and Servaes, 2012; Reuter and Zitzewitz, 2021; Ben-David et al., 2022). To match the share classes between these two databases, we use the fund's nine-digit CUSIP (Hillert, Niessen-Ruenzi, and Ruenzi, 2025) as a common unique identifier. To avoid well-known inconsistencies between the two databases (Elton, Gruber, and Blake, 2001; Berk and van Binsbergen, 2015) we perform various robustness checks. Our final data set contains 13,376 unique share classes. Based on this sample Table I.1 reports the number of share classes for each year and the corresponding flow distribution described by its mean, standard deviation, lower (Q1) and upper quantile (Q3). It is notable that the average (median) fund recorded outflows of -0.47% (-0.65%)[6] throughout the sample period. More specifically, every year since 2004 has seen negative fund flows indicating that the market for actively managed equity mutual funds in the U.S. has been shrinking over the last two decades, while it was characterized by strong external growth during the previous years (Sirri and Tufano, 1998).

[Insert Table I.1 here]

To be comprehensive in objectively selecting the most important determinants of fund flows, we follow a systematic semi-structured approach. We search for studies on mutual fund flows in the top finance and related journals. We identify potential predictor variables from these studies and focus on variables that can be computed based on readily available information on fund characteristics and historical returns and are thus relatively easy to calculate using data provided by CRSP and Morningstar. We thus exclude holding-based characteristics. Based on this approach, we identify 55 mutual fund characteristics. In addition, we use nine variables that proxy for macroeconomic conditions. In total, we construct a dataset

[6]Note that these numbers differ slightly from the descriptive statistics reported in Table I.3, which are based on the training sample.

of 64 predictors, shown in Table I.2, to be used in forecasting fund flows using machine learning techniques.

[Insert Table I.2 here]

2.2. Mutual fund characteristics

Our main variable of interest is monthly fund flows determined as

$$flow_{i,t} = \frac{TNA_{i,t} - TNA_{i,t-1}\,(1 + R_{i,t})}{TNA_{i,t-1}}, \tag{I.1}$$

where $R_{i,t}$ denotes the monthly net return of share class i. To account for possible outliers caused by fund mergers and splits, and as common in the literature, monthly flows are cross-sectionally winsorized at the 0.01 and 0.99 percentiles (see, among others, Huang, Wei, and Yan, 2022; Hillert, Niessen-Ruenzi, and Ruenzi, 2025). As in Cen et al. (2024), we use the natural logarithm of one plus the monthly flows in our computations since flows are the relative growth in the share class TNA adjusted for net returns.

Using data provided by CRSP, for each share class i in month t we obtain the total net assets ($TNA_{i,t}$), expense ratio ($ER_{i,t}$, Barber, Huang, and Odean (2016)), the turnover ratio (Ivković and Weisbenner, 2009) as well as front-end and back-end loads (Ivković and Weisbenner, 2009; Chen, Goldstein, and Jiang, 2010).[7] The return volatility (Franzoni and Schmalz, 2017) for each share class, using returns net of fees, is calculated based on a rolling window approach with the requirement of at least 30 months of non-missing return observations over the last 36 months. In addition, we compute the $Age_{i,t}$ of each share class (Chevalier and Ellison, 1997), which is the rounded number of months between its inception date and the last calendar date of month t. Following Barber, Huang, and Odean (2016), we estimate the market-adjusted return ($MAR_{i,t}$) by subtracting the market return from the fund return.

Several characteristics used in our analyses are based on time-series regressions of share-class excess returns on well-known systematic risk factors (Fama and French, 1993; Carhart, 1997; Fama and French, 2015).[8] While there is still a debate about which asset pricing model is used for risk adjustment by mutual fund investors

[7] If no information about front-end or back-end loads is provided by CRSP we assume that no such loads are charged and set the corresponding variable to 0.

[8] Excess returns ($R^e_{i,t}$) are computed by subtracting the one-month T-bill yield which serves as a proxy for the risk-free rate. The monthly risk-free rate data are obtained from Ken French's data library.

(Berk and van Binsbergen, 2016; Barber, Huang, and Odean, 2016; Jegadeesh and Mangipudi, 2021), as Barber, Huang, and Odean (2016), we rely on five different factor models: the capital asset pricing model (CAPM), the Fama and French (1993) three-factor model (FF3), the Carhart (1997) four-factor model (FF4), the Fama and French (2015) five-factor model (FF5) and the Fama and French (2015) five-factor model plus momentum (FF6). The time-series regression, estimated over a 36-month rolling estimation window, for the FF6 factor model is

$$\begin{aligned} R^e_{i,t} = \alpha_{i,t} + b_{i,t}(R_{M,t} - R_{f,t}) + s_{i,t}SMB_t + h_{i,t}HML_t \\ + r_{i,t}RMW_t + c_{i,t}CMA_t + m_{i,t}MoM_t + \varepsilon_{i,t}, \end{aligned} \tag{I.2}$$

where $R^e_{i,t}$ is the excess return on share class i for month t, $R_{f,t}$ is the risk-free rate, $R_{M,t}$ is the market return, SMB_t and HML_t are the size and value-growth returns, MoM_t is the momentum return, $\alpha_{i,t}$ is the return left unexplained by the asset pricing model and $\varepsilon_{i,t}$ is the regression residual.[9] The other applied factor models are nested in Equation I.2 and are the CAPM in which (dropping time subscript) $R_M - R_f$ is the only factor, the Fama and French (1993) three-factor model (FF3) that adds SMB and HML, the Carhart (1997) four-factor model (FF4) that considers MoM as an additional explanatory variable, the Fama and French (2015) five-factor extension (FF5) that adds RMW and CMA to the three-factor model, and the six-factor model (FF6) that adds MoM to the five-factor model. We compute the monthly realized alpha, $\alpha^r_{i,t}$, of share class i in month t for the six-factor model in Equation I.2 as

$$\begin{aligned} \alpha^r_{i,t} = R^e_{i,t} - \hat{b}_{i,t}(R_{M,t} - R_{f,t}) - \hat{s}_{i,t}SMB_t - \hat{h}_{i,t}HML_t \\ - \hat{r}_{i,t}RMW_t - \hat{c}_{i,t}CMA_t - \hat{m}_{i,t}MoM_t, \end{aligned} \tag{I.3}$$

where $\hat{b}_{i,t}$, $\hat{s}_{i,t}$, $\hat{h}_{i,t}$, $\hat{r}_{i,t}$, $\hat{c}_{i,t}$ and $\hat{m}_{i,t}$ are the factor loadings of the ith share class excess return with respect to the six-factor model estimated using a rolling-window regression from $t-36$ to $t-1$. Monthly realized alphas for the other factor models are calculated following the same methodology. As in Barber, Huang, and Odean (2016) we account for the impact of past performance on fund flows (flow-performance sensitivity)[10] by including lagged alphas of the five different factor models introduced above. While Barber, Huang, and Odean (2016) use an exponential decay model

[9] Note that our computations are based on log-returns and we require at least 30 months of non-missing return observations over the last 36 months.

[10] See, among others, Ippolito (1992); Chevalier and Ellison (1997); Sirri and Tufano (1998); Del Guercio and Tkac (2002); Huang, Wei, and Yan (2007); Frazzini and Lamont (2008); Berk and van Binsbergen (2016); Barber, Huang, and Odean (2016); Goldstein, Jiang, and Ng (2017); Roussanov, Ruan, and Wei (2020).

to weight the sum of the realized alphas over the prior 18 months, we instead use a simple average of the realized alphas over the past three, six, twelve and 18 months. For market-adjusted returns, the same procedure is applied. Our decisions are primarily driven by the predictive framework using machine learning and avoiding any form of forward-looking bias by estimating the decay factor using data not available at the time of the forecast. Furthermore, our approach allows for more flexibility in modeling the impact of performance over different horizons. In addition to the monthly realized factor model alphas we also consider the factor loadings of the FF6 model as predictive variables (Barber, Huang, and Odean, 2016).[11] Furthermore, as in DeMiguel et al. (2023) we use valued added and the adjusted $adj.\ R^2_{i,t}$ as additional performance-related predictors. More specifically, following Amihud and Goyenko (2013), for each forecasting horizon, we use the adjusted $adj.\ R^2_{i,t}$ from the FF6 factor model rolling-window regression as a predictor of future fund flows. The $adj.\ R^2_{i,t}$ is a measure of activeness and a lower $adj.\ R^2_{i,t}$ means that the share class tracks the benchmark less closely. Value added as suggested by Berk and van Binsbergen (2015) is computed as

$$va_{i,t} = (\alpha_{i,t} + ER_{i,t}/12) \cdot (TNA_{i,t-1}), \tag{I.4}$$

where we use the realized $\alpha_{i,t}$ defined in Equation I.3 and $ER_{i,t}$ denotes the annual expense ratio of share class i.[12]

Based on information provided by Morningstar we define a dummy variable, management team, which is set to 1 if the share class is team-managed, and 0 otherwise, to consider the impact of the organizational structure on future fund flows (Dass, Nanda, and Wang, 2013). We also add the Morningstar rating (Del Guercio and Tkac, 2008; Khorana and Servaes, 2012; Reuter and Zitzewitz, 2021; Evans and Sun, 2021) to the set of fund characteristics used to predict mutual fund flows. The Morningstar rating assigns one (worst) to five (best) stars, with fixed proportions (10%, 22.5%, 35%, 22.5%, and 10%), based on a mutual fund's historical risk and load-adjusted returns versus category peers.[13] Moreover, we consider variables that

[11] Barber, Huang, and Odean (2016) compute the fund's return related to each risk factor using the exponential decay model analogous to the weighting of alphas explained above. In contrast, we directly use the factor loadings of the FF6 factor model based on rolling-window regressions from $t-36$ to t.

[12] This version of value added follows DeMiguel et al. (2023) and differs from Berk and van Binsbergen (2015) which estimate before-fee alpha using passively managed index funds as benchmarks.

[13] Morningstar launched its mutual fund rating system in 1985. In June 2002, Morningstar revised its rankings significantly. From its inception until the revision, Morningstar ranked all funds against each other based on their returns, while in June 2002 the new rating methodology was that funds were ranked within style categories rather than against the entire fund universe.

account for fund flow persistence (Dou, Kogan, and Wu, 2024). These predictors include past flows over the previous month, past mean flows over a six and 12 months horizon, respectively, and one year lagged monthly fund flows, i.e., flows in the same calendar month in the previous year to account for fund-specific seasonalities in fund flows (Kamstra et al., 2017). Additionally, we include an integer between 1 and 12 indicating the calendar month as a characteristic to capture aggregate seasonalities. Moreover, we compute the volatility of flows as the standard deviation of monthly flows using a rolling window approach with the requirement of at least 30 months of non-missing return observations over the last 36 months. Including lagged flows also captures the impact of non- or slowly varying fund characteristics and is thus to be viewed as analogous to adding fund fixed effects in regression frameworks typically used in fund flow studies (which would be inconsistent with the predictive nature of our ML approach). Finally, we use monthly changes in the Morningstar rating (Del Guercio and Tkac, 2008) as an additional predictor.

2.3. Macroeconomic information

The basic econometric model in fund flow studies (see, e.g., Nanda, Wang, and Zheng, 2004; Barber, Huang, and Odean, 2016; Ben-David et al., 2022) are panel regressions controlling for time-fixed effects. However, how to account for the equivalent of time-fixed effects in a predictive machine learning framework is unclear. Instead, we include various measures of financial markets and macroeconomic uncertainty that may have an impact on aggregate flows in the next period. Ferson and Kim (2012), and in a more recent study, Dou, Kogan, and Wu (2024) show that fund flows share a significant degree of common time-series variation with such factors. Guided by these studies we include the following variables: short-term interest rates (3-month T-bill rates), long-term government bond yields, the term spread as the difference between the long term yield on government bonds and the treasury-bill, the credit spread as the difference between long-term corporate bond and long-term government bond yields, stock market returns and stock market volatility based on the CRSP value-weighted index composed of NYSE, Amex and Nasdaq stocks as well as the volatility index VIX (Ben-Rephael, Kandel, and Wohl, 2012). The data are obtained from Amit Goyal's webpage, Ken French's data library and the Chicago Board Options Exchange. To take macroeconomic uncertainty into account, we include the risk aversion and uncertainty (annual volatility

Ben-David et al. (2022) show that this change in rating methodology did not affect the relationship between flows and ratings and that investors continued to rely on the star rating despite the change in its economic meaning.

percentage) indices of Bekaert, Engstrom, and Xu (2022) as additional predictors.[14]

In Table I.3 we provide the descriptive statistics for our final sample (training sample) used in the empirical analysis. Fund-related variables are measured at the share class level. During our sample period, the average share class has a negative monthly flow (-0.52%) with a standard deviation of 5.63% and an interquartile range of 2.44%, which indicates that there is considerable cross-sectional variation in fund flows. These flow-related summary statistics are similar to the previous literature (see, e.g., Barber, Huang, and Odean, 2016; Ben-David et al., 2022) except that the standard deviation in our sample period (January 1991 to December 2023) is larger. Consistent with the mutual-fund literature the average realized alphas of all factor models in our sample are negative and returns load positively on the market and size factor. The average adjusted R^2 of the FF6 factor model is 0.89, suggesting that it explains a substantial part of the time-series variation in equity mutual fund returns. At the fund level, the average TNA is USD 3.59 billion, while the median TNA is considerably smaller at 695 million. The average annual expense ratio is 1.24% and the average front-end (back-end) loads are 1.05% (0.64%), respectively, with a standard deviation of 2.14% (1.28%). As discussed above, if no information about front-end or back-end loads is provided in CRSP we impute this variable with 0. As a result, the median share class does not charge any loads. The average age is 168 months (14 years). The average Morningstar rating is 3.06, while the first quartile (Q1) corresponds to a Morningstar rating of 2.0 and the third quartile (Q3) is equivalent to a rating of 4.0. The vast majority of changes in the Morningstar rating are upgrades or downgrades of one star (Del Guercio and Tkac, 2008). In our sample, the mean and median change in the Morningstar rating is 0 with a standard deviation of 0.37. Overall, our sample descriptive statistics are consistent with other studies in the mutual fund literature (e.g., Barber, Huang, and Odean, 2016; Ben-David et al., 2022; DeMiguel et al., 2023).

[Insert Table I.3 here]

Figure I.1 shows the correlation coefficients of predictor variables that exceed a threshold of $|0.6|$ and are thus considered substantially correlated.[15] High correlations (> 0.9) are observed between various factor model alphas as well as between the VIX and the risk aversion index. The target variable, fund flows ($t + 1$), has correlations with the lagged predictors that are all below the 0.6 threshold. The largest correlations with monthly flows (our target variable) are observed with mean

[14] These data are provided by Nancy Xu on https://www.nancyxu.net/risk-aversion-index.

[15] We do not show the full correlation matrix due to size and readability restrictions.

flows over the past 6 (0.31) and 12 months (0.31), lagged monthly flows (0.20) and the Morningstar rating (0.17). All other correlations are below 0.15.

[Insert Figure I.1 here]

2.4. Data preprocessing

Before we can apply machine learning methods for predicting future mutual fund flows, we need to preprocess the data to generate the target and predictor variables which requires us to make several implementation decisions. The target variable is defined as winsorized monthly mutual fund flows, $flow_{i,t+h}$ for a forecasting horizon of $h = 1$ month. The predictor variables correspond to the mutual fund characteristics outlined in Subsection 2.2 and the macroeconomic information in Subsection 2.3. For our analyses, we use non-imputed data and thus avoid relying on a specific imputation method, which implies omitting any observation for which at least one characteristic or the target variable is not available in a given month.[16] Moreover, to determine the predicted flow distribution we build an additional data set, the prediction sample, that is used for the actual prediction and includes share classes with missing target variables to avoid any form of forward-looking bias. More specifically, when utilizing the training dataset to generate predictions at time t, it is important to note that no predictive inference would be obtained for share classes lacking a corresponding target variable at time $t+h$. This is an implication of excluding share classes with missing target variables from the training sample. After all these adjustments, our training sample consists of 1,183,518 observations (10,557 unique share classes) and the prediction sample contains 1,185,094 observations and 10,562 unique share classes. Both samples range from January 1991 to December 2023.

To ensure scale invariant predictors, we follow Green, Hand, and Zhang (2017) and standardize each variable[17] used in OLS and the applied machine learning algorithms in this paper, except for the neural networks[18], to have a mean of zero and a standard deviation of one. To avoid a look-ahead bias, this standardization is based on the data used in the training set specific to the forecasting cycle.

[16] We deviate from this approach in the case of front-end or back-end loads. If no information is available in CRSP, we assume that no such loads are charged and impute the corresponding variable with 0.

[17] No standardization is required for the variable size of the management team since this predictor is a dummy variable.

[18] In the context of neural networks, we adopt a min-max scaling technique between the range of -1 and 1.

3. Prediction framework and performance evaluation

Our data are organized in a panel structure, with months indexed as $t = 1, 2, \ldots, T$ and share classes as $i = 1, 2, \ldots, N$. In this section, we briefly describe our prediction framework, the five-fold cross-validation approach used for hyperparameter tuning as well as the performance evaluation methodology. Our out-of-sample period spans from January 2000 to December 2023.

3.1. Linear regression - baseline model

The simple linear predictive regression model serves as a benchmark in our prediction exercise. The model is estimated via ordinary least squares (OLS) by solving the following optimization problem, which yields the pooled OLS estimator:

$$\min_{\theta} \mathcal{L}(\theta) = \min_{\theta} \sum_{t=1}^{T-h} \sum_{i=1}^{N} \left(flow_{i,t+h} - z'_{i,t-1}\theta_{t-1} \right)^2 \tag{I.5}$$

where $flow_{i,t+h}$ are the realized fund flows of the ith share class in month $t+h$, $z_{i,t-1}$ is a P-dimensional vector of standardized characteristics (features) for the ith share class in month $t-1$, and $\theta_{t-1} = (\theta_{0,t-1}, \theta_{1,t-1}, \ldots, \theta_{P,t-1})'$ is the P-dimensional parameter vector. The forecast of fund flow in month $t+h$ is then obtained as:

$$\widehat{flow}_{i,t+h} = z'_{i,t}\hat{\theta}_{t-1}. \tag{I.6}$$

If the number of predictors P is relatively small, the OLS estimate of θ is unbiased and efficient. However, the basic OLS model tends to perform poorly when the number of the predictors is large (overfitting) and when nonlinearities and interactions among predictors are important to the forecast. In our empirical analysis, we consider three machine-learning methods that have been recently used in the empirical asset pricing literature: elastic net, tree-based methods (e.g., decision tree, random forest, gradient boosting) and feed-forward neural networks. All these methods can handle large numbers of predictor variables with less risk of overfitting compared to OLS.

To provide a conservative estimate of the improvements possible based on modern ML methods, we let the existing literature inform our benchmark OLS model and saturate it with variables capturing the well-known nonlinear impact of past

performance on fund flows (Sirri and Tufano, 1998; Chevalier and Ellison, 1997). Specifically, we include quadratic terms of the predictors realized alpha ($\alpha^r_{i,t}$) for various factor models and market-adjusted return ($MAR_{i,t}$) as well as the corresponding means of these variables over the past three, six, twelve and 18 months to account for the convexity of the flow-performance relationship. We also create dummy variables for Morningstar ratings (two-star, three-star, four-star, and five-star, where one-star rated share classes form a reference group) to capture a potential nonlinear impact of MS ratings.

Since our realized alpha estimates are based on five different factor models (see Subsection 2.2), for each forecasting period, we train all factor models on the training set and use 5-fold cross-validation (CV)[19] to eventually determine the factor model used to estimate the realized monthly alpha that minimizes the out-of-sample mean squared forecast error (MSFE) for the respective month. The resulting model, using alphas from possibly different factor models in each forecast period, is then re-trained using all data in the training set and applied to predict fund flows using data from the test set. Such an approach is similar to the ML paradigm of using CV to find the best possible set of hyperparameters and guarantees a fair comparison between OLS and ML algorithms. We refer to this model as OLS mixed model. In addition, as a robustness check, we consider one model specification consisting of all predictor variables (full model) and five additional OLS models where each specification uses alphas from one of the five factor models introduced above (CAPM, FF3, FF4, FF5 and FF6) in combination with the remaining predictor variables.

3.2. Machine learning algorithms

ML methods are well-suited for predictive tasks and have recently been applied in the context of empirical asset pricing. Compared to traditional econometric methods the ML literature has focused heavily on out-of-sample performance as the main criterion of interest (Athey and Imbens, 2019). We complement the growing literature on the application of ML in finance (see, among others, Gu, Kelly, and Xiu, 2020; Li and Rossi, 2021; Bali et al., 2022; DeMiguel et al., 2023; Kaniel et al., 2023; Cao et al., 2024; Murray, Xia, and Xiao, 2024; Fausch et al., 2025a) by comparing and evaluating a variety of machine learning algorithms in their ability to predict future fund flows. The ML models implemented in this paper, in an increasing degree of complexity, are elastic net, decision trees, random forest,

[19] A more detailed description of k-fold CV for hyperparameter tuning is provided in Subsection 3.3.

gradient boosting, and feedforward neural networks (FFNNs).[20] We primarily rely on these supervised ML methods as they are reported to exhibit strong predictive performance in regression tasks using structured (panel) data (Chen and Guestrin, 2016; Lundberg et al., 2020) and are known as the most effective machine learning algorithms in various financial studies (e.g., among others Gu, Kelly, and Xiu, 2020; Bianchi, Büchner, and Tamoni, 2021; Li and Rossi, 2021; Bali et al., 2022; DeMiguel et al., 2023).

We now very shortly describe the ML model approaches that we implement. For a more detailed description, we refer the reader to Section I.B in the Appendix. The elastic net (Zou and Hastie, 2005) is a linear method like OLS but uses regularization and variable selection to reduce overfitting. As the elastic net cannot handle nonlinearities independently, it follows the specification of the full OLS model and is based on all predictor variables including the nonlinear terms for the impact of past performance to allow for a fair comparison. The other machine learning methods are able to identify and exploit nonlinearities and higher order interactions that are undetected by OLS and linear machine learning models, such as elastic net, and thus may result in more accurate forecasts. Moreover, tree ensemble methods (random forest and gradient boosting) are very effective at eliminating irrelevant features while neural networks are more sensitive to those predictors. In addition, neural networks are based on a large number of parameters and therefore require a large number of observations to provide accurate estimates. In an empirical asset pricing setting measuring equity risk premia, Gu, Kelly, and Xiu (2020) identify tree-based methods and neural networks as the best-performing algorithms. In another application, Kaniel et al. (2023) use feed-forward neural networks (FFNNs) for predicting the performance (alpha) of actively managed mutual funds. In this context, we also evaluate the forecasting performance of two different FFNNs and compare it to the other machine learning algorithms and the baseline OLS models.[21] However, for application on tabular-style datasets (e.g. panel data sets) using individually meaningful predictors, tree-based models consistently outperform standard deep learning models (Chen and Guestrin, 2016; Lundberg et al., 2020).

While ML models have attracted a lot of attention based on their ability to provide superior forecasts in many settings, recent papers also critically discuss the pitfalls and problems in using these models (Kapoor et al., 2024). These are related

[20] All algorithms are implemented in Python using the *scikit-learn* package, except for neural networks, for which we rely on *TensorFlow*.

[21] The first FFNN consists of one to three hidden layers and two to 32 neurons per hidden layer. In contrast, the second FFNN is a more complex network with three to ten hidden layers and 32 to 1,024 neurons per hidden layer.

to modeling choices and parameter tuning, contamination of results by including future information (forward-looking bias) during the training phase or ambiguous coding, among others, which can lead to concerns regarding reproducibility and validity of results, eventually undermining the credibility of ML-based forecasts. To address these concerns, Kapoor et al. (2024) develop a consensus-based Recommendations for Machine-learning-based Science (REFORMS) checklist. We follow their protocol and present the completed checklist in Table I.F1 in the Appendix.

3.3. Cross-validation, hyperparameter tuning and performance evaluation

Hyperparameter tuning and training: To find the hyperparameters[22] with the best model fit, we use k-fold CV[23] (Hastie et al., 2009), except for the neural network, where we rely on a randomly selected validation set to minimize computation time. As shown by Bergmeir, Hyndman, and Koo (2018) and Goulet Coulombe et al. (2022) k-fold CV performs favorably to alternative techniques that explicitly account for the time-series properties of the data. DeMiguel et al. (2023) confirm this finding when predicting mutual fund alpha using machine learning methods. The k-fold CV approach involves dividing the data set randomly into five folds of approximately equal size. Four folds are used as a training sample with a given combination of hyperparameters and the remaining fold is used as a validation set to evaluate the out-of-sample performance of the trained model based on the chosen hyperparameters using the MSFE (cross-validation error). After completing this procedure for each of the five folds, the hyperparameters that minimize the average cross-validation error are selected for the corresponding ML algorithms. For all algorithms we rely on the *Optuna* hyper-parameter tuning approach suggested by Akiba et al. (2019). Compared to commonly used greedy grid-search and random-search approaches, *Optuna* provides more sophisticated sampling approaches based on Bayesian optimization techniques to find the best possible set of hyperparameters. In this context, we use the tree-structured Parzen estimator approach (TPE) based on independent sampling suggested by Bergstra et al. (2011). In particular, the TPE sampler models the search space by using one Gaussian mixture model $l(x)$ to the set of parameter values associated with the best objective values, and another Gaussian mixture model $g(x)$ to the remaining parameter values. Finally,

[22] The specific hyperparameters that undergo tuning for each machine learning (ML) algorithm are outlined in Section I.B in the Appendix.

[23] In our application, we set $k = 5$.

the parameter value x that maximizes the ratio $l(x)/g(x)$ is chosen.[24] To optimize the computational time for the hyperparameter tuning of the neural networks, we additionally use the hyperband algorithm proposed by Li et al. (2017) as a pruning mechanism and a callback function that stops the training of the neural network if the value of the MSFE on the validation set does not further decrease after five epochs.

As machine learning algorithms are computationally intensive, we determine the best possible set of hyperparameters of the predictive models every two years instead of every month (similar as in Gu, Kelly, and Xiu, 2020; Leippold, Wang, and Zhou, 2022). However, after each monthly forecast, we extend the training sample by one month and retrain the algorithms using the previously obtained hyperparameters. In particular, the hyperparameters used in the first forecast of fund flows in January 2000 are based on the training sample using data from January 1991 to November 1999 and the prediction set of December 1999. These hyperparameters remain unchanged for all monthly fund flow forecasts until December 2001 while the training sample is extended every month.[25] For predictions in January 2002, new hyperparameters using an extended training sample from January 1991 to November 2001 are determined. This procedure is repeated until the end of the sample period.

Evaluating out-of-sample predictive ability: To gauge the out-of-sample predictive ability of all applied machine learning models and the benchmark OLS model we compute the out-of-sample R^2 (OOS R^2) statistic on a share class level, which is defined as

$$R^2_{OOS} = 1 - \frac{\sum_{(i,t)\in\tau_{OOS}} \left(flow_{i,t+h} - \widehat{flow}_{i,t+h}\right)^2}{\sum_{(i,t)\in\tau_{OOS}} \left(flow_{i,t+h} - \overline{flow}_{i,t}\right)^2} \tag{I.7}$$

and measures the reduction in the MSFE compared to a benchmark consisting of the historical mean flow for fund i up to period t, $\overline{flow}_{i,t}$. The forecasting horizon is $h = 1$ month. The predictive power is evaluated on a sample, τ_{OOS}, that is disjoint from the data used in model estimation and hyperparameter tuning. The OOS R^2 statistics pools forecast errors across share classes and over time into a

[24] In our application, for all ML algorithms, except random forest and neural networks, we set the number of trials required for the optimization routine to choose a suitable set of hyperparameter values from a given range to 150. For random forest and neural networks, 100 and 200 trials are used, respectively.

[25] To check the robustness of the obtained results we also apply a rolling window approach of 12-, 36-, and 60-months and find our results to hold.

panel-level assessment of each model. We compute the statistic for each monthly forecasting cycle as well as for the entire out-of-sample period. To evaluate the statistical significance of each model we apply the Clark and West (2007) statistic (two-sided test) to test $H_0 : R^2_{OOS} = 0$ against $H_1 : R^2_{OOS} \neq 0$.

Forecast comparison: To compare the predictive performance of two forecasting methods, $m = 1, 2$, consisting of machine learning methods and linear models, we apply the Diebold-Mariano-West (DMW) pairwise test (Diebold and Mariano, 1995; West, 1996). For this purpose, we define the loss differential as

$$\hat{d}_t^{(1,2)} = \left(\hat{e}_{i,t+h}^{(1)}\right)^2 - \left(\hat{e}_{i,t+h}^{(2)}\right)^2 \tag{I.8}$$

where $\hat{e}_{i,t+h}^{(m)} = flow_{i,t+h} - \widehat{flow}_{i,t+h}$ refers to the forecast error on share class i at time $t+h$ for method m. The DMW statistic to test the null hypothesis of equal predictive accuracy, $H_0 : E\left(\hat{d}_t^{(1,2)}\right) = 0$, against the two-sided alternative is obtained as the t-statistic of a regression with intercept only, $\hat{d}_t^{(1,2)} = \mu + \varepsilon_t$:

$$DMW^{(1,2)} = \frac{\hat{\mu}^{(1,2)}}{\hat{\sigma}_{\hat{\mu}}^{(1,2)}} \tag{I.9}$$

where $\hat{\mu}^{(1,2)}$ denotes the estimated coefficient (time series average of $\hat{d}_t^{(1,2)}$) and $\hat{\sigma}_{\hat{\mu}}^{(1,2)}$ is the corresponding Newey and West (1987) HAC standard error. It is well known that the DMW test can reject the null hypothesis too often, depending on the sample size and the degree of serial correlation in the forecast errors. To address this issue and obtain improved small-sample properties we follow Harvey, Leybourne, and Newbold (1997) and make a bias correction to the DMW statistic in (I.9). The corrected test statistic is obtained as

$$HLN - DMW^{(1,2)} = \sqrt{\frac{T + 1 - 2h + T^{-1}h(h-1)}{T}} DMW^{(1,2)}. \tag{I.10}$$

which is compared to the critical values of a Student t-distribution.

4. Empirical analysis

In our empirical analysis, we explore the predictive ability of the applied statistical models via out-of-sample R^2 and discuss the importance of fund characteristics

and their interactions in predicting future fund flows. Finally, we evaluate whether fund flow prediction can be utilized to consistently differentiate high-performing from low-performing mutual funds. Although our analyses are conducted at the share class level, for simplicity, we refer to share classes as funds in the remainder of this paper.

4.1. Out-of-sample predictability comparison

We begin our analysis by reporting the out-of-sample results of applying OLS regressions and state-of-the-art machine learning algorithms to predict future monthly fund flows. The obtained results are evaluated using the out-of-sample R^2 statistic described in Subsection 3.3. In addition, to discriminate between the forecasting performance of two competing models, we apply the DMW test.

Table I.4 reports the forecasting performance, measured by the out-of-sample R^2 statistic, for the different baseline OLS specifications for a one-month forecasting horizon for the full sample. Additionally, we also report the forecasting performance for the 10% of the share classes with the largest (top) and lowest (bottom) monthly flows. Predicting extreme flows is of particular interest for fund managers in the context of their liquidity management.

The out-of-sample performance of OLS is fairly robust among the various model specifications. For the full sample, R^2_{OOS} is very similar across model specifications and varies between 17.85% (OLS full model) and 18.21% (OLS-FF4). Interestingly, the mixed model specification does not perform particularly well. All reported out-of-sample R^2 are statistically significant at the 1% level, indicating that each applied model specification performs significantly better relative to the corresponding historical mean flow benchmark. For all models, out-of-sample R^2 is considerably higher for the funds with the 10% predicted largest outflows (ranging from 18.75% and 19.41%) than for funds receiving the 10% of predicted largest inflows, where R^2_{OOS} is only between 1.31% to 1.39%.

[Insert Table I.4 here]

Table I.5 shows the forecasting performance of all applied machine learning models based on the out-of-sample R^2 for the full sample and the 10% of the share classes with the largest (top) and lowest (bottom) predicted monthly flows. For the full sample, all machine learning algorithms achieve statistically highly significant positive R^2_{OOS} and show predictive ability for one-month future fund flows. In terms of linear statistical methods, the elastic net performs slightly worse than the

baseline OLS models, implying that the effect of dimension reduction inherent to the algorithm does not lead to improved out-of-sample forecasts.

Compared to the best-performing OLS model (OLS - FF4, 18.21%), which serves as our OLS benchmark model in the remainder of the paper, the decision tree (18.07%) achieves a slightly lower R^2_{OOS}. In contrast, a substantial increase in predictive accuracy is achieved by the random forest and gradient boosting models, improving the R^2_{OOS} to above 22%. While the two neural networks perform better than the decision tree, they do not beat random forest and gradient boosting. A comparison of the predictive accuracy of NN I and NN II reveals that the less complex NN I achieves an out-of-sample R^2 which is 0.45 percentage points higher than that of NN II. Therefore, the increased computational effort associated with the more complex neural network, NN II, is not justified given its inferior performance.

Similar to OLS, the predictive accuracy for machine learning algorithms is much higher for the funds in the bottom decile of the predicted flow distribution. However, the two best performing machine learning methods, random forest and gradient boosting, deliver a R^2_{OOS} in the top decile of the predicted flow distribution that is 9.57 and 8.94 percentage points higher compared to the OLS benchmark model. This shows that the best machine learning methods are much better in predicting extreme flows. Generally, the improved predictive ability of machine learning methods like random forest and gradient boosting demonstrate their effectiveness and dominance in capturing nonlinearities and complex interactions between predictors that appear to be relevant in achieving higher prediction accuracy. Although the absolute level of out-of-sample explanatory power remains moderate, the improvement delivered by non-linear ML models is economically meaningful.

[Insert Table I.5 here]

Overall, the magnitude of improvement in out-of-sample R^2 is comparable to those achieved in studies from the asset pricing literature using machine learning to forecast monthly stock returns (Gu, Kelly, and Xiu, 2020; Leippold, Wang, and Zhou, 2022). While our best-performing forecasting model, random forest, yields an improvement in R^2_{OOS} for the full sample of 4.57 percentage points, the previously mentioned papers on mutual fund performance prediction report an improvement in out-of-sample R^2, based on a comparison of nonlinear ML methods relative to OLS, of between 1.90 (from 0.81 to 2.71, Leippold, Wang, and Zhou (2022)) and 3.86 (from -3.46 to 0.40, Gu, Kelly, and Xiu (2020)) percentage points.

The results are robust when we benchmark our R^2 metric against a forecast value of zero instead of the historical mean. The out-of-sample R^2 of the best-performing

ML method, random forest, is now 4.87 percentage points higher relative to the OLS benchmark. Interestingly, the overall level of R^2 for all models drops by about 5 percentage points when we use zero as naive forecast. This suggests that a forecast of zero is a superior naive forecast compared to the historical mean (Table I.C1).

Instead of applying an expanding window, we repeat the above analysis for the full sample using a 12-, 36-, and 60-month rolling window for training the algorithms (and hyperparameter tuning) to demonstrate the robustness of the forecasting performance documented in Table I.5. The results are shown in Table I.6 and provide three important insights. First, a longer training window implies higher predictive accuracy, documented by a larger R^2_{OOS} measure. Second, the forecasting ability of machine learning algorithms, with the exception of neural networks, is relatively robust regardless of the corresponding training window, while the baseline OLS model shows considerably inferior performance based on the short 12-month training window. Finally, neural networks employ a larger number of hyperparameters and thus require a large number of observations to provide accurate forecasts. In our setting, neural networks are associated with inferior forecasting performance compared to tree-based ensemble models but outperform linear methods (OLS, elastic net) on larger training windows.

[Insert Table I.6 here]

To statistically evaluate the difference in the forecasting abilities of two competing forecasting models we apply modified Diebold and Mariano (1995), West (1996) tests following Equation I.10. The results of the pairwise tests of predictive accuracy are shown in Table I.7. We find that different classes of linear and nonlinear models differ considerably in their predictive performance. Based on the $HLN - DMW$ statistic suggested by Harvey, Leybourne, and Newbold (1997), machine learning methods that capture nonlinearities and allow for interactions outperform linear models. Regarding neural networks, the more complex network structure of NN II surprisingly delivers inferior predictive performance (1% significance level) compared to the relatively simple architecture of NN I. However, gradient boosting and random forest manage to beat both neural networks. Among the linear models, OLS-FF4 significantly outperforms the predictions made by the elastic net. In terms of nonlinear models, the forecasts produced by random forest are significantly better than the predictions obtained by any other machine learning algorithm. Overall, random forest is found to be the best-performing model with the highest predictive accuracy, followed by gradient boosting. While gradient boosting is computationally less expensive, random forest takes the most computing

time of all applied ML algorithms. In conclusion, these results further corroborate the findings obtained by the out-of-sample R^2 analysis above.

[Insert Table I.7 here]

Next, we explore whether the predictive ability of our models depends on the state of the economy and is stable over different economic regimes. As described in Jank (2012), investors react to macroeconomic news by leaving riskier asset classes and entering less risky ones when there is news of an economic downturn. This is empirically confirmed by Pástor and Vorsatz (2020), as during the COVID-19 crisis, actively managed mutual funds experienced rapid outflows during and after the crash. Relatedly, Warther (1995) reports positive correlations between aggregate equity market returns and fund flows. In a more recent study, Ben-Rephael, Kandel, and Wohl (2012) find that aggregate monthly net exchanges to equity funds, as a proxy for shifts between bond and equity funds, are positively contemporaneously correlated with aggregate stock market returns. Furthermore, fund managers are particularly interested in precisely predicting potential outflows during periods of aggregate outflows, where returns and liquidity are typically under increased stress.

Hence, to examine whether our results depend on different market conditions based on the business cycle, monthly aggregate flows, and the return of the aggregate stock market, we compute the out-of-sample R^2 for each algorithm and month, yielding a total of 288 observations. Subsequently, we separate these observations according to the corresponding market conditions (e.g., expansion and recession) and employ Welch (1947) tests to evaluate whether the mean R^2_{OOS} in two states differs significantly. The results are reported in Table I.8.[26] Panel A (business cycle) refers to the state of the business cycle and separates the sample in expansions and recessions as defined by the NBER. All forecasting models except gradient boosting and NN (II) perform slightly better during recessions than expansions based on their R^2_{OOS}, but the differences are not statistically significant based on Welch (1947) tests. As in the overall sample, random forest performs best in both states. Panel B splits the sample into periods where monthly aggregate flows are positive (inflows) or negative (outflows). Results are stable across the two regimes. Although some models perform better during periods of aggregate inflows and some worse, the differences in R^2_{OOS} are small and never statistically significant. Again, random forest remains the best-performing model independent of market conditions. In panel C we divide the sample into periods of positive and

[26]Note that these results differ marginally from the ones obtained in Table I.5, as the reported mean R^2_{OOS} is based on monthly means.

negative aggregate monthly stock market returns using the CRSP value-weighted index. All statistical models, except random forest and gradient boosting (which are still the best performing models in both regimes), show statistically significant higher accuracy in months with negative stock market returns.

Overall, our findings from Table I.8 suggest that the predictive power of our best-performing ML methods in forecasting fund flows, random forest and gradient boosting, is robust and independent of the state of the economy or market regime.

[Insert Table I.8 here]

4.2. Characteristic importance and interactions

In this subsection, we study the importance of fund characteristics and their interactions for the predictions obtained by random forest, which we identified as the machine learning method with the highest forecasting accuracy in the previous section. For that purpose, we estimate model-agnostic SHAP (SHapley Additive exPlanations) values (Lundberg and Lee, 2017; Lundberg et al., 2020). More specifically, for each forecasting cycle, share class, and characteristic (predictor), a SHAP value is calculated, reflecting the predictor's main effect on the respective prediction along with the averaged interaction effects between the predictor and the remaining 63 characteristics. A positive SHAP value indicates that the characteristic has a positive influence on the respective prediction relative to the average forecast, whereas a negative SHAP value indicates that the characteristic has a negative influence. To evaluate the overall importance of each characteristic, we calculate the mean absolute SHAP values, which represent the average strength of that predictor's influence on the model's predictions.[27] In our analysis, we rely on the treeSHAP algorithm[28] which delivers accurate estimates of SHAP values for random forest (Lundberg et al., 2020).

Table I.9 reports the mean of the absolute SHAP values across all observations over the out-of-sample period from January 2000 to December 2023 for the 30 most important characteristics. We find that past monthly flows is the most influential fund characteristic for predicting future fund flows followed by the average flows over the past six and 12 months, respectively. This finding suggests that relatively stable fund characteristics and/or the relatively stable investment behavior of a fund's clientele explain a large part of the predicted flows (which thus are relatively persistent). This finding is also consistent with the strong impact of adding fund

[27] A general discussion of SHAP can be found in Molnar (2019).

[28] We use the *FastTreeSHAP* package of Yang (2022) implemented in Python for computing the SHAP values.

fixed effects in standard OLS regression models for flows on the models' R^2 observed in previous studies.

The next most important predictors are the Morningstar rating and the size of the share class measured by its TNA. It is noteworthy that various measures of past performance, including realized alpha for various factor models, market-adjusted returns and value added seem to be less important in predicting future fund flows.[29] These model-based implications are consistent with the literature showing that mutual fund investors react strongly to simplistic fund rankings (Del Guercio and Tkac, 2008; Reuter and Zitzewitz, 2021; Evans and Sun, 2021) rather than to more sophisticated performance measures based on the CAPM or other commonly used asset pricing models (Ben-David et al., 2022) when allocating capital to mutual funds.

[Insert Table I.9 here]

In the following, we will analyze the stability of these findings for top-10% and bottom-10%-flow fund share classes as well as for fund share classes targeted towards retail and institutional investors, respectively. Panels A and B of Table I.10 show the mean of the absolute SHAP values for the funds in the top (panel A) and bottom (panel B) deciles of the predicted fund flow distribution. For both deciles, flow-related predictors are identified as the most important characteristics, followed by the Morningstar rating and TNA. In the bottom decile of the predicted flow distribution, costs seem to play a more important role: back-end loads and the expense ratio are ranked among the ten most important characteristics for funds with predicted outflows, while they do not appear among the top 15 for funds with the highest predicted inflows.

Panel C and D show the corresponding mean absolute SHAP values for institutional (panel C) and retail (panel D) share classes. The results between these two types of share classes are very similar and flow-related variables and the Morningstar rating are the most relevant features in predicting fund flows. Interestingly, even for institutional investors the Morningstar rating is more important than alphas based on factor models. Furthermore, for institutional investors, the expense ratio is of importance in predicting flows, while for retail funds cost-related characteristics (front-end and back-end loads, expense ratio) are not among the most important

[29] To address concerns that SHAP values could be affected by the high correlation among certain features (in particular the alphas from different factor models, see Figure I.1), we revisited the analysis ex-post, including only the alpha with the highest mean absolute SHAP value (i.e., the realized mean CAPM alpha over the past six months) and dropped all other alphas. Our main results do not qualitatively change and the out-of-sample R^2 remains very similar.

predictors of flows. This pattern suggests that our forecasting model assumes that institutional investors are more fee-sensitive than retail investors.

[Insert Table I.10 here]

To analyze the time variability in the importance of fund characteristics in predicting mutual fund flows, Figure I.2 depicts the importance of the 15 most relevant predictors (according to Table I.9) using the monthly mean absolute SHAP values for random forest. The figure illustrates that from November 2011 onwards, past monthly flows is consistently the most influential fund characteristic for flow predictions over time. However, from January 2000 to October 2011, average flows over the past six months was identified as the most important predictor in the vast majority of months and remained in second place for the rest of the out-of-sample period. These two are followed by the average flows over the past 12 months.

The Morningstar rating is less important during the earlier years of the out-of-sample period, but gained in relevance over time. Starting in the late 2000s, it consistently ranks among the top five variables. This period coincides with the years when Morningstar became a media staple, with mainstream media regularly reporting Morningstar ratings and Morningstar analysts prominently appearing on financial news programs.

While the importance of flow-related measures and the Morningstar rating is generally relatively stable over time, all other predictor variables show a much greater degree of time variation in the out-of-sample period. The most pronounced change in characteristic importance is observed for the expense ratio, which was of marginal relevance at the beginning of the sample and has become much more influential for fund flow predictions recently. To the best of our knowledge, the increased importance of the expense ratio is a new finding and is consistent with investors being more fee-sensitive in more recent years. One possible explanation could be the increased competition from low-cost, passive investment products that have gained in importance over recent years. In this context Cremers et al. (2016) find that actively managed funds charge lower fees when they face more competitive pressure from low-cost index funds.

We also observe that monthly flows lagged by 12 months increased in importance over time, suggesting that fund-specific seasonalities have become more important in more recent years. This effect is likely to be driven by the increased popularity of savings plans, where recurring investments are often automatically made in the same calendar month.

[Insert Figure I.2 here]

While the SHAP values presented hitherto give us some insights into which variables generally matter for flow predictions, they do not tell us anything about the direction in which they influence flows. Thus, we now focus on a better understanding of how various fund characteristics affect the predictions made by random forest in greater detail and analyze the directional impact of these predictors. Figure I.3 shows a beeswarm SHAP plot of the 15 most important features identified by their mean absolute SHAP values over the whole out-of-sample period from January 2000 to December 2023 implied by Table I.9. Each point represents a SHAP value for one observation from a forecast cycle for a specific predictor. If this point is on the positive domain of the horizontal axis, this means that the respective predictor has a positive impact on predicted flows (as compared to the average prediction). The color of the corresponding points indicates whether the realization of the characteristic that led to the respective prediction was high (violet) or low (yellow). The plots for the various predictors are sorted in descending order with the most important variables at the top. We obtain information on the magnitude (low or high feature value) and the directional impact on the fund flow prediction (SHAP value) for each of these characteristics. More specifically, the beeswarm plot does not only reveal the relative importance of features, but also their actual relationships with the predicted outcome.

With the exception of monthly flows, most variables have a relatively clear relationship. Typically, a positive impact can be seen for features with high values and vice versa. Average flows over longer horizons, Morningstar ratings as well as average realized CAPM alphas over various horizons show a predominantly positive directional impact on predicted flows if the feature value is high. This is consistent with the flow-performance relationship in which flows are positively related to past performance (Chevalier and Ellison, 1997; Sirri and Tufano, 1998; Del Guercio and Tkac, 2002; Huang, Wei, and Yan, 2007). A similar finding is observed for market-adjusted returns. Additionally, we find that the relationship of feature value and directional impact of the expense ratio on flow predictions is negative, meaning that expensive funds are predicted to incur outflows. While these findings are consistent with economic rationality of investors, several older studies on mutual fund flows show that funds that charge higher fees typically enjoy higher flows (Sirri and Tufano, 1998; Gil-Bazo and Ruiz-Verdú, 2009; Ivković and Weisbenner, 2009).

However, it is noteworthy that the relationship between the feature value and the directional impact of our most important predictor variable (with the highest mean SHAP value), previous month flows, is ambiguous. In some cases, a positive relationship is observed between high past monthly flows and predicted flows, while

in other instances a negative directional impact is found, meaning that high past flows are associated with low predicted flows. This finding can be interpreted as evidence of a more complex, nonlinear underlying true relationship between previous month flows and predicted flows or a significant change in the directional impact of this variable over time. We will discuss this pattern in greater detail below.

[Insert Figure I.3 here]

To first analyze the temporal stability of the results documented in Figure I.3, we divide the out-of-sample period into consecutive two-year intervals. Figure I.4 shows the corresponding SHAP values. As in the overall out-of-sample period shown above, the relationship between past monthly flows and their impact on predicted flows is ambiguous for most two-year subperiods. This suggests that the relationship documented in the overall sample is not just driven by some periods with positive and some periods with negative relationships. Rather, these findings are consistent with a nonlinear relationship between past monthly flows and predicted flows.

In contrast, the directional impact of average flows over the past six and 12 months is fairly stable across sub-periods, such that higher average past flows are positively related to predicted flows. Additionally, we observe that a higher Morningstar rating consistently leads to a positive impact on the prediction of fund flows in all sub-periods. Moreover, a high expense ratio is negatively related to predicted flows, and the directional importance has become stronger over the sample period.

[Insert Figure I.4 here]

As discussed above, the relationship between flows in the previous month and predicted flows appears to be nonlinear. Hence, in Figure I.5, we further explore the functional form of past monthly flows in more detail by plotting the SHAP value as a function of the corresponding feature value over the whole out-of-sample period. We observe a non-linear relationship between past monthly flows and predicted flows. More specifically, in quadrant I (III) a positive (negative) feature value is associated with a positive (negative) impact on predicted flows, while in quadrant II (IV) a negative (positive) feature value implies a positive (negative) predictive impact.[30] Such a complex non-linear pattern may be influenced by either the main effect or an interaction effect with other predictors. We will explore this possibility in the following.

[Insert Figure I.5 here]

[30] About 93% of the observations are located in quadrant I and III.

A major advantage of most modern machine learning methods is their ability to model a large number of potentially relevant interaction terms. Thus, we next analyze interaction effects for all possible combinations of predictor variables using SHAP interaction values.[31] However, compared to standard SHAP values, calculating SHAP interaction values is impractical for certain ML algorithms due to high computational complexity. For random forest, the time complexity increases substantially making full-scale calculations infeasible. To address this issue adequately, we limit computations of SHAP interaction values to a random subset of share classes as outlined in Section I.D in the Appendix. Testing different subset sizes (1%, 10% and 20%) reveals that a 10% random subset balances accuracy and efficiency, reducing computation time substantially while maintaining reliable results. As an additional robustness check, we compute the SHAP interactions for our second best performing ML method, gradient boosting, based on the full sample of share classes.

Table I.11 shows the strength of the ten most important interactions based on mean absolute SHAP interaction values for random forest based on a random subset of share classes (Panel A) and gradient boosting using the full sample (Panel B). Interactions among the flow-related variables are the three most important interactions, with the strongest interaction found between lagged monthly flows and average flows over the past six months in both Panels. The table further reveals that monthly mean flows (six months) interacted with the Morningstar rating is the next most important interaction that is not entirely flow related, followed by interactions between lagged monthly flows and the Morningstar rating. Generally, the top-5 interactions are very similar across methods, while divergences between Panel A and B can only be observed for the less important interactions.

[Insert Table I.11 here]

As documented in earlier studies (see, e.g., Sirri and Tufano, 1998), there is evidence of a nonlinear relationship between past performance metrics and subsequent fund flows. Building on these findings, we examine this relationship in our predictive framework for the Morningstar rating which has been identified to be both, the most important performance metric as well as the most important non purely flow-related interaction. For that purpose, we use a violin plot[32] and display for all funds the five Morningstar rating categories and their associated SHAP value over the entire out-of-sample period for random forest, displayed in Figure I.6. We observe

[31] This corresponds to $(64 \times (64-1))/2 = 2,016$ interactions.

[32] Note that to better visualize the results, the violin plot does not show data (outliers) that are 1.5 $\times$ the interquartile range (IQR) above the third quartile and below the first quartile.

that the four- and five-star rated funds are predominantly associated with positive flow predictions, as indicated by positive SHAP values, while the lower rated funds are all primarily associated with negative predicted flows. This suggests that the level of the Morningstar rating is fundamentally important in predicting fund flows. Overall, the relationship between the Morningstar rating and its associated SHAP value is convex. The predictions for the funds in the two highest rating categories are on average (median) positively impacted, while for all other funds, the Morningstar rating has a negative impact on the forecast, resulting in lower predicted fund flows.

[Insert Figure I.6 here]

We now further examine the non-linear relationship between past monthly flows and the associated SHAP values from Figure I.5 to verify if this pattern could be influenced by the non-linear impact of the most important non-flow related interacted variable, the Morningstar rating. Figure I.7 is based on Figure I.5, showing the convex relationship between past flows and predicted flows (functional form) and adds the interaction effect with the Morningstar rating (colored dots) for the whole out-of-sample period from January 2000 to December 2023. This figure provides two main insights: First, high past inflows and a high Morningstar rating imply a positive impact on predicted flows, while high past inflows and a low Morningstar rating are associated with a negative impact. This might be due to 'new money' being more sensitive to easily digestible performance information as mirrored in the Morningstar rating. Second, while high past outflows positively impact predictions, low past outflows are mostly associated with negative predicted flows. Overall, this analysis highlights the usefulness of jointly modeling nonlinearities and interaction effects.

[Insert Figure I.7 here]

4.3. Predicted fund flows and performance

The academic literature documents a significantly positive relationship between past fund flows and future fund performance. One strand of the literature attributes this empirically observed flow-performance relationship to investors' ability to identify managerial skill, known as smart money effect (see, among others, Gruber, 1996; Zheng, 1999; Keswani and Stolin, 2008), while another strand attributes it to the persistence of fund flows (see, among others, Coval and Stafford, 2007; Frazzini and Lamont, 2008; Lou, 2012). Our goal in this section is not to analyze the drivers of

the positive flow-performance relationship, but rather to evaluate whether machine learning-based fund flow predictions can be used out-of-sample to consistently differentiate mutual funds with high and low future net returns. We follow Gu, Kelly, and Xiu (2020), Bali et al. (2023), DeMiguel et al. (2023), Kaniel et al. (2023) and Fausch et al. (2025a) and form portfolios using machine learning forecasts. Specifically, we first sort individual share classes into equally weighted top and bottom decile (percentile) portfolios based on their predicted fund flows for the next month. Second, we compute the average realized return for each portfolio in the corresponding month. For every successive month, we follow the procedure outlined in Subsection 3.3 and expand the training sample by one month, train the algorithm on the expanded sample and construct new portfolios using one-month ahead machine learning implied fund flow predictions. As a result, we obtain a time series of monthly out-of-sample net returns of the top and bottom decile (percentile) portfolios from January 2000 to December 2023 (288 monthly observations). Finally, we construct a long-short prediction portfolio of each top and bottom decile (percentile) portfolio to evaluate whether sorting based on high and low fund flow forecasts results in positive mutual fund alpha. The out-of-sample performance of all fund portfolios is evaluated by running a time-series regression of the 288 out-of-sample monthly portfolio excess returns on various risk-factor returns. The portfolio alpha is obtained as the intercept of the time-series regression. We use five risk-factor models to evaluate portfolio performance: the CAPM, the Fama and French (1993) three-factor model (FF3), the Carhart (1997) four-factor model (FF4), the Fama and French (2015) five-factor model (FF5), and the FF5 model augmented with the momentum factor of Carhart (1997) (FF6). The fund flow predictions are obtained using our baseline OLS model with the highest forecasting accuracy (OLS - FF4) as well as the machine learning algorithms used above.

Panel A of Table I.12 reports out-of-sample alphas based on net-returns of the top and bottom decile portfolios and the corresponding long-short portfolio implied by the predicted fund flow distribution based on one-month ahead forecasts. Our results reveal three main findings. First, by using fund-flow predictions we are able to ex-ante separate high from low-performing funds resulting in positive and often significant out-of-sample alphas for the long-short portfolios. The best performing model is gradient boosting which achieves a significant positive alpha relative to all applied factor models. For example, relative to the FF5 and FF6 model, gradient boosting delivers a highly significant alpha of 21 bp per month (2.52% per year). Interestingly, this number is virtually identical to the outperformance achieved by Kaniel et al. (2023) in their analysis aimed at directly predicting fund performance

(rather than flows).[33] However, it is important to note that linear models (OLS, elastic net) perform reasonably well for long-short portfolios based on the top and bottom decile of the predicted fund flow distribution and achieve similar long-short alphas compared to nonlinear algorithms. Second, our machine-learning-based fund flow predictions do not allow us to identify long-only fund portfolios that deliver statistically significant out-of-sample alphas. This finding is consistent with Fausch et al. (2025a), who demonstrate that, after correcting for the forward-looking bias in DeMiguel et al. (2023), directly predicting performance likewise fails to identify mutual fund portfolios that generate significant long-only alpha. Third, our approach does allow investors to identify the worst-performing funds implied by the predicted fund flow distribution. For the bottom decile portfolio and the FF5 and FF6 factor model, all of the applied predictive methods yield a statistically significant negative alpha at the 5% level or higher. This shows that funds associated with high predicted inflows subsequently outperform funds associated with high predicted outflows.

Moving to the extremes of the predicted fund flow distribution and analyzing the performance implications of high and low fund flow forecasts for the top and bottom percentiles yields very favorable results for nonlinear machine learning methods regarding long-short portfolios. Panel B of Table I.12 reports that at the 5% significance level or higher, gradient boosting delivers a significant alpha between 32 and 38 bps (3.84% to 4.56% per year) depending on the applied factor model. Slightly weaker results are obtained for the other non-linear ML models, while the linear model, OLS and elastic net, fail to deliver significant alpha.

[Insert Table I.12 here]

4.4. Fund flow prediction accuracy and performance

In the final part of our paper, we analyze whether the higher forecasting accuracy offered by ML models has an impact on performance at the aggregate fund level. If, for a given fund, ML models can improve the accuracy of flow prediction compared to more basic methods (e.g., historical mean), then the liquidity management of that fund might be improved by having access to such flow prediction methods.[34] Put

[33] Depending on the factor model analyzed and using only fund-specific predictor variables, they document an outperformance ranging from 1.80% to 2.52%. They further show that adding stock-specific characteristics and sentiment does not result in long-short portfolios with higher alpha.

[34] Etula et al. (2020) find that the month-end liquidity management of mutual funds impacts their performance.

differently, if the basic prediction rules perform particularly poorly (relative to ML-based predictions), this could have a negative impact due to difficulties these funds might face in predicting flows with eventual adverse performance implications. In the following, we analyze whether performance is indeed worse for funds where ML models would deliver better flow predictions than basic forecasts. The underlying assumption of this analysis is that fund managers were not yet using sophisticated ML-based flow forecasts during our sample period.

To quantify the potential performance effect of higher flow prediction accuracy, we sort funds into quintile portfolios based on the $R^2_{OOS,j}$, for the baseline OLS model and our most accurate ML algorithm (random forest), for each period t with a one-month holding period $(t+1)$. Funds sorted in the bottom quintile are those with the lowest forecasting accuracy, while funds sorted in the top quintile are those with the highest forecasting accuracy relative to a historical mean benchmark. In this context, we suggest that a higher R^2_{OOS} in period t, i.e., a more accurate forecast with a more sophisticated forecasting model compared to the naïve forecast (historical average), is associated with a greater potential for improvement in model-based liquidity management, resulting in a lower portfolio alpha in the following month, since for example using advanced forecasting techniques prevent large flow surprises that might adversely affect fund performance (Edelen, 1999).

In particular, we analyze a long-short portfolio comprising the bottom and top quintiles to evaluate whether the enhanced forecasting accuracy of OLS and random forest, compared to a historical mean forecast, is related to fund performance. Using this methodology, we generate a time series of monthly quintile portfolio returns and assess their performance implications by estimating a regression of these returns on various factor models. The results are reported in Table I.13. We observe an inverse relationship between forecasting accuracy and performance for both OLS and random forest. The portfolio consisting of funds in the lowest quintile, where historical means perform relatively well (i.e., where predictive models offer little improvement), generates the highest alpha. In contrast, funds in the highest quintile, which might offer greater potential for liquidity management optimization, yield the lowest portfolio alpha. The difference in performance between the top and bottom quintiles (long-short portfolios) is only statistically significant for our best ML-based prediction model, random forest. More specifically, the difference in performance in terms of forecasting accuracy of random forest is positive and ranges from 4.4 bp to 6.1 bp per month (0.53% to 0.73% per year) for all considered factor models. These magnitudes are not very large but do seem realistic for the potential impact caused by difficulties in predicting flows and thus an eventually more

challenging liquidity management.

[Insert Table I.13 here]

To further examine the previously identified effect of forecasting accuracy on fund performance and to alleviate concerns that the previous results may be attributable to fund characteristics other than flow prediction accuracy, we also conduct a panel regression analysis. We regress various measures of mutual fund performance (CAPM, FF3, FF4, FF5, FF6 alpha) in $t+1$ on a dummy variable (R^2_{OOS} dummy) that takes the value 1 if a fund j is in the top quintile (highest forecasting accuracy) and equals 0 if the fund is in the bottom quintile (lowest forecasting accuracy) according to the monthly cross-sectional R^2_{OOS} distribution for random forest in period t. We include the respective lagged performance measure, the log of fund size (TNA), the log of fund age, the expense ratio as well as fund and month fixed effects as control variables which are common in the literature. Standard errors are clustered at the fund level. The results in Table I.14 show that the performance of funds in the top quintile, where the historical mean is a particularly poor predictor relative to advanced forecasting models, is significantly lower than that of funds in the bottom quintile.[35] The effect is again economically meaningful as well as statistically significant and implies an improvement in performance between 0.31% to 0.48% per year, consistent with the results reported in Table I.13. Furthermore, our results regarding the other control variables also generally confirm previous findings from the mutual fund literature. Overall, our results suggest that improving the accuracy of flow forecasting using ML-based fund flow predictions can lead to better fund performance.

[Insert Table I.14 here]

5. Conclusion

In this paper, we apply machine learning techniques and a large set of predictors to forecast one-month ahead fund flows. Our results show that nonlinearities and interactions that can be taken into account by modern machine learning models prove to be important in predicting future fund flows. Hence, these models significantly outperform standard linear frameworks in terms of out-of-sample R^2. We find that random forest is the best performing machine learning algorithm, followed

[35] If we repeat the analysis for OLS-based forecasts, the coefficient of the dummy variable is statistically insignificant for all performance measures, confirming the findings in Table I.13 that using OLS as a forecasting method does not yield a significant performance gain.

by gradient boosting. However, other machine learning methods such as neural networks also possess a good forecasting ability and still clearly outperform the baseline OLS model, while less complex machine learning algorithms such as elastic net and decision tree are inferior in their predictive ability compared to OLS.

Our results also show that predictors based on previous flows (past 1-month flows, average flows over six and 12 months), the Morningstar rating, and TNA are the most important variables for predicting future fund flows. In contrast, measures of past performance such as realized alpha from various factor models, value added, or market-adjusted returns are only of minor importance when controlling for the Morningstar rating. These findings are consistent with the recent literature showing that investors react strongly to simplistic fund rankings rather than to more sophisticated performance measures based on commonly used asset pricing models.

Moreover, our results reveal that over time the expense ratio became much more influential in predicting fund flows and - in contrast to the earlier literature - can be shown to have a strong negative impact on flows. We also identify a convex relationship between past flows and predicted flows as well as a nonlinear impact of the Morningstar rating on predicted flows. In addition, the interaction effect of past flows with the Morningstar rating substantially impacts the forecasts. This effect reveals that high past flows and a high Morningstar rating are associated with higher predicted flows, while high past inflows and a low Morningstar rating are associated with a negative impact, suggesting that 'new money' is strongly rating sensitive.

Finally, we also show that ML-based fund flow predictions can be used to ex-ante separate high- and low-performing mutual funds. Furthermore, funds where flow prediction accuracy can be improved based on ML models as compared to more basic flow forecasts perform significantly worse, suggesting hat these funds suffer from suboptimal liquidity management due to the difficulty of forecasting fund flows accurately.

Overall, our findings highlight the importance of the complex interplay between various variables in predicting flows. They show how asset managers could use state-of-the-art ML techniques to better predict flows and optimize their liquidity policy as well how investors can use these predictions to ex-ante separate high from low-performing funds.

Table I.1: **Number of share classes per year with monthly average, median and quantile flows**

This table reports the number of share classes for each year in our sample and the corresponding flow distribution (in percent) described by its mean, median, standard deviation, lower (Q1) and upper quantile (Q3). The sample period spans from January 1991 to December 2023.

		Flow distribution				
Year	Number of share classes	Mean	Std	Q1	Median	Q3
1991	309	0.98	5.10	-0.89	0.15	2.00
1992	339	1.27	4.75	-0.54	0.48	2.24
1993	536	1.33	6.76	-0.78	0.39	2.29
1994	618	0.77	5.73	-1.01	0.12	1.75
1995	787	0.79	6.66	-1.10	0.05	1.74
1996	1,033	0.87	6.28	-1.01	0.21	1.97
1997	1,263	1.00	6.89	-1.03	0.22	2.02
1998	1,611	0.52	6.77	-1.33	-0.03	1.69
1999	1,923	-0.06	6.47	-2.11	-0.44	1.27
2000	2,237	0.50	7.17	-1.67	-0.15	1.76
2001	2,236	0.38	5.77	-1.30	-0.27	1.14
2002	2,374	-0.27	5.60	-1.91	-0.69	0.68
2003	3,753	0.40	6.41	-1.41	-0.24	1.30
2004	4,038	-0.05	5.77	-1.81	-0.54	0.98
2005	4,434	-0.13	7.39	-2.13	-0.75	0.88
2006	4,686	-0.30	5.76	-2.13	-0.73	0.81
2007	4,929	-0.51	5.90	-2.22	-0.79	0.64
2008	6,461	-0.85	5.84	-2.51	-1.00	0.32
2009	6,379	-0.50	5.77	-1.93	-0.70	0.55
2010	6,312	-0.47	5.97	-1.84	-0.73	0.49
2011	6,311	-0.45	6.17	-1.92	-0.75	0.56
2012	6,002	-0.73	5.54	-1.97	-0.85	0.31
2013	6,107	-0.12	5.57	-1.49	-0.44	0.84
2014	6,281	-0.33	5.23	-1.52	-0.51	0.62
2015	6,575	-0.42	5.57	-1.55	-0.54	0.55
2016	6,624	-0.90	5.68	-1.99	-0.87	0.15
2017	6,945	-0.98	6.60	-1.94	-0.82	0.17
2018	6,936	-0.83	5.37	-1.74	-0.73	0.26
2019	6,953	-0.93	5.03	-1.85	-0.80	0.16
2020	6,984	-0.98	5.13	-2.13	-0.89	0.19
2021	7,041	-0.51	4.30	-1.49	-0.53	0.43
2022	7,080	-0.66	3.73	-1.46	-0.57	0.26
2023	7,104	-0.85	3.46	-1.56	-0.66	0.14
Total	**13,376**	**-0.47**	**5.62**	**-1.79**	**-0.65**	**0.55**

Table I.2: **Predictor variables by category**

This table shows all 64 predictor variables sorted into two categories. The first category represents fund-specific characteristics, while the second set of characteristics is related to macroeconomic variables.

	Fund characteristics
(1)	Flows
(2)	Flows (mean past 6 months)
(3)	Flows (mean past 12 months)
(4)	Flows (lagged 12 months)
(5)	Volatility of flows
(6)	Month
(7)	Expense ratio
(8)	Front-end load
(9)	Back-end load
(10)	Turnover ratio
(11)	Age
(12)	Size of management team
(13)	Morningstar rating
(14)	Change in Morningstar rating
(15)	Volatility of return
(16)	TNA
(17)	TNA (Fund)
(18)	MAR
(19)	MAR (mean past 3 months)
(20)	MAR (mean past 6 months)
(21)	MAR (mean past 12 months)
(22)	MAR (mean past 18 months)
(23)	CAPM alpha
(24)	CAPM alpha (mean past 3 months)
(25)	CAPM alpha (mean past 6 months)
(26)	CAPM alpha (mean past 12 months)
(27)	CAPM alpha (mean past 18 months)
(28)	FF3 alpha
(29)	FF3 alpha (mean past 3 months)
(30)	FF3 alpha (mean past 6 months)
(31)	FF3 alpha (mean past 12 months)
(32)	FF3 alpha (mean past 18 months)
(33)	FF4 alpha
(34)	FF4 alpha (mean past 3 months)
(35)	FF4 alpha (mean past 6 months)
(36)	FF4 alpha (mean past 12 months)
(37)	FF4 alpha (mean past 18 months)
(38)	FF5 alpha
(39)	FF5 alpha (mean past 3 months)
(40)	FF5 alpha (mean past 6 months)
(41)	FF5 alpha (mean past 12 months)
(42)	FF5 alpha (mean past 18 months)
(43)	FF6 alpha
(44)	FF6 alpha (mean past 3 months)
(45)	FF6 alpha (mean past 6 months)
(46)	FF6 alpha (mean past 12 months)
(47)	FF6 alpha (mean past 18 months)
(48)	Factor loading RMRF (FF6)
(49)	Factor loading SMB (FF6)
(50)	Factor loading HML (FF6)
(51)	Factor loading RMW (FF6)
(52)	Factor loading CMA (FF6)
(53)	Factor loading MoM (FF6)
(54)	Value added
(55)	Adjusted R^2 (FF6)
	Macroeconomic variables
(56)	Short-term interest rate
(57)	Long-term yield
(58)	Term spread
(59)	Credit spread
(60)	Stock market return
(61)	Volatility of stock market return
(62)	VIX
(63)	Risk aversion index
(64)	Uncertainty index

Table I.3: **Summary statistics of predictor variables**

This table reports monthly descriptive statistics of all fund characteristics (except lags and means of various variables) and macroeconomic variables considered in the empirical analysis except the expense and turnover ratio which are reported based on annual data. The size of management team is represented as a dummy variable taking a value of zero if the fund is single-managed and one otherwise. All fund-related variables are measured at the share-class level for actively managed US domestic equity funds and the sample spans the period from January 1991 to December 2023.

Characteristic	**Mean**	**Std**	**Q1**	**Median**	**Q3**
Flows	-0.52%	5.63%	-1.91%	-0.66%	0.53%
Volatility of flows	4.31%	4.13%	1.59%	3.07%	5.63%
Month	6.50	3.44	4.00	7.00	9.00
Expense ratio	1.24%	0.54%	0.88%	1.16%	1.55%
Front-end load	1.05%	2.14%	0.00%	0.00%	0.00%
Back-end load	0.64%	1.28%	0.00%	0.00%	1.00%
Turnover ratio	70.87%	86.05%	28.00%	51.00%	88.00%
Age	168.02	121.47	89.00	137.00	208.00
Size of Management Team	0.71	0.45	0.00	1.00	1.00
Morningstar rating	3.06	1.03	2.00	3.00	4.00
Change in Morningstar rating	0.00	0.37	0.00	0.00	0.00
Volatility of return	5.00%	2.05%	3.51%	4.70%	6.07%
TNA	815.79	3447.95	33.30	116.00	461.00
TNA (Fund)	3,588.93	13,454.07	199.50	695.00	2,124.00
MAR	-0.10%	2.71%	-1.20%	-0.12%	0.98%
CAPM alpha	-0.16%	2.70%	-1.21%	-0.14%	0.93%
FF3 alpha	-0.19%	2.27%	-1.03%	-0.14%	0.70%
FF4 alpha	-0.20%	2.28%	-1.03%	-0.15%	0.69%
FF5 alpha	-0.16%	2.34%	-1.02%	-0.14%	0.73%
FF6 alpha	-0.18%	2.35%	-1.03%	-0.15%	0.71%
Factor loading RMRF (FF6)	0.98	0.19	0.90	0.98	1.06
Factor loading SMB (FF6)	0.19	0.35	-0.07	0.08	0.43
Factor loading HML (FF6)	0.01	0.35	-0.17	0.02	0.19
Factor loading RMW (FF6)	-0.05	0.32	-0.18	-0.02	0.11
Factor loading CMA (FF6)	-0.11	0.41	-0.29	-0.09	0.08
Factor loading MoM (FF6)	0.01	0.18	-0.06	0.00	0.08
Value added	-0.40	47.28	-1.05	-0.02	0.83
Adjusted R^2 (FF6)	0.89	0.14	0.89	0.94	0.96
Short-term interest rate	1.65%	1.83%	0.09%	0.95%	2.74%
Long-term yield	3.56%	1.48%	2.55%	3.44%	4.75%
Term spread	1.91%	1.43%	0.84%	1.85%	3.06%
Credit spread	1.01%	0.40%	0.78%	0.92%	1.13%
Stock market return	0.73%	4.58%	-1.77%	1.33%	3.54%
Volatility of stock market return	4.31%	1.33%	3.05%	4.37%	5.56%
VIX	19.93	8.03	13.95	18.00	23.84
Risk-aversion index	3.07	0.75	2.67	2.86	3.17
Uncertainty index	2.16	0.52	1.86	2.11	2.42

Figure I.1: **Correlation matrix between fund characteristics**

This figure shows the correlation coefficients of predictor variables that exceed a threshold of 0.6 and are thus considered substantially correlated. The sample period spans from January 1991 to December 2023.

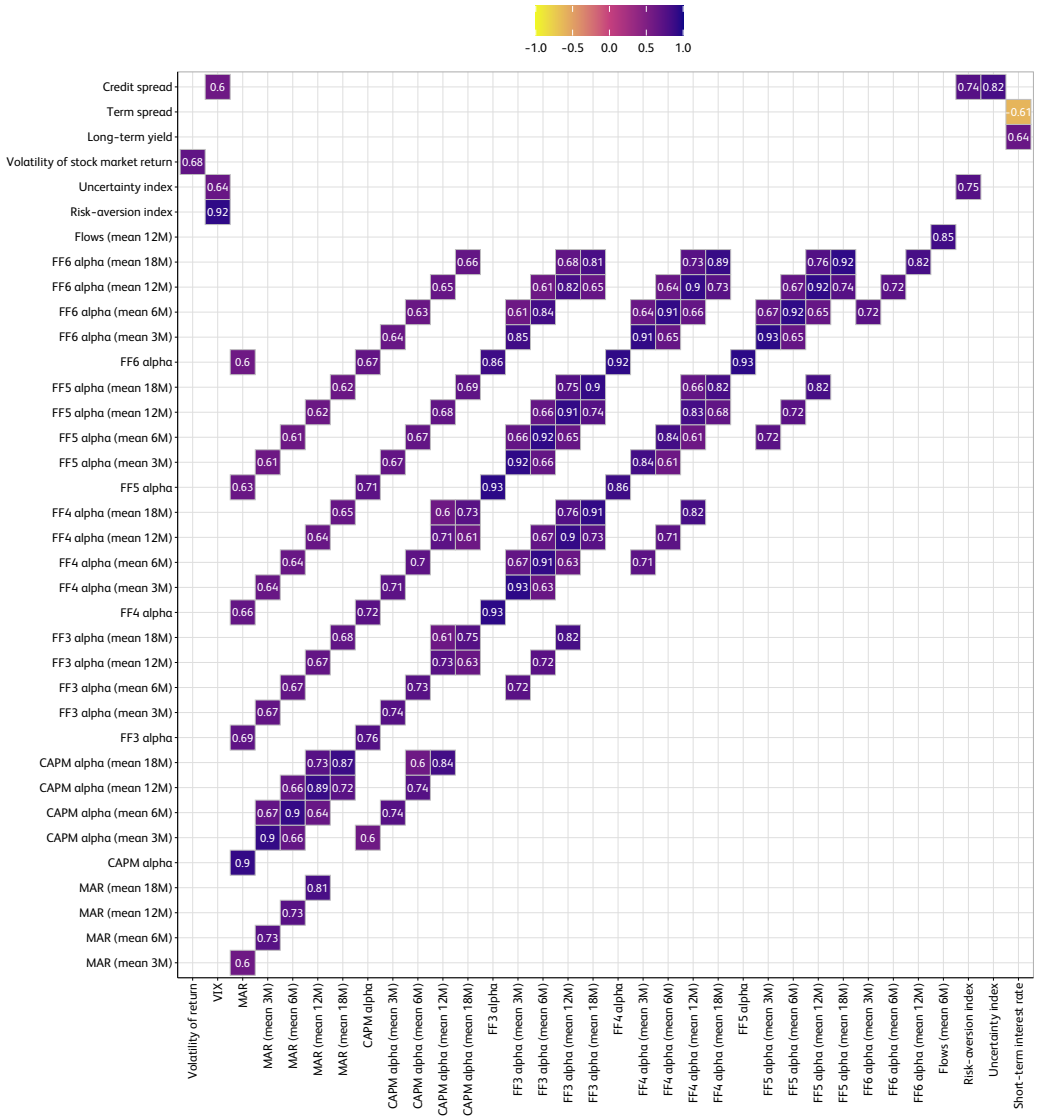


Table I.4: **Out-of-sample OLS model comparison based on R^2_{OOS}**

This table reports the out-of-sample performance measured by the out-of-sample R^2 statistic, as defined in Equation I.7, for the different baseline OLS models for a one-month forecasting horizon for the full sample including all predictions and the monthly top (bottom) 10% of the share classes with the highest (lowest) flows. The statistical significance of the R^2_{OOS} measure is evaluated by the Clark and West (2007) statistic (two-sided test). The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5%, and 1% level, respectively. The out-of-sample period spans from January 2000 to December 2023.

	Out-of-sample R^2		
	Full sample	**Top 10%**	**Bottom 10%**
OLS - Full Model	0.1785***	0.0138***	0.1875***
OLS - CAPM	0.1800***	0.0139***	0.1886***
OLS - FF3	0.1819***	0.0131***	0.1937***
OLS - FF4	0.1821***	0.0131***	0.1941***
OLS - FF5	0.1820***	0.0134***	0.1939***
OLS - FF6	0.1819***	0.0135***	0.1938***
OLS - Mixed Model	0.1788***	0.0133***	0.1882***
Observations	1,106,802	110,686	110,686

Table I.5: **Out-of-sample machine learning model comparison based on R^2_{OOS}**

This table reports the out-of-sample performances measured by the out-of-sample R^2 statistic, as defined in Equation I.7, for the applied machine learning methods for a one-month forecasting horizon for the full sample including all predictions and the monthly top (bottom) 10% of the share classes with the highest (lowest) flows. The statistical significance of the R^2_{OOS} measure is evaluated by the Clark and West (2007) statistic (two-sided test). The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5%, and 1% level, respectively. The out-of-sample period spans from January 2000 to December 2023.

	Out-of-sample R^2		
	Full sample	**Top 10%**	**Bottom 10%**
Elastic net	0.1721***	-0.0186***	0.1760***
Decision tree	0.1807***	0.0375***	0.1739***
Random forest	0.2278***	0.1088***	0.2010***
Gradient boosting	0.2237***	0.1025***	0.1980***
Neural network (I)	0.1963***	0.0407***	0.2005***
Neural network (II)	0.1918***	0.0268***	0.2002***
Observations	1,106,802	110,686	110,686

Table I.6: **Out-of-sample machine learning model comparison based on R^2_{OOS} using a 12-, 36-, and 60-month rolling-window on the training data**

This table reports the out-of-sample performances measured by the out-of-sample R^2 statistic, as defined in Equation I.7, for the applied machine learning methods for a one-month forecasting horizon for the full sample including all predictions using a 12-, 36-, and 60-month rolling window for the training data. The statistical significance of the R^2_{OOS} measure is evaluated by the Clark and West (2007) statistic (two-sided test). The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5%, and 1% level, respectively. The out-of-sample period spans from January 2000 to December 2023.

	Out-of-sample R^2		
	12 Months	**36 Months**	**60 Months**
OLS - FF4	-0.0558***	0.1756***	0.1774***
Elastic net	0.1719***	0.1737***	0.1733***
Decision tree	0.1600***	0.1711***	0.1776***
Random forest	0.2158***	0.2228***	0.2254***
Gradient boosting	0.2081***	0.2188***	0.2222***
Neural network (I)	-0.2475***	0.1761***	0.2019***
Neural network (II)	0.1231***	0.1719***	0.1836***
Observations	1,106,802	1,106,802	1,106,802

Table I.7: **Forecast comparison based on the modified DMW statistic**

This table shows a comparison of the predictive performance of the applied machine learning methods and the best performing OLS model (OLS-FF4) according to the R^2_{OOS} using the modified Diebold and Mariano (1995), West (1996) test statistic defined in Equation I.10. A positive number indicates that the model in the column outperforms the model in the row. The forecasting horizon corresponds to one month. The level of significance of this outperformance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5%, and 1% level, respectively. The out-of-sample period spans from January 2000 to December 2023.

	Elastic Net	Decision Tree	Random Forest	Gradient Boosting	Neural Network (I)	Neural Network (II)
OLS - FF4	-23.78***	-0.64	33.04***	25.99***	14.78***	8.98***
Elastic Net		4.21***	40.87***	32.36***	26.52***	19.32***
Decision Tree			24.47***	19.79***	7.80***	5.44***
Random Forest				-4.22***	-28.56***	-31.76***
Gradient Boosting					-20.37***	-23.05***
Neural Network (I)						-5.58***

Table I.8: **Out-of-sample model comparison based on R^2_{OOS} in different market conditions**

This table reports the out-of-sample performances measured by the out-of-sample R^2 statistic, separately in various market conditions for the applied machine learning methods and the best performing OLS model (OLS-FF4) for a one-month forecasting horizon. Panel A (business cycle) refers to the state of the economy and separates the sample in expansions and recessions as defined by the NBER. Panel B (aggregate flows) splits the sample in periods where aggregate flows are positive (inflows) or negative (outflows) while panel C (market returns) divides the sample into periods of positive and negative aggregate monthly stock market returns using the CRSP value-weighted index. To test whether the average forecasting accuracy differs in each panel we apply a Welch (1947) test and report the corresponding value of the $t-$statistic in the column $t-stat$. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5%, and 1% level, respectively. The out-of-sample period spans from January 2000 to December 2023.

	A. Business cycle			**B. Aggregate flows**			**C. Market returns**		
	Expansion	Recession	$t-stat$	Inflows	Outflows	$t-stat$	Positive	Negative	$t-stat$
OLS - FF4	0.1922	0.1951	-0.1874	0.1949	0.1921	0.1758	0.1830	0.2093	-2.7924***
Elastic Net	0.1828	0.1940	-0.6761	0.1848	0.1838	0.0613	0.1696	0.2091	-4.1563***
Decision Tree	0.1867	0.1916	-0.2918	0.1843	0.1879	-0.1960	0.1801	0.1997	-1.8512*
Random Forest	0.2355	0.2367	-0.0777	0.2473	0.2333	0.8743	0.2314	0.2429	-1.1898
Gradient Boosting	0.2327	0.2245	0.5015	0.2380	0.2306	0.4505	0.2270	0.2402	-1.3586
Neural Network (I)	0.2059	0.2128	-0.4248	0.2008	0.2078	-0.3785	0.1975	0.2227	-2.5293**
Neural Network (II)	0.2009	0.1901	0.6119	0.1887	0.2019	-0.5999	0.1906	0.2156	-2.1995**
Observations	998,720	108,082	E=257	136,111	970,691	I=48	719,601	387,201	P=183
Number of months	257	31	R=31	48	240	O=240	183	105	N =105

Table I.9: **Mean of absolute SHAP values for random forest**

This table reports the mean absolute SHAP values over all forecasting periods for the 30 most important fund characteristics for a one-month forecasting horizon for random forest. SHAP values are calculated over the out-of-sample period from January 2000 to December 2023.

Rank	Characteristic	Mean
1	Flows	0.1102
2	Flows (mean 6M)	0.1000
3	Flows (mean 12M)	0.0630
4	Morningstar Rating	0.0267
5	TNA	0.0137
6	Realized CAPM alpha (mean 6M)	0.0125
7	Flows (lag 12M)	0.0110
8	Volatility of flows	0.0105
9	Realized CAPM alpha (mean 3M)	0.0103
10	Realized CAPM alpha (mean 12M)	0.0102
11	Market-adjusted return (mean 12M)	0.0091
12	Market-adjusted return (mean 6M)	0.0091
13	Market-adjusted return (mean 3M)	0.0077
14	Expense ratio	0.0072
15	Realized FF3 alpha (mean 12M)	0.0070
16	Realized CAPM alpha	0.0063
17	Realized FF4 alpha (mean 12M)	0.0063
18	Long-term yield	0.0062
19	Month	0.0062
20	TNA (Fund)	0.0060
21	Market-adjusted return	0.0058
22	Back-end load	0.0054
23	Market return	0.0053
24	Market-adjusted return (mean 18M)	0.0052
25	Volatility of market return	0.0047
26	Age	0.0042
27	Realized FF3 alpha (mean 18M)	0.0039
28	Turnover ratio	0.0037
29	Realized CAPM alpha (mean 18M)	0.0036
30	Realized FF4 alpha (mean 18M)	0.0036

Table I.10: **Mean of absolute SHAP values for random forest - Top and bottom decile of the predicted fund flow distribution and for institutional and retail share classes**

This table reports the mean absolute SHAP values over all forecasting periods for the 15 most important fund characteristics for a one-month forecasting horizon for random forest. SHAP values are calculated over the out-of-sample period from January 2000 to December 2023. Panels A and B report SHAP values for the share classes in the top and bottom decile of the predicted fund flow distribution. Panels C and D show the corresponding SHAP values for institutional and retail share classes.

A. Top 10%			**B. Bottom 10%**	
Rank	Characteristic	Mean	Characteristic	Mean
1	Flows	0.3006	Flows (mean 6M)	0.1923
2	Flows (mean 6M)	0.2331	Flows (mean 12M)	0.1636
3	Flows (mean 12M)	0.1065	Flows	0.1471
4	Morningstar Rating	0.0583	Morningstar Rating	0.0236
5	TNA	0.0367	TNA	0.0174
6	Volatility of flows	0.0307	Volatility of flows	0.0171
7	Realized CAPM alpha (mean 6M)	0.0197	Flows (lag 12M)	0.0169
8	Flows (lag 12M)	0.0172	Realized CAPM alpha (mean 6M)	0.0142
9	Realized CAPM alpha (mean 3M)	0.0171	Back-end load	0.0127
10	TNA (Fund)	0.0146	Expense ratio	0.0124
11	Realized CAPM alpha	0.0145	Market-adjusted return (mean 6M)	0.0123
12	Realized CAPM alpha (mean 12M)	0.0143	Realized CAPM alpha (mean 12M)	0.0121
13	Turnover ratio	0.0135	Market-adjusted return (mean 12M)	0.0120
14	Market-adjusted return (mean 6M)	0.0128	Realized CAPM alpha (mean 3M)	0.0104
15	Market-adjusted return (mean 12M)	0.0127	Month	0.0097
C. Institutional			**D. Retail**	
Rank	Characteristic	Mean	Characteristic	Mean
1	Flows	0.1188	Flows	0.1049
2	Flows (mean 6M)	0.1020	Flows (mean 6M)	0.0988
3	Flows (mean 12M)	0.0676	Flows (mean 12M)	0.0602
4	Morningstar Rating	0.0318	Morningstar Rating	0.0236
5	TNA	0.0147	TNA	0.0131
6	Flows (lag 12M)	0.0131	Realized CAPM alpha (mean 6M)	0.0125
7	Realized CAPM alpha (mean 6M)	0.0126	Volatility of flows	0.0116
8	Realized CAPM alpha (mean 3M)	0.0103	Realized CAPM alpha (mean 3M)	0.0104
9	Realized CAPM alpha (mean 12M)	0.0102	Realized CAPM alpha (mean 12M)	0.0102
10	Market-adjusted return (mean 6M)	0.0097	Flows (lag 12M)	0.0098
11	Market-adjusted return (mean 12M)	0.0092	Market-adjusted return (mean 12M)	0.0090
12	Volatility of flows	0.0088	Market-adjusted return (mean 6M)	0.0088
13	Expense ratio	0.0085	Market-adjusted return (mean 3M)	0.0074
14	Market-adjusted return (mean 3M)	0.0080	Realized FF3 alpha (mean 12M)	0.0068
15	Realized FF3 alpha (mean 12M)	0.0073	Realized CAPM alpha	0.0067

Figure I.2: **Evolution of characteristic importance for random forest over time**

This figure plots the time-variation of characteristic importance of the 15 most important predictors identified by their mean absolute SHAP value over the whole out-of-sample period (January 2000 to December 2023) and a one-month forecasting horizon for random forest. Characteristics are ranked according to their importance and range from 1 for the most important predictor and 64 for the least important predictor.

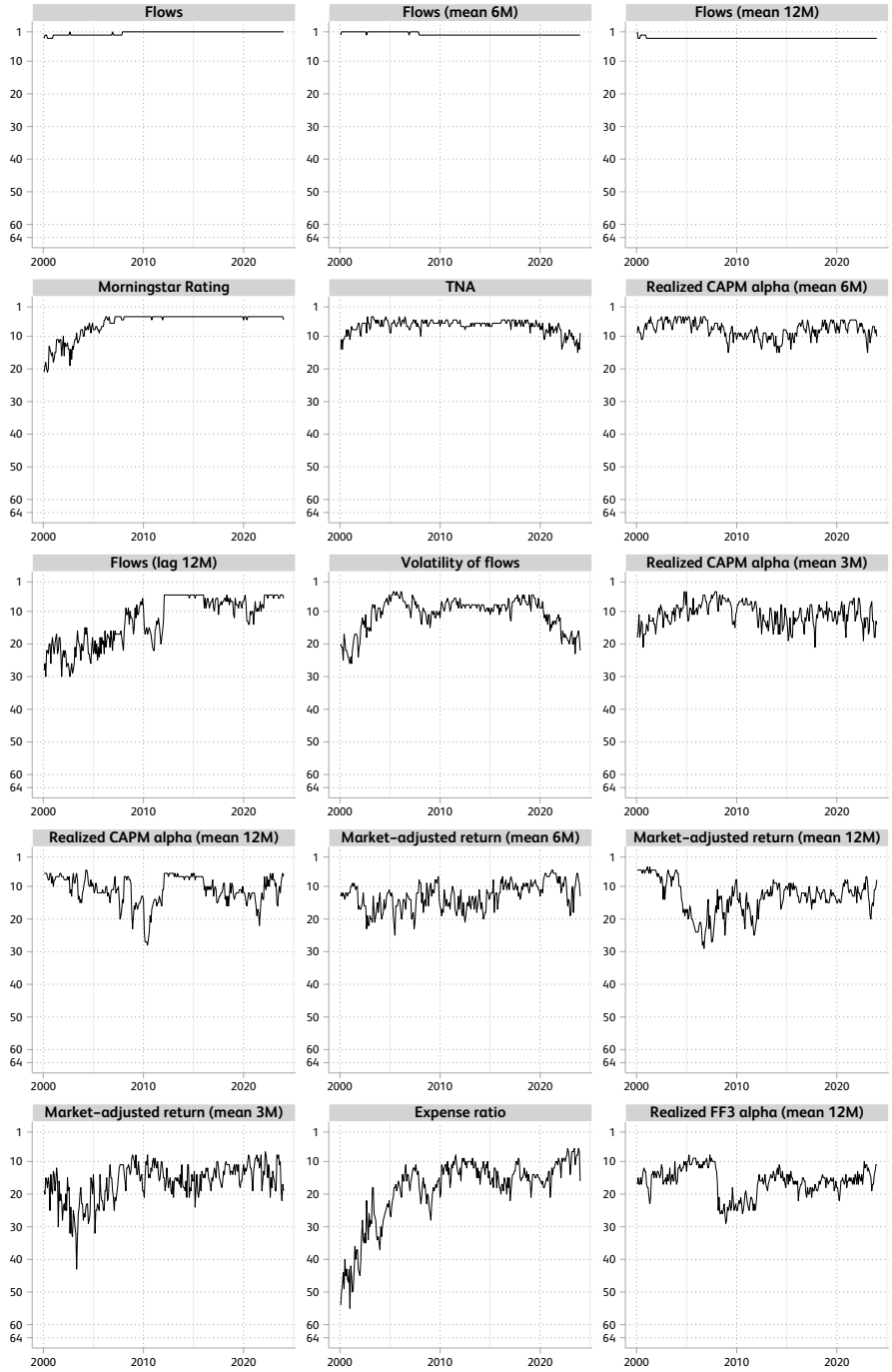


Figure I.3: **Directional impact of the characteristics for random forest**

This figure shows a beeswarm SHAP plot which indicates the directional impact of the 15 most important characteristics identified by their mean absolute SHAP value over the whole out-of-sample period (2000 to 2023) and a one-month forecasting horizon for random forest. Predictors are sorted in descending order with the most important variables at the top. Each dot on the chart corresponds to one SHAP value for a prediction and feature. The magnitude of feature values is color-coded from yellow (low) to purple (high). The points are distributed horizontally according to their SHAP values which measure each feature's directional impact on the predictive model outcome (flow prediction).

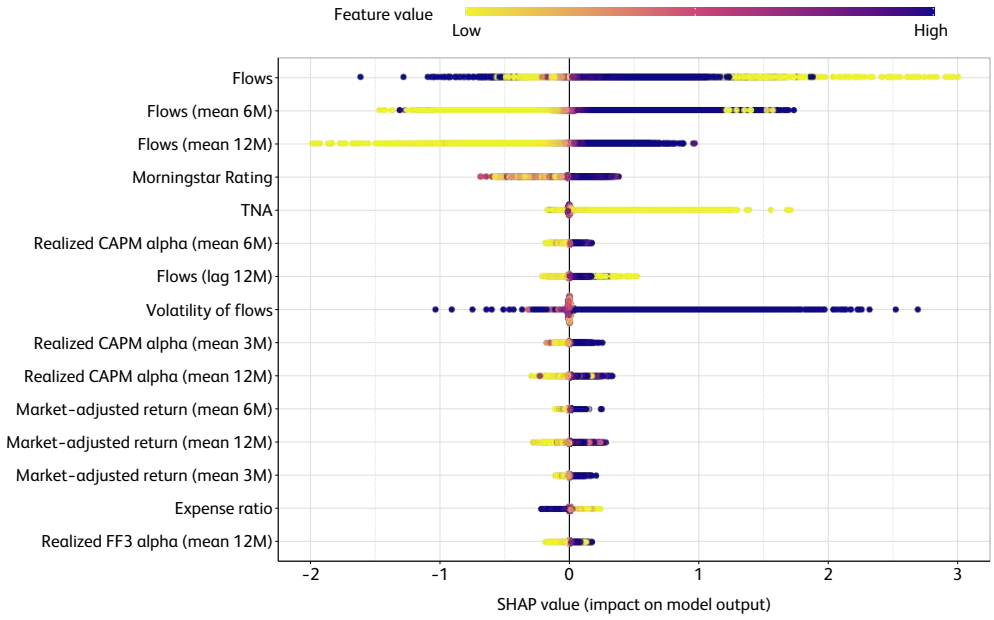


Figure I.4: **Directional impact of the characteristics for random forest over time**

This figure shows a beeswarm SHAP plot indicating the directional impact of the 15 most important characteristics identified by their mean absolute SHAP value for consecutive two-year periods and a one-month forecasting horizon for random forest. Predictors are sorted in descending order with the most important variables at the top, Each dot on the chart corresponds to one SHAP value for a prediction and feature. The magnitude of feature values is color-coded from yellow (low) to purple (high). The points are distributed horizontally according to their SHAP values which measure each feature's directional impact on the predictive model outcome (flow prediction).

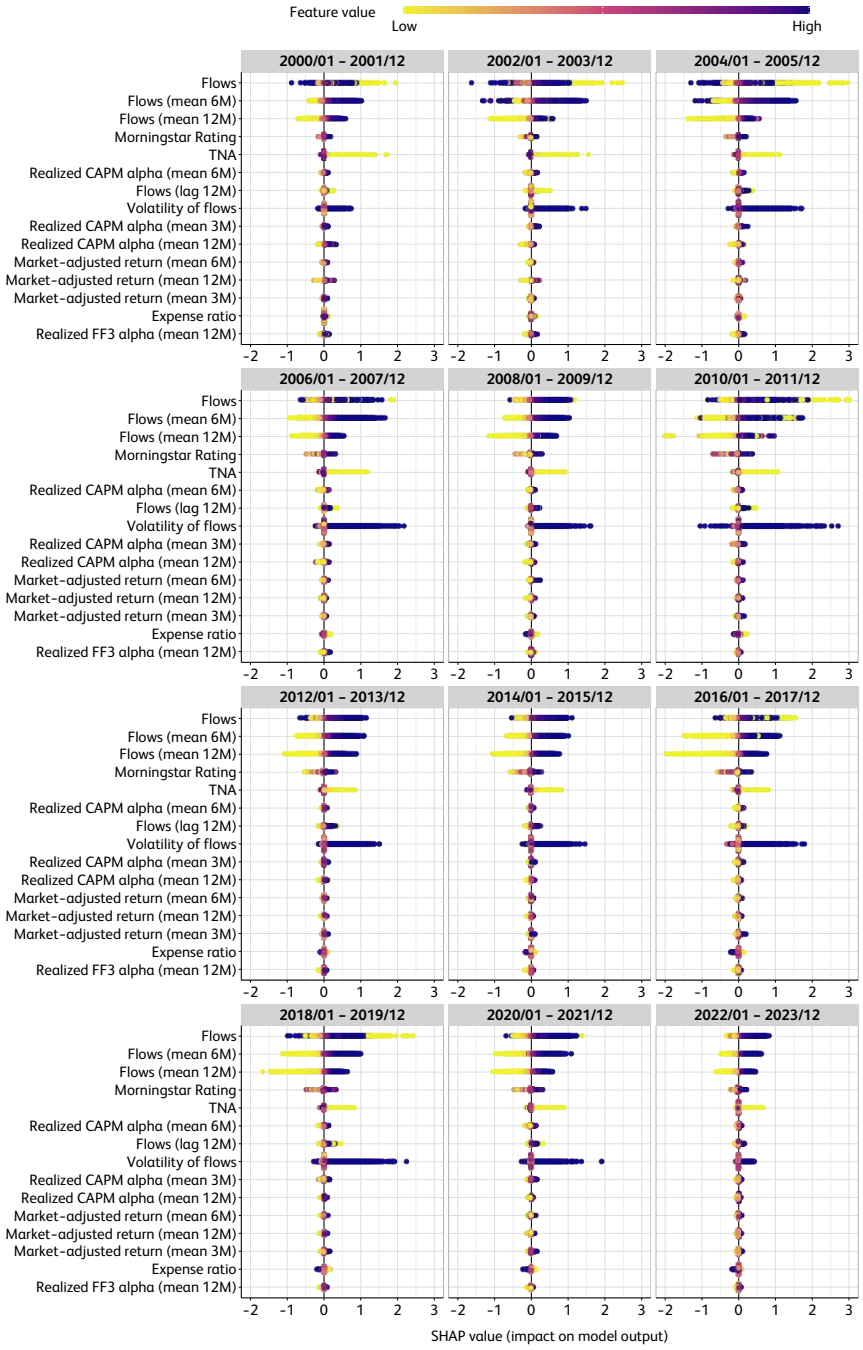


Figure I.5: **Functional form of past fund flows**

This figure plots the nonlinear (convex) relationship for each observation between past monthly fund flows for the whole out-of-sample period (January 2000 to December 2023) and a one-month forecasting horizon for random forest.

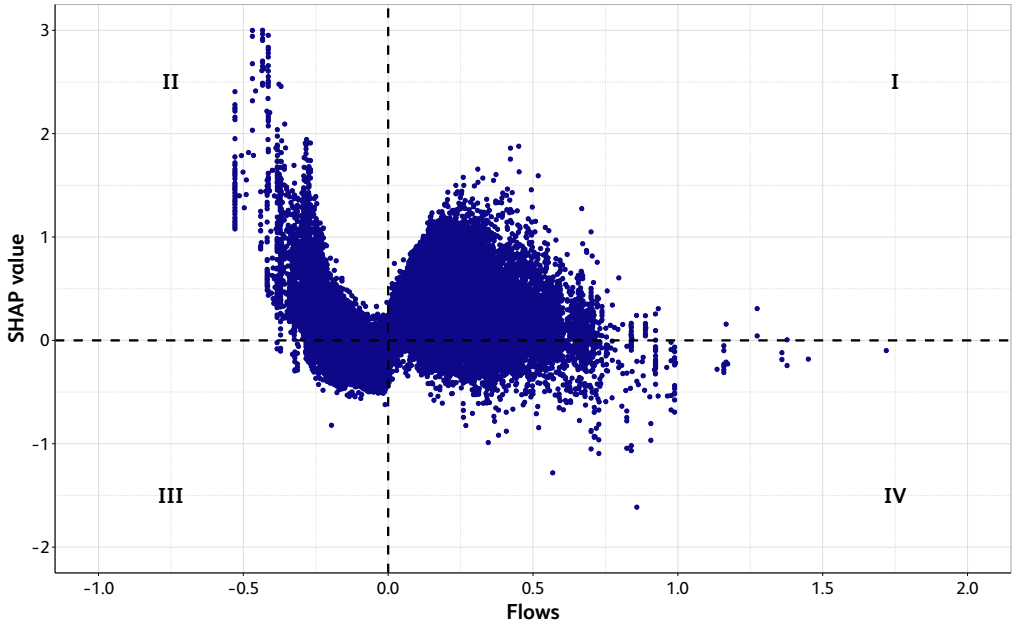


Table I.11: **Mean absolute SHAP interaction values for random forest and gradient boosting**

This table reports the mean absolute SHAP interaction values for all monthly forecasting periods for the ten most important interactions for random forest (Panel A) based on monthly random samples (10% of the share classes) and gradient boosting (Panel B) using the full sample. SHAP interaction values are calculated over the out-of-sample period from January 2000 to December 2023.

Panel A - random forest			
Rank	Characteristic I	Characteristic II	Mean
1	Flows	Flows (mean 6M)	0.0301
2	Flows (mean 6M)	Flows (mean 12M)	0.0170
3	Flows	Flows (mean 12M)	0.0140
4	Flows (mean 6M)	Morningstar Rating	0.0076
5	Flows	Morningstar Rating	0.0076
6	Flows (mean 6M)	Volatility of Flows	0.0047
7	Flows	TNA	0.0037
8	Flows	Volatility of Flows	0.0035
9	Flows (mean 12M)	Volatility of Flows	0.0034
10	Flows (mean 6M)	TNA	0.0030
Panel B - gradient boosting			
Rank	Characteristic I	Characteristic II	Mean
1	Flows	Flows (mean 6M)	0.0256
2	Flows	Flows (mean 12M)	0.0088
3	Flows (mean 6M)	Flows (mean 12M)	0.0081
4	Flows	Morningstar Rating	0.0079
5	Flows (mean 6M)	Morningstar Rating	0.0061
6	TNA	Value added	0.0057
7	Flows	TNA	0.0051
8	Expense ratio	TNA	0.0041
9	TNA	Volatility of Flows	0.0039
10	Morningstar Rating	TNA	0.0037

Figure I.6: **Nonlinear impact of the Morningstar rating**

This figure plots the five rating categories (stars) of the Morningstar rating and their corresponding SHAP values for the whole out-of-sample period (January 2000 to December 2023) and a one-month forecasting horizon for random forest. The Morningstar rating is based on a scale of one to five stars, with one being the worst and five being the best.

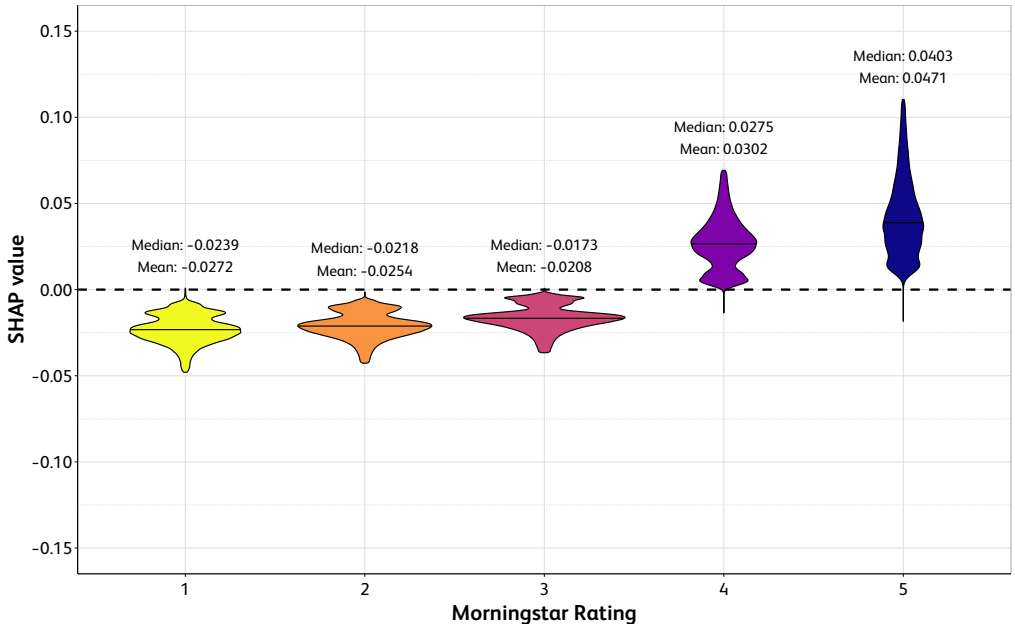


Figure I.7: **Interaction effect of past fund flows and Morningstar rating**

This figure plots the nonlinear (convex) relationship for each observation between past fund flows (predictor) and the associated SHAP value as well as the interaction with the Morningstar rating for the whole out-of sample period (January 2000 to December 2023) and a one-month forecasting horizon for random forest.

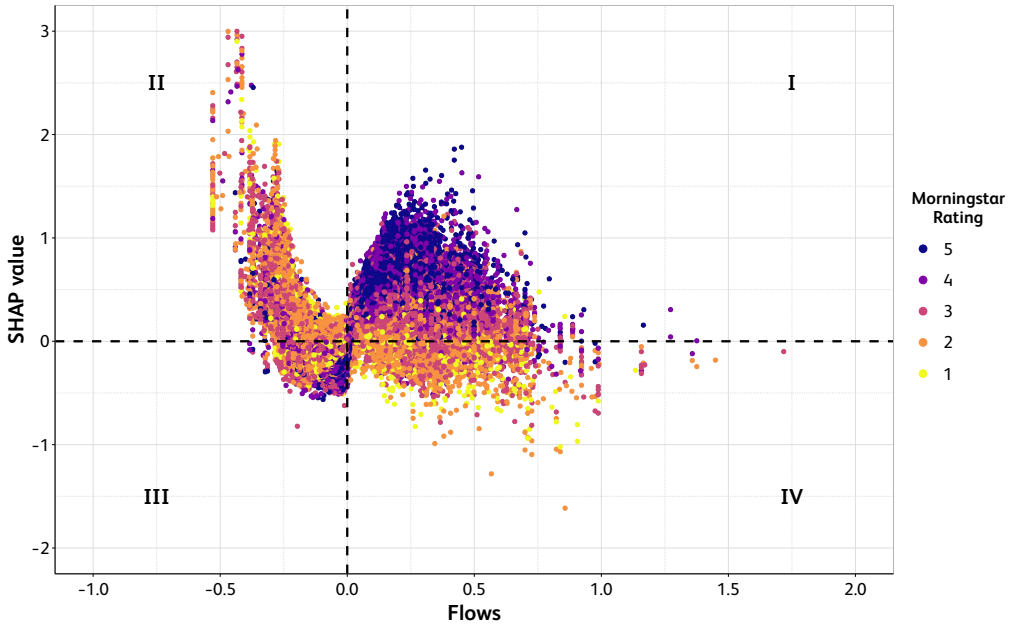


Table I.12: **Monthly out-of-sample alpha (in %) of the top and bottom decile (1st percentile) portfolios**

Panel A (Panel B) reports the monthly out-of-sample alpha of the equally weighted top and bottom decile (1st percentile) portfolios and the corresponding long-short portfolio of the predicted flow distribution based on a one-month ahead forecasts. Additionally, in both panels, a comparison with a naïve strategy (equally weighted portfolio of all share classes) is provided. The out-of-sample alphas are obtained by regressing the log excess return (long-short log return) of the top and bottom (long-short) portfolios on four different risk-factor models (in log): the CAPM, the Fama and French (1993) three-factor model (FF3), the Carhart (1997) four-factor model (FF4), the Fama and French (2015) five-factor model (FF5), and the FF5 model augmented with the momentum factor of Carhart (1997) (FF6). The out-of-sample period spans from January 2000 to December 2023 (288 observations). Newey and West (1987) robust standard errors with 12 lags (DeMiguel et al., 2023; Kaniel et al., 2023; Fausch et al., 2025a) are reported in parentheses. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5%, and 1% level, respectively.

Panel A - top and bottom decile portfolios

	Top 10%:					**Bottom 10%:**					**Long-Short:**				
	CAPM	FF3	FF4	FF5	FF6	CAPM	FF3	FF4	FF5	FF6	CAPM	FF3	FF4	FF5	FF6
Equally weighted	-0.05	-0.08*	-0.08*	-0.08*	-0.08*	-0.05	-0.08*	-0.08*	-0.08*	-0.08*					
	(0.06)	(0.05)	(0.05)	(0.04)	(0.04)	(0.06)	(0.05)	(0.05)	(0.04)	(0.04)					
OLS - FF4	0.04	0.00	-0.03	0.02	0.02	-0.20*	-0.22**	-0.19*	-0.17**	-0.17**	0.23*	0.22*	0.16	0.19*	0.19**
	(0.08)	(0.06)	(0.05)	(0.08)	(0.06)	(0.12)	(0.11)	(0.10)	(0.08)	(0.08)	(0.13)	(0.12)	(0.11)	(0.10)	(0.08)
Elastic Net	0.03	-0.01	-0.04	0.02	0.02	-0.22*	-0.25**	-0.21*	-0.20**	-0.20**	0.25	0.24	0.17	0.22*	0.22**
	(0.09)	(0.07)	(0.06)	(0.08)	(0.07)	(0.13)	(0.12)	(0.11)	(0.09)	(0.08)	(0.16)	(0.15)	(0.13)	(0.12)	(0.09)
Decision Tree	0.00	-0.03	-0.05	-0.02	-0.02	-0.11	-0.13*	-0.12	-0.13**	-0.13**	0.11	0.11	0.07	0.11	0.11*
	(0.07)	(0.05)	(0.05)	(0.06)	(0.05)	(0.10)	(0.08)	(0.08)	(0.06)	(0.06)	(0.10)	(0.09)	(0.08)	(0.07)	(0.06)
Random Forest	0.06	0.02	-0.00	0.01	0.01	-0.15	-0.18*	-0.16*	-0.19***	-0.19***	0.21*	0.20*	0.16	0.20**	0.20**
	(0.08)	(0.05)	(0.05)	(0.06)	(0.06)	(0.12)	(0.09)	(0.09)	(0.07)	(0.07)	(0.12)	(0.11)	(0.10)	(0.09)	(0.08)
Gradient Boosting	0.06	0.02	0.00	0.02	0.02	-0.18	-0.21**	-0.18*	-0.19***	-0.19***	0.23*	0.23*	0.18*	0.21**	0.21***
	(0.08)	(0.06)	(0.05)	(0.06)	(0.06)	(0.12)	(0.10)	(0.10)	(0.07)	(0.07)	(0.13)	(0.12)	(0.11)	(0.09)	(0.08)
Neural Network (I)	0.04	0.01	-0.02	0.00	0.00	-0.18	-0.21**	-0.19*	-0.20***	-0.20***	0.23*	0.22*	0.17	0.20**	0.20**
	(0.08)	(0.06)	(0.05)	(0.07)	(0.06)	(0.11)	(0.10)	(0.10)	(0.07)	(0.07)	(0.14)	(0.13)	(0.11)	(0.10)	(0.09)
Neural Network (II)	0.03	-0.01	-0.03	0.01	0.01	-0.16	-0.19**	-0.18*	-0.19***	-0.19***	0.19	0.18	0.14	0.20**	0.20**
	(0.07)	(0.06)	(0.05)	(0.08)	(0.06)	(0.12)	(0.10)	(0.09)	(0.07)	(0.07)	(0.13)	(0.12)	(0.11)	(0.10)	(0.08)

Monthly out-of-sample alpha (in %) of the top and bottom decile (1st percentile) portfolios (continued)

Panel B - top and bottom 1st percentile portfolios

	Top 1%:					Bottom 1%:					Long-Short:				
	CAPM	FF3	FF4	FF5	FF6	CAPM	FF3	FF4	FF5	FF6	CAPM	FF3	FF4	FF5	FF6
Equally weighted	-0.05	-0.08*	-0.08*	-0.08*	-0.08*	-0.05	-0.08*	-0.08*	-0.08*	-0.08*					
	(0.06)	(0.05)	(0.05)	(0.04)	(0.04)	(0.06)	(0.05)	(0.05)	(0.04)	(0.04)					
OLS - FF4	0.03	-0.02	-0.06	-0.05	-0.05	-0.30*	-0.32*	-0.28*	-0.26**	-0.26**	0.33	0.30	0.21	0.21	0.21
	(0.12)	(0.10)	(0.09)	(0.11)	(0.09)	(0.17)	(0.17)	(0.15)	(0.12)	(0.11)	(0.21)	(0.21)	(0.18)	(0.18)	(0.15)
Elastic Net	0.02	-0.03	-0.08	-0.07	-0.07	-0.32	-0.34	-0.28	-0.27	-0.27*	0.35	0.31	0.20	0.20	0.20
	(0.13)	(0.11)	(0.10)	(0.13)	(0.11)	(0.24)	(0.24)	(0.21)	(0.17)	(0.16)	(0.29)	(0.30)	(0.25)	(0.25)	(0.21)
Decision Tree	0.22	0.17	0.15	0.13	0.13	-0.18	-0.20	-0.18	-0.18*	-0.18*	0.40**	0.37**	0.32**	0.30**	0.30**
	(0.15)	(0.11)	(0.11)	(0.10)	(0.10)	(0.14)	(0.12)	(0.12)	(0.10)	(0.10)	(0.16)	(0.15)	(0.15)	(0.13)	(0.12)
Random Forest	0.11	0.05	0.02	0.02	0.02	-0.22*	-0.25**	-0.23**	-0.25***	-0.25***	0.33**	0.30**	0.25*	0.27**	0.27**
	(0.12)	(0.08)	(0.08)	(0.09)	(0.07)	(0.12)	(0.10)	(0.10)	(0.08)	(0.08)	(0.14)	(0.15)	(0.13)	(0.13)	(0.11)
Gradient Boosting	0.13	0.09	0.06	0.07	0.07	-0.25**	-0.28**	-0.26**	-0.25***	-0.25***	0.38***	0.36***	0.32**	0.33***	0.33***
	(0.10)	(0.07)	(0.07)	(0.08)	(0.07)	(0.12)	(0.11)	(0.11)	(0.09)	(0.09)	(0.14)	(0.13)	(0.12)	(0.11)	(0.10)
Neural Network (I)	0.12	0.06	0.03	0.06	0.06	-0.23*	-0.26**	-0.24*	-0.22**	-0.22**	0.35**	0.32**	0.27*	0.28**	0.28**
	(0.12)	(0.08)	(0.08)	(0.08)	(0.07)	(0.14)	(0.13)	(0.12)	(0.10)	(0.10)	(0.16)	(0.15)	(0.14)	(0.13)	(0.12)
Neural Network (II)	0.07	0.02	-0.01	0.04	0.04	-0.28**	-0.31***	-0.29***	-0.29***	-0.29***	0.35**	0.33**	0.28**	0.33**	0.33***
	(0.11)	(0.09)	(0.08)	(0.09)	(0.08)	(0.13)	(0.11)	(0.10)	(0.08)	(0.08)	(0.14)	(0.14)	(0.13)	(0.14)	(0.12)

Table I.13: **Monthly out-of-sample alpha (in %) based on prediction accuracy sorted quintile portfolios**

This table reports the portfolio alpha for various factor models (CAPM, FF3, FF4, FF5, FF6) using monthly log excess returns (long-short log return). Portfolios are formed by sorting all funds, every month (t) into quintiles by R^2_{OOS}. In the following month ($t+1$), we compute the monthly quintile portfolio (excess) returns for each portfolio of funds. The process is repeated every month so that we obtain a time-series of portfolio (excess) returns which are regressed on five different risk-factor models: the CAPM, the Fama and French (1993) three-factor model (FF3), the Carhart (1997) four-factor model (FF4), the Fama and French (2015) five-factor model (FF5), and the FF5 model augmented with the momentum factor of Carhart (1997) (FF6). The sample period spans from February 2000 to December 2023 (287 observations). Newey and West (1987) robust standard errors with 12 lags (DeMiguel et al., 2023; Kaniel et al., 2023; Fausch et al., 2025a) are reported in parentheses. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5%, and 1% level, respectively.

	OLS					**Random Forest**				
	CAPM	FF3	FF4	FF5	FF6	CAPM	FF3	FF4	FF5	FF6
Bottom	-0.048	-0.073*	-0.078*	-0.075*	-0.076*	-0.019	-0.046	-0.053	-0.049	-0.050
	(0.056)	(0.041)	(0.041)	(0.042)	(0.040)	(0.064)	(0.045)	(0.044)	(0.049)	(0.046)
Quintile 2	-0.024	-0.049	-0.056	-0.054	-0.055	-0.015	-0.042	-0.047	-0.044	-0.045
	(0.065)	(0.045)	(0.045)	(0.048)	(0.045)	(0.071)	(0.047)	(0.047)	(0.050)	(0.049)
Quintile 3	-0.008	-0.037	-0.039	-0.040	-0.040	-0.034	-0.060	-0.062	-0.066*	-0.066*
	(0.075)	(0.049)	(0.050)	(0.052)	(0.051)	(0.061)	(0.040)	(0.041)	(0.039)	(0.038)
Quintile 4	-0.033	-0.059	-0.057	-0.069	-0.069	-0.040	-0.066	-0.065	-0.074*	-0.074*
	(0.068)	(0.044)	(0.045)	(0.043)	(0.043)	(0.066)	(0.045)	(0.045)	(0.041)	(0.041)
Top	-0.077	-0.100**	-0.095**	-0.101***	-0.101***	-0.081	-0.103**	-0.098**	-0.107***	-0.107***
	(0.059)	(0.042)	(0.042)	(0.033)	(0.033)	(0.060)	(0.042)	(0.043)	(0.036)	(0.037)
Long-Short (Bottom-Top)	0.026	0.025	0.015	0.024	0.024	0.061***	0.056**	0.044*	0.057*	0.056**
	(0.021)	(0.022)	(0.020)	(0.026)	(0.020)	(0.023)	(0.026)	(0.024)	(0.032)	(0.026)

Table I.14: **Fund flow prediction accuracy and performance**

This table presents estimates from panel regressions of various measures of mutual fund performance (CAPM, FF3, FF4, FF5, FF6 alpha) in $t+1$ on a dummy variable (R^2_{OOS} dummy) which equals 1 if a fund j is in the top quintile (highest forecasting accuracy) and equals 0 if a fund is in the bottom quintile (lowest forecasting accuracy) according to their monthly $R^2_{OOS,j}$ in period t based on random forest. Controls include the respective lagged performance measure, the log of fund size (TNA), the log of fund age, the expense ratio, and fund and month fixed effects. The sample period spans from February 2000 to December 2023. Standard errors are clustered at the fund level and reported in parentheses. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5%, and 1% level, respectively.

	CAPM	FF3	FF4	FF5	FF6
R^2_{OOS} dummy	-0.026*	-0.040***	-0.031**	-0.036***	-0.028**
	(0.015)	(0.013)	(0.012)	(0.013)	(0.013)
$TNA_{i,t}$	-0.185***	-0.112***	-0.120***	-0.099***	-0.108***
	(0.013)	(0.009)	(0.009)	(0.009)	(0.009)
$Age_{i,t}$	-0.039	0.029	0.034	0.018	0.038
	(0.054)	(0.045)	(0.043)	(0.047)	(0.045)
Turnover ratio$_{i,t}$	-0.156***	-0.127**	-0.116**	-0.120**	-0.104*
	(0.056)	(0.056)	(0.056)	(0.054)	(0.053)
Expense ratio$_{i,t}$	-9.408	-9.846**	-10.999**	-9.242*	-8.987*
	(6.388)	(4.964)	(5.044)	(5.070)	(5.279)
Performance$_{i,t}$	2.563***	-1.191*	-0.707	0.144	-0.565
	(0.616)	(0.708)	(0.833)	(0.837)	(0.984)
Adjusted R^2	0.004	0.002	0.002	0.002	0.002
Observations	148,561	148,561	148,561	148,561	148,561
Month FE	Yes	Yes	Yes	Yes	Yes
Fund FE	Yes	Yes	Yes	Yes	Yes

Appendix to

"Machine Learning Mutual Fund Flows"

by

Jürg Fausch Moreno Frigg Stefan Ruenzi Florian Weigert

Appendix I.A. Process of matching CRSP with Morningstar

Table I.A1: **Detailed matching procedure between CRSP and Morningstar share classes**

This table documents the sequential steps used to merge mutual fund share class data from CRSP and Morningstar Direct (MSTAR). Starting with 14,248 CRSP share classes, we first eliminate missing or duplicate CUSIPs, followed by CUSIP-based identification in Morningstar. A two-step matching process is then performed using inception dates and validated by return-based consistency checks. This process yields 13,376 matched share classes and identifies 872 unmatched cases.

Step		n	Matched	Not matched
(1)	CRSP starting sample:	= 14,248		
(2)	No CUSIP provided by CRSP:	-319		319
(2)	Number of duplicated CUSIPs in CRSP:	-124		124
	Possible share classes to be matched:	= 13,805		
(3)	CUSIPs not found in MSTAR:	-179		179
(3)	Number of duplicated CUSIPs in MSTAR:	-24		24
	Possible share classes to be matched:	= 13,602		
(4)	Matching I	-11,800	11,800	
	Possible share classes to be matched:	= 1,802		
(5)	Maching II	-1,802	1,576	226
	Total	**= 0**	**= 13,376**	**= 872**

1. We begin by constructing the sample using the CRSP Survivor-Bias-Free US Mutual Fund database, applying the filters described in Subsection 2.1. This yields 14,248 distinct share classes.
2. Since our database merge primarily relies on the CUSIP of each share class, following Hillert, Niessen-Ruenzi, and Ruenzi (2025), we exclude 319 share classes without a CUSIP and 124 share classes with duplicate CUSIPs in the CRSP dataset, leaving 13,805 share classes available for potential matching.
3. At this stage, we import the 13,805 unique CUSIP numbers into Morningstar Direct. Morningstar fails to identify entries for 179 of these CUSIPs, and for an additional 24, it assigns multiple share classes to a single CUSIP. Consequently, we exclude these cases, resulting in 13,602 share classes available for potential matching.

4. For the 13,602 share classes, we collect data on the inception date from the two databases. Using the CUSIP and inception date, we successfully match 11,800 share classes. However, given the well-documented inconsistencies between the databases (Elton, Gruber, and Blake, 2001; Berk and van Binsbergen, 2015), we perform a robustness check using net return data from both databases to verify that the matched observations indeed represent the same share class.

   (a) For each share class and month, we retrieve net return data from both CRSP and Morningstar.
   (b) To enable a direct comparison, we first divide the Morningstar returns by 100 and then round the returns from both sources to five decimal places.
   (c) We then calculate the squared difference in CRSP and Morningstar returns for each share class and month.
   (d) Finally, for each share class, we compute the median of these squared differences to evaluate the magnitude of these differences.

   Of the 11,800 matched share classes, Morningstar lacks return data for 396, resulting in 11,404 share classes available for comparison. Among these, only 79 share classes (0.69%) exhibit a non-zero median squared difference. This confirms the robustness of our matching procedure. We attribute the small number of discrepancies to occasional data entry errors in either CRSP or Morningstar and therefore retain these observations in our analysis.
5. At this point, 1,802 share classes remain unmatched from the previous step. Upon inspection, we observe that for many of these cases, the inception dates in CRSP and Morningstar differ by only a few days. Nevertheless, to ensure accurate matching of identical share classes across both databases, we maintain the stringent criteria outlined in the previous step. By repeating this procedure, we successfully match an additional 1,576 share classes, for which the median squared difference in returns equals zero.

Appendix I.B. Machine learning methods

2.1. Elastic net

A common approach to mitigate the overfitting issue is achieved by adding a penalty term to the OLS loss function in Equation I.5:

$$\min_{\theta} \mathcal{L}(\theta; \cdot) = \min_{\theta} [\mathcal{L}(\theta) + \phi(\theta; \cdot)], \tag{I.11}$$

where $\phi(\theta; \cdot)$ is the penalty on θ. We follow Zou and Hastie (2005) and use the elastic net penalty, which consists of two regularization terms and takes the form

$$\phi(\theta; \lambda, \rho) = \lambda (1 - \rho) \sum_{j=1}^{P} |\theta_j| + \frac{1}{2} \lambda \rho \sum_{j=1}^{P} \theta_j^2. \tag{I.12}$$

The elastic net has two positive hyperparameters, $\lambda > 0$ and $\rho > 0$, which are determined separately by cross-validation. The loss function in Equation I.11 reduces to OLS when $\lambda = 0$. $\rho = 0$ corresponds to the Least Absolute Sum of Squares (LASSO) operator and uses the l_1 norm or absolute value penalties for penalized regression. LASSO imposes sparsity on the model specification in the sense that a subset of the parameters θ is exactly zero and can be used for variable selection. $\rho = 1$ corresponds to a ridge regression model, which relies on l_2-norm parameter penalization, which shrinks all coefficients closer to zero, but does not impose exact zeros anywhere. For values of $\rho \in (0, 1)$, both shrinkage and selection are imposed on the model. Moreover, the elastic net has good properties in handling highly-correlated predictors (Zou and Hastie, 2005; Diebold and Shin, 2019).

2.2. Decision tree

Decision trees are flexible, nonparametric models that split the feature space into K partitions based on one of the predictive variables:

$$g(z_{i,t}; \theta, K, L) = \sum_{k=1}^{K} \theta_{t-1}^{(k)} 1_{\{z_{i,t} \in C_k(L)\}} \tag{I.13}$$

where K denotes the number of terminal nodes, L is the depth, $C_k(L)$ is one of the K partitions of the data, $1_{\{.\}}$ is an indicator function and $\theta_{t-1}^{(k)}$ is defined as the sample average of outcomes within the corresponding partition $k \in K$.

Decision trees are able to jointly incorporate categorical and numerical predictors, are invariant to monotonic transformations, and capture interactions and nonlinearities. Decision trees are grown using the CART algorithm of Breiman et al. (1984), and the decision on which predictor and value to use for a split is determined based on minimizing the l_2 norm. The algorithm recursively splits observations into smaller subsets (decision nodes) until no further split is possible. The predicted $flow$ for each leaf reflects the average of the realized $flow_{t+1}$ of the share classes in the training sample sorted into this leaf. Due to the complexity of the tree, a greedy approach tends to overfit the training data, and thus requires strong regularization to obtain better prediction results. To reduce the complexity of the decision tree, we apply a pre-pruning approach based on a given early stopping criterion. The stopping criterion is based on the maximum depth of the tree L that has, on average, the lowest mean squared forecast error in the data set used for cross-validation.

2.3. Random forest

Random forests are a modification of bootstrap aggregation or bagging (Breiman, 2001). The basic idea is to randomly draw $b \in B$ bootstrap samples, $\{(z_{i,t}, \alpha_{i,t+h}), (i,t) \in Bootstrap(b)\}$ from the original data set and grow a decision tree

$$g_b(z_{i,t}; \theta_{b,t-1}, K, L) = \sum_{k=1}^{K} \theta_{b,t-1}^{(k)} 1_{\{z_{i,t} \in C_k(L)\}}. \tag{I.14}$$

The final prediction of the random forest is given by the average of the outputs of all B decision trees resulting in an ensemble forecast:

$$g(z_{i,t}; K, L, B) = \frac{1}{B} \sum_{b=1}^{B} g_b(z_{i,t}; \theta_{b,t-1}, K, L). \tag{I.15}$$

To estimate $\theta_{b,t-1}^{(k)}$, we follow the algorithm of Breiman et al. (1984). The hyperparameters to be tuned are the depth of trees L, the number of predictors P considered as split variables at each node, the size of each bootstrapped sample b and the number of bootstrapped samples B to grow a decision tree.

2.4. Gradient boosting

Gradient boosting is a method that combines multiple oversimplified (shallow) decision trees, known as weak learners, into a single strong learner with potentially greater stability than a single complex tree (Hastie et al., 2009). It is based on an iterative procedure. In the first step, an oversimplified tree with a high forecast error ($flow_{i,t+h} - \widehat{flow}_{i,t+h}$) is computed to predict $\widehat{flow}_{i,t+h}$. As suggested by Hastie et al. (2009) the loss function to be optimized is based on the l_2 norm since the target value is continuous. In the next step, a second shallow tree is used to fit the forecast residuals from the first tree. The predictions from these two trees are then added together to form an ensemble prediction. To prevent the model from overfitting the forecast residuals, the forecast component from the second tree is shrunken by a factor $\nu \in (0, 1)$. This hyperparameter corresponds to the learning rate and determines the weight the ensemble gives to the most recent decision tree. Furthermore, the risk of overfitting is also controlled by the depth of the tree L. At each iteration step, $b \in B$, an additional shallow tree is fitted to the residuals from the preceding ensemble prediction based on the model with $b-1$ trees, and its residual forecast is added to the total ensemble forecast with the pre-defined shrinkage weight of ν, resulting in a stronger model. This iteration continues until there are a total of B trees in the ensemble.

More formally, starting with $\hat{g}_0(z_{i,t}) = 0$, for each boosting iteration b from 1 to B, for each $i = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$ the negative gradient of the loss function $l(\cdot, \cdot)$ based on the l_2 norm

$$\varepsilon_{i,t+h} \leftarrow -\frac{\partial l\left(flow_{i,t+h}, g\right)}{\partial g} | g = \hat{g}_{b-1}\left(z_{i,t}\right) \tag{I.16}$$

is computed and a shallow regression tree of depth L is fit to the training data $\{(z_{i,t}, \varepsilon_{i,t+h}) : \forall i, \forall t\}$

$$\hat{f}_b(z_{i,t}) \leftarrow g\left(z_{i,t}; \theta, L\right). \tag{I.17}$$

$\hat{g}_b(z_{i,t})$ is updated by adding a shrunken version of the new tree together with an update of the residuals:

$$\hat{g}_b\left(z_{i,t}\right) \leftarrow \hat{g}_{b-1}\left(z_{i,t}\right) + \nu \hat{f}_b(z_{i,t}), \tag{I.18}$$

$$\varepsilon_{i,t+h} \leftarrow \varepsilon_{i,t+h} - \nu \hat{f}_b(z_{i,t}). \tag{I.19}$$

The output of the boosted model is

$$\hat{g}_B\left(z_{i,t}; B, \nu, L\right) = \sum_{b=1}^{B} \nu \hat{f}_b(z_{i,t}). \tag{I.20}$$

The final result is thus an additive model of shallow trees. The three tuning parameters (L, ν, B) are determined by cross-validation. We rely on histogram-based gradient boosting because it is known to reduce computational time while maintaining high accuracy (Ke et al., 2017).

2.5. Feedforward neural network

Feedforward neural networks (FFNN) are very flexible and highly parameterized methods for solving complex machine learning problems due to their ability to entangle many overlapping layers of nonlinear predictor interactions. These algorithms are very often referred to as deep learning. In our empirical analysis, we follow Gu, Kelly, and Xiu (2020) and Kaniel et al. (2023), and rely on multi-layer feedforward neural networks. These consist of an input layer of raw predictors at least one hidden layer that interacts and nonlinearly transforms the predictors, and an output layer that aggregates hidden layers into a prediction (Svozil, Kvasnicka, and Pospichal, 1997).

In a FFNN, each neuron is connected to all neurons in the previous layer, and the connections are one-way from the input layer to the output layer. The number of units in the input layer is equal to the P-dimensional vector of predictor variables z_i defined in Subsection 2.2. The nonlinear transformation applied to the predictor variables is based on an element-wise activation function. More specifically, each neuron $k \in K$ in the hidden layer j transforms inputs from the previous hidden layer, $x_i^{(j-1)}$, into output according to:

$$x_k^{(j)} = f\left(\theta_{k,0}^{(j)} + \sum_{i=1}^{P} x_i{}^{(j-1)} \theta_{k,i}^{(j)}\right) \tag{I.21}$$

where $f(\cdot)$ represents a nonlinear activation function, θ is a P-dimensional parameter vector that includes an intercept $\theta_{k,0}^{(j)}$ (bias term) and one weight parameter for each predictor ($\theta_{k,i}^{(j)}$). Since multiple options for the nonlinear activation function exist, we aim to identify the optimal function for each hidden layer during the tuning process by considering rectified linear unit (ReLU), sigmoid and tanh. The chosen activation function will be implemented across all neurons within the corresponding hidden layer. A deep neural network that combines multiple layers uses

the outputs of one hidden layer as an input to the next hidden layer. In the final hidden layer, the results from each activation are then fed into the output layer, resulting in

$$g(z;\theta) = \theta_0 + \sum_{k=1}^{K} x_k \theta_k, \tag{I.22}$$

a linear regression model in the K activations, providing the ultimate prediction of $\widehat{flow}_{i,t+h}$.

The (weight) parameters in our FFNN are computed iteratively using the back-propagation algorithm *Adam* suggested by Kingma and Ba (2014). The algorithm is initialized by assigning random weights to each connection of neurons. In the learning process, after each epoch[1], the algorithm sequentially updates the weights and biases of the FFNN based on minimizing a mean squared error loss function. To find the best possible architecture of our network, we search for the optimal number of hidden layers, the optimal number of neurons per hidden layer, the optimal activation function per hidden layer, the dropout rate per hidden layer, the number of epochs the model is trained on and the learning rate[2] which determines the iterative update of weights based on minimizing the MSFE. By not deciding ex-ante on the nonlinear activation function, fixing the number of hidden layers or by choosing the number of neurons in each hidden layer according to the geometric pyramid rule (Masters, 1993) we allow for considerable flexibility in building the network architecture. In particular, we employ two types of FFNNs that solely differ in the number of hidden layers and number of neurons per hidden layer. Neural network I (NN I) permits a lower set hidden layers (between 1 and 3) and neurons per hidden layer (ranging from 2 to 32) while neural network II (NN II) is a more complex model with 3 to 10 hidden layers and 32 to 1,024 neurons per hidden layer. Due to the stochastic nature and similar to Kaniel et al. (2023) and Chen, Pelger, and Zhu (2024), all our neural networks are averaged over 10 model fits after finding the optimal hyperparameters.

[1]Epochs refer to the number of times the fitting algorithm passes through the full training set.

[2]We start with a learning rate of 0.003 and reduce it with a factor of 0.5 if the mean-squared forecast error (MSFE) does not decrease for two consecutive epochs.

Appendix I.C. Additional robustness check

Table I.C1: **Out-of-sample machine learning model comparison based on $R^2_{OOS,Null}$**

This table reports the out-of-sample performances measured by the out-of-sample R^2 statistic, based on a null forecast, for the applied machine learning methods for a one-month forecasting horizon for the full sample including all predictions and the monthly top (bottom) 10% of the share classes with the highest (lowest) flows. The statistical significance of the $R^2_{OOS,Null}$ measure is evaluated by the Clark and West (2007) statistic (two-sided test). The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5%, and 1% level, respectively. The out-of-sample period spans from January 2000 to December 2023.

	Out-of-sample R^2		
	Full sample	**Top 10%**	**Bottom 10%**
OLS - FF4	0.1285***	0.1232***	0.1389***
Elastic net	0.1179***	0.0951***	0.1195***
Decision tree	0.1271***	0.1449***	0.1172***
Random forest	0.1772***	0.2082***	0.1463***
Gradient boosting	0.1728***	0.2026***	0.1430***
Neural network (I)	0.1436***	0.1477***	0.1457***
Neural network (II)	0.1388***	0.1354***	0.1453***
Observations	1,106,802	110,686	110,686

Appendix I.D. SHAP interaction values

A major advantage of most modern machine learning methods is their ability to model a large number of potentially relevant interaction terms. While in the paper SHAP values were computed for each prediction (share class) in each month and for every predictor, extending this approach to SHAP interaction values seems impractical due to the drastically increased computational complexity. Specifically, the time complexity for computing SHAP values with our best performing model, random forest, is $O(TLD^2)$, where T represents the number of trees, L the maximum number of leaves in any tree, and D the maximum depth of any tree (Lundberg et al., 2020). In contrast, the time complexity for SHAP interaction values increases to $O(TMLD^2)$, where M denotes the number of features (Lundberg et al., 2020).

More specifically, in our application, with 64 predictor variables, the theoretical computation time for SHAP interaction values is 64 times higher than that for SHAP values. For the computationally most intensive iteration (iteration 278 out of 288 total iterations), the calculation of SHAP values takes approximately 10 hours using parallel computing on an Intel Xeon Platinum 8358 processor with 64 cores. This means that the calculation of SHAP interaction values for this iteration would theoretically take about 640 hours, or about 27 days, which seems rather impractical.

To address the computational challenges associated with the SHAP interaction values, we restricted the computations in each iteration (month) to a random subset of predictions. Recognizing that the choice of subset size introduces a degree of arbitrariness, we systematically evaluated its impact during the first ten iterations, when computational demands were less constraining. Specifically, we computed SHAP interaction values for subsets representing randomly 1%, 5%, 10%, and 20% of the share classes, alongside computations for the full set of share classes. For each subset and for each pair of interactions, we calculated the corresponding mean and standard deviation. To assess the robustness of these results, Welch t-tests (Welch, 1947) were employed to determine the proportion of interaction pairs for which the subset means differed significantly (at the 5% level) from those of the full sample of share classes. The results are presented in Table I.D1. As expected, the smaller subsets reveal a greater proportion of statistically significant differences compared to the full sample (100%). Specifically, for the 1% subset, 98 interaction pairs (4.86%) exhibit significant differences. This proportion decreases to 73 pairs (3.62%) and 45 pairs (2.23%) for the 5% and 10% subsets, respectively. For the 20% subset, the number of significant pairs further declines to 34 (1.69%). However, this

additional reduction of 0.54 percentage points between the 10% and 20% subsets is obtained at the expense of doubling the computational time. In this context a meaningful approach to reducing computational time while obtaining consistent SHAP interaction values is to rely on a random set containing 10% of the share classes. The computational time is then approximately 3 days.

Table I.D1: **Robustness analysis of SHAP interaction values based on subsets of varying size for random forest**

The table presents a comparison of the number and proportion of interaction pairs for which the mean SHAP interaction value in the various subsets differs significantly at the 5% level from the mean value in the full sample (100%) based on the first ten iterations (out of 288 iterations). The results are presented for subsets comprising 1%, 5%, 10%, and 20% of the share classes.

	Set			
	1% **100%**	**5%** **100%**	**10%** **100%**	**20%** **100%**
Number of interaction pairs	2,016	2,016	2,016	2,016
No difference in means	1,918	1,943	1,971	1,982
Differences in means	98	73	45	34
Differences in means (proportion)	0.0486	0.0362	0.0223	0.0169

Appendix I.E. Missing data

Table I.E1: **Percentage of missing values per predictor variable**

This table reports the percentage of missing values for each of the 64 predictor variables used in the analyses.

	Characteristic	NA in %		Characteristic	NA in %
(1)	Flows	1.48%	(33)	FF4 alpha	2.27%
(2)	Flows (mean 6M)	2.70%	(34)	FF4 alpha (mean 3M)	2.45%
(3)	Flows (mean 12M)	4.02%	(35)	FF4 alpha (mean 6M)	2.75%
(4)	Flows (lag 12M)	2.49%	(36)	FF4 alpha (mean 12M)	5.51%
(5)	Volatility of flows	4.47%	(37)	FF4 alpha (mean 18M)	8.59%
(6)	Month	0.00%	(38)	FF5 alpha	2.27%
(7)	Expense ratio	18.20%	(39)	FF5 alpha (mean 3M)	2.45%
(8)	Front-end load	76.16%	(40)	FF5 alpha (mean 6M)	2.75%
(9)	Back-end load	63.55%	(41)	FF5 alpha (mean 12M)	5.51%
(10)	Turnover ratio	18.55%	(42)	FF5 alpha (mean 18M)	8.59%
(11)	Age	0.00%	(43)	FF6 alpha	2.27%
(12)	Size of Management Team	0.00%	(44)	FF6 alpha (mean 3M)	2.45%
(13)	Morningstar rating	0.49%	(45)	FF6 alpha (mean 6M)	2.75%
(14)	Change in Morningstar rating	0.76%	(46)	FF6 alpha (mean 12M)	5.51%
(15)	Volatility of return	2.19%	(47)	FF6 alpha (mean 18M)	8.59%
(16)	TNA	1.24%	(48)	Factor loading RMRF (FF6)	2.19%
(17)	TNA (Fund)	1.24%	(49)	Factor loading SMB (FF6)	2.19%
(18)	MAR	0.01%	(50)	Factor loading HML (FF6)	2.19%
(19)	MAR (mean 3M)	0.17%	(51)	Factor loading RMW (FF6)	2.19%
(20)	MAR (mean 6M)	0.41%	(52)	Factor loading CMA (FF6)	2.19%
(21)	MAR (mean 12M)	0.90%	(53)	Factor loading MoM (FF6)	2.19%
(22)	MAR (mean 18M)	1.39%	(54)	Value added	18.63%
(23)	CAPM alpha	2.27%	(55)	Adjusted R^2 (FF6)	2.19%
(24)	CAPM alpha (mean 3M)	2.45%	(56)	Short-term interest rate	0.00%
(25)	CAPM alpha (mean 6M)	2.75%	(57)	Long-term yield	0.00%
(26)	CAPM alpha (mean 12M)	5.51%	(58)	Term spread	0.00%
(27)	CAPM alpha (mean 18M)	8.59%	(59)	Credit spread	0.00%
(28)	FF3 alpha	2.27%	(60)	Stock market return	0.00%
(29)	FF3 alpha (mean 3M)	2.45%	(61)	Volatility of stock market return	0.00%
(30)	FF3 alpha (mean 6M)	2.75%	(62)	VIX	0.00%
(31)	FF3 alpha (mean 12M)	5.51%	(63)	Risk-aversion index	0.00%
(32)	FF3 alpha (mean 18M)	8.59%	(64)	Uncertainty index	0.00%

* Note that as discussed in Subsection 2.3, if no information about front-end or back-end loads is provided in CRSP we impute this variable with 0.

Appendix I.F. REFORMS checklist for ML-based science

Table I.F1: **REFORMS Checklist for ML-based Science**

This table reports the checklist for ML-based science suggested by Kapoor et al. (2024), consisting of the module, the respective item, and the reference in either (i) the paper or (ii) the GitHub repository.

Module	Item	Reference Paper/GitHub
Study goals	1a. State the population or distribution about which the scientific claim is made.	Section 1; Subsection 2.1; Subsection 2.4
	1b. Describe the motivation for choosing this population or distribution (1a.).	Section 1; Subsection 2.1
	1c. Describe the motivation for the use of ML methods in the study.	Section 1; Subsection 3.2
Computational reproducibility	2a. Describe the dataset used for training and evaluating the model and provide a link or DOI to uniquely identify the dataset	Subsection 3.3; GitHub
	2b. Provide details about the code used to train and evaluate the model and produce the results reported in the paper along with link or DOI to uniquely identify the version of the code used.	Subsection 3.2; GitHub
	2c. Describe the computing infrastructure used.	GitHub
	2d. Provide a README file which contains instructions for generating the results using the provided dataset and code.	GitHub
Data quality	3a. Describe source(s) of data, separately for the training and evaluation datasets (if applicable), along with the time when the dataset(s) are collected, the source and process of ground-truth annotations, and other data documentation.	Section 2; GitHub
	3b. State the distribution or set from which the dataset is sampled (i.e., the sampling frame).	Subsection 2.1
	3c. Justify why the dataset is useful for the modeling task at hand.	Section 1; Subsection 2.1
	3d. State the outcome variable of the model, along with descriptive statistics (split by class for a categorical outcome variable) and its definition.	Subsection 2.2; Equation I.1; Subsection 2.4; Table I.1
	3e. State the sample size and outcome frequencies.	Subsection 2.4; Table I.1
	3f. State the percentage of missing data, split by class for a categorical outcome variable.	Table I.E1
	3g. Justify why the distribution or set from which the dataset is drawn (3b.) is representative of the one about which the scientific claim is being made (1a.)	Section 2
Data preprocessing	4a. Describe whether any samples are excluded with a rationale for why they are excluded.	Subsection 2.4
	4b. Describe how impossible or corrupt samples are dealt with.	Not applicable
	4c. Describe all transformations of the dataset from its raw form (3a.) to the form used in the model, for instance, treatment of missing data and normalization — preferably through a flow chart.	Subsection 2.4
Modeling	5a. Describe, in detail, all models trained.	Subsection 3.2; Section I.B
	5b. Justify the choice of model types implemented.	Section 1; Subsection 3.2; Section I.B
	5c. Describe the method for evaluating the model(s) reported in the paper, including details of train- test splits or cross-validation folds.	Subsection 3.3
	5d. Describe the method for selecting the model(s) reported in the paper.	Subsection 3.3
	5e. For the model(s) reported in the paper, specify details about the hyperparameter tuning.	Subsection 3.3
	5f. Justify that model comparisons are against appropriate baselines.	Subsection 3.1; Subsection 3.3

REFORMS Checklist for ML-based Science (continued)

Module	Item	Reference Paper/GitHub
Study goals	1a. State the population or distribution about which the scientific claim is made.	Section 1; Subsection 2.1; Subsection 2.4
Data leakage	6a. Justify that preprocessing (Module 4) and modeling (Module 5) steps only use information from the training dataset (and not the test dataset).	Subsection 2.4; Subsection 3.3
	6b. Describe methods used to address dependencies or duplicates between the training and test datasets (e.g., different samples from the same patients are kept in the same dataset partition)	Subsection 3.3
	6c. Justify that each feature or input used in the model is legitimate for the task at hand and does not lead to leakage.	Section 2
Metrics and uncertainty	7a. State all metrics used to assess and compare model performance (e.g., accuracy, AUROC etc.). Justify that the metric used to select the final model is suitable for the task.	Subsection 3.3
	7b. State uncertainty estimates (e.g., confidence intervals, standard deviations), and give details of how these are calculated.	Not applicable (see 3g.)
	7c. Justify the choice of statistical tests (if used) and a check for the assumptions of the statistical test.	Subsection 3.3
Generalizability and limitations	8a. Describe evidence of external validity.	Not included
	8b. Describe contexts in which the authors do not expect the study's findings to hold.	Not included

Chapter II

Does machine learning really help to select mutual funds with positive alpha?

(with Jürg Fausch, Thomas Johann, Emil Mussbach, Christian Westheide and Wolfgang Drobetz)

1. Introduction

The mutual fund industry has grown considerably over the last two decades. By the end of 2024, mutual funds in the US managed assets exceeding USD 28 trillion, compared to around USD 5.5 trillion in 1998 (Investment Company Institute, 2025). Through their investment decisions, these funds play a major role in allocating capital within the economy. In an attempt to outperform their benchmarks and generate value for investors, many of these mutual funds actively manage their portfolio in order to achieve positive risk-adjusted returns (alpha).

Given the importance of the mutual fund industry, assessing whether active portfolio management leads to superior investment performance has been a major focus of academic finance. While some studies provide empirical evidence that a (small) subset of fund managers possess the skill to outperform their benchmarks (see, e.g., Wermers, 2000; Barras, Scaillet, and Wermers, 2010; Fama and French, 2010; Kacperczyk, van Nieuwerburgh, and Veldkamp, 2014; Berk and van Binsbergen, 2015), a large body of literature shows that the average actively managed fund earns negative alpha after transaction costs, fees and other expenses (see, e.g., Sharpe, 1966; Jensen, 1968; Gruber, 1996; Ferreira et al., 2013). More recently, a new strand of literature has emerged, focusing on the application of machine learning (ML) methods to ex-ante identify funds that are likely to outperform. In this context, DeMiguel et al. (2023) and Kaniel et al. (2023) use ML algorithms to forecast the out-of-sample[1] risk-adjusted performance of US equity mutual funds,

[1]The out-of-sample period is "pseudo" in that its observations are available to the researcher during the study. Nevertheless, since they are not used to train the model, the financial machine learning literature commonly refers to this testing sample as the "out-of-sample" period (e.g. Gu, Kelly, and Xiu, 2020; Bianchi, Büchner, and Tamoni, 2021; Bali et al., 2023; Fausch et al., 2025b).

exploiting potential nonlinearities and interactions between various fund characteristics and performance.

The results of DeMiguel, Gil-Bazo, Nogales, and Santos (2023), hereafter referred to as DGNS, imply that ML methods, such as gradient boosting and random forest, can be used to build tradable long-only portfolios of equity mutual funds that earn significant annualized out-of-sample alpha[2] of 2.69% net of all costs, relying solely on fund characteristics as model inputs. Kaniel et al. (2023) employ feedforward neural networks to forecast mutual fund alpha based on a large set of predictors, including stock characteristics, fund characteristics and macroeconomic variables. In contrast to DGNS, Kaniel et al. (2023) base their out-of-sample analysis on long-short portfolios and report a statistically significant alpha of 2.04%.[2] The finding of DGNS is remarkable as the positive net alpha is obtained using a tradable long-only portfolio, while most of the alpha in Kaniel et al. (2023) stems from the short leg (bottom decile of the predicted alpha distribution) of their non-tradable, long-short mutual fund portfolio. The results reported by DGNS have considerable implications, as they challenge prevailing notions on active mutual fund performance and suggest that investors can earn economically substantial risk-adjusted returns by investing in ML-generated long-only mutual fund portfolios.

In this study, we demonstrate that the conclusions presented by DGNS arise from a coding error that introduces a forward-looking bias into the portfolio weighting scheme. After correcting this bias in our reproduction[3], the annual alphas of the two best-performing machine learning algorithms, gradient boosting and random forest, decrease by 1.37 and 1.42 percentage points. Consequently, the alphas are no longer statistically significant at any conventional level. In addition to the bias in the portfolio weights, we identify a survivorship bias and uncover methodological issues in the computation of characteristic importance based on SHAP values. By fully replicating the study of DGNS and implementing the required corrections, we obtain results for gradient boosting and random forest that are weaker than those reported in the reproduction, again yielding no significant positive alpha for long-only portfolios of actively managed equity mutual funds. However, we are able to conceptually confirm the findings of Kaniel et al. (2023) by showing that long-short portfolios based on machine learning predictions can achieve significant

[2]The alpha is computed based on the Fama and French (2015) 5-factor model extended by the momentum factor of Carhart (1997).

[3]To avoid semantic ambiguity, we use the term reproduction to denote the validation of the results in DGNS using their original data (that they provide in a data repository) while correcting for the forward-looking bias and the term replication to denote the construction of an independent sample while adhering to their stated methodology but correcting for additional biases and inconsistencies in their study.

positive annual alpha. While gradient boosting yields an annual alpha of 3.05 percent, random forest achieves 3.00 percent. These results are largely driven by the inferior performance of the short leg (bottom decile). Nevertheless, we also find that ordinary least squares (OLS) and elastic net can separate high- from low-performing funds and deliver statistically significant long–short portfolio alphas that are slightly higher than those obtained by nonlinear ML methods. For the best-performing algorithm (elastic net), the annualized alpha of the long-short portfolio amounts to 3.58 percent.

Beyond replicating the findings of DGNS, the final part of our study extends the empirical analysis along three dimensions. First, we extend the sample through December 2024 and find that our results remain robust, with no ML algorithm yielding any significant long-only alpha. Second, acknowledging the long-term investment horizon of institutional investors (McCahery, Sautner, and Starks, 2016), we extend the forecasting horizon and corresponding holding period to 36 months. The results remain largely consistent, with significant alpha only observed for the short legs and the resulting long-short portfolios (3.06% per year for random forest). However, unlike for a 12-month holding period, linear models fail to deliver statistically significant positive long-short alphas. This suggests that the benefits of ML methods are most evident when the forecasting horizon extends beyond 12 months. Third, we implement a univariate portfolio-sorting framework based on the 17 mutual fund characteristics used as features in DGNS and our replication. This approach ranks funds annually by individual predictors based on their magnitude, allowing us to assess the predictive ability of single variables in forecasting future risk-adjusted returns. For a forecasting horizon of 12 and 36 months, the performance of certain single-sort portfolios is broadly comparable to that of ML methods, with differences below 10 bps per month.

The remainder of the paper is structured as follows: Section 2 outlines the impact of the forward-looking bias on the main findings of DGNS. In Section 3 and 4, we provide an independent replication of DGNS and extend the analysis along several dimensions. Section 5 concludes.

2. Reproduction of DeMiguel et al. (2023)

In this section, we reexamine the validity of the main findings reported in DGNS. In this context, we use the code and data provided by the authors in their replication

package, which ensures a consistent and reliable comparison.[4] In their paper, DGNS report portfolio results based on four risk-factor models: the Fama and French (1993) three-factor model augmented with the momentum factor (FF3+MOM) suggested by Carhart (1997); the Fama and French (2015) five-factor model (FF5); the FF5 model augmented with momentum (FF5+MOM) and the aggregate liquidity factor of Pástor and Stambaugh (2003) (FF5+MOM+LIQ). However, since their reported results are broadly similar across asset-pricing models, we follow DGNS and adopt the FF5+MOM specification as our main model for the remainder of the paper. In Table II.B1 of Appendix B, we demonstrate that our main conclusions remain unchanged when alternative factor models are employed.

We begin our analysis by noting that DGNS report an average monthly net alpha for an equally weighted portfolio (EW) of all share classes of -1.8 bps (-0.22% per year with a t-statistic of -0.41) that far exceeds the magnitudes usually reported in the mutual fund literature (see, e.g., Jensen, 1968; Malkiel, 1995; Gruber, 1996; Fama and French, 2010). To reconcile the discrepancy between the results reported in DGNS and the empirical literature, we examined their code provided in their replication package and uncovered a look-ahead bias in the construction of portfolio returns, as illustrated in Algorithm 1.

Algorithm 1 Compute Portfolio Returns (with look-ahead bias)

```
computePortfolioReturns <- function(w, fund.returns) {
  portf_ret <- array(NA, c(12, 1))
  w_matrix <- c()
  for(r in 1:12) {
    portf_ret[r] <- sum(w * fund.returns[, r], na.rm = TRUE)
    if(r < 12) {
      w <- w * (1 + fund.returns[, r+1]) %>%
        ifelse(is.na(.), 0, .)
      w <- w / sum(w)
      w_matrix <- cbind(w_matrix, w)
    }
  }
  return(list(portf_ret = portf_ret, w_matrix = w_matrix))
}
```

The bias arises because the portfolio weights at the start of period $t + 1$ are updated using the realized returns from period $t + 1$ (written in magenta in the

[4]The data repository used in this study refers to Version 3 published on October 9, 2023. The replication package of DGNS can be accessed here: https://data.mendeley.com/datasets/rpgb99m5zy/3.

code) rather than return information available at period t.[5] This procedure inadvertently incorporates future information into the weighting scheme, thereby introducing look-ahead bias and substantially overstating portfolio performance.

As shown in Algorithm 2, the correct computation of the updated portfolio weights requires multiplying the end-of-period t weights by the period's returns (followed by normalization such that the resulting weights sum to 1). Hence, the function of DGNS should correctly reference `fund.returns[, r]` rather than `fund.returns[, r+1]`. Apart from this adjustment, every other step of the analysis is left unchanged.

Algorithm 2 Compute Portfolio Returns (without look-ahead bias)

```
computePortfolioReturns <- function(w, fund.returns) {
  portf_ret <- array(NA, c(12, 1))
  w_matrix <- c()
  for(r in 1:12) {
    portf_ret[r] <- sum(w * fund.returns[, r], na.rm = TRUE)
    if(r < 12) {
      w <- w * (1 + fund.returns[, r]) %>%
        ifelse(is.na(.), 0, .)
      w <- w / sum(w)
      w_matrix <- cbind(w_matrix, w)
    }
  }
  return(list(portf_ret = portf_ret, w_matrix = w_matrix))
}
```

Column 1 of Table II.1 reports the published results of DGNS, column 2 presents the estimates based on their data repository (which contains added noise due to the proprietary nature of the CRSP data) and column 3 shows the estimates obtained from the repository after removing the look-ahead bias (reproduction).

A comparison of the published results from DGNS with those derived from the data repository suggests that the noise introduced into their dataset is negligible, as indicated by the nearly identical results in the first two columns.[6] However, correcting the coding error materially affects the empirical findings of DGNS and has

[5] For example, the weight of a given fund in the portfolio on the first trading day of February 1991 is computed by taking its January 1991 weight and adjusting it by the fund's February 1991 return, rather than its January 1991 return. According to the replication code, this issue occurs in all months except January for every year in the out-of-sample period.

[6] Rerunning the code from the data repository reproduces results that are very similar to those reported in DGNS for all asset pricing models except the FF3+MOM specification (see Table II.B1 in the appendix). For this model, the top-decile alphas from gradient boosting and random forest become statistically insignificant (an alpha of 0.105 with a t-stat of 1.38 for gradient boosting and an alpha of 0.098 with a t-stat of 1.18 for random forest). Although we cannot definitively identify the source of this discrepancy, we suspect that it arises from the use of wrong factors in

two main implications (column 3). First, the EW portfolio now exhibits a monthly alpha of -8.2 bps (-0.98% per year with a t-statistic of -2.10), closely aligning with prior evidence in the mutual fund literature. A comparable pattern emerges for the asset-weighted (AW) portfolio constructed from all share classes. Second, the adjustment substantially lowers the monthly alphas for both the machine learning methods and the benchmark OLS model. For gradient boosting and random forest, the net alphas decrease by 11.4 bps and 11.8 bps per month (1.37 and 1.42 percentage points per year), respectively, implying that the null hypothesis of zero true alpha can no longer be rejected at any common level of statistical significance (t-statistic of 1.14 and 1.29, respectively). Overall, these results suggest that the findings reported in DGNS do not hold once their code is corrected for the look-ahead bias in the portfolio-weight updating process. Instead, the alphas reported in their study arise from the unintended incorporation of future return information into the portfolio weighting scheme.

[Insert Table II.1 here]

Figure II.1 presents the cumulative out-of-sample alpha[7] over the out-of-sample period from 1991 to 2020 for the portfolios constructed using the top decile of the predicted alpha distribution. A comparison between the cumulative out-of-sample portfolio alpha reported by DGNS in panel (a) and the corresponding results from the reproduction in panel (b) reveals notable differences in their time-series dynamics. In particular, the cumulative alpha of DGNS exhibits a more pronounced increase in the early 2000s and a stronger subsequent upward movement between 2001 and 2012 relative to the reproduction. According to the result presented by DGNS, an investor implementing a strategy that selects the top decile of mutual funds based on random forest (gradient boosting) forecasts would have earned a cumulative abnormal return of 78 percent (69%). In contrast, for the reproduction, random forest and gradient boosting portfolios achieve cumulative net alphas of 34.77% and 27.50%, less than half the magnitude reported by DGNS.[8]

[Insert Figure II.1 here]

the FF3+MOM model. In particular, substituting RMW for HML in the FF3+MOM specification reconciles our estimates with the results reported in the DGNS data repository.

[7]As outlined in footnote 27 of their paper, DGNS compute monthly net alphas as the portfolio excess returns net of all costs each month minus the product of the factor realization in that month and the portfolio betas estimated over the whole out-of-sample sample period using the FF5 model augmented with momentum.

[8]The equally weighted and asset-weighted portfolios earn negative cumulative net alphas of -30.12% and -28.25%, respectively.

3. Replication of DeMiguel et al. (2023)

In addition to the coding-related forward-looking bias identified in DGNS and discussed in Section 2, we uncover several additional issues that may distort the interpretation of their findings. To confirm our findings from Section 2 and to address these concerns, this section provides a full replication of their analysis using the same data source, the CRSP Survivor-Bias-Free U.S. Mutual Fund Database, while closely following their stated methodology but correcting for the issues discussed. More specifically, we reconstruct the sample, replicate the forecasting procedure, reevaluate performance and reassess the characteristic importance and the underlying economic mechanism.

3.1. Replication design, methodology and data

In this subsection, we outline the replication design and provide an overview of the sample and predictors used in our analysis. We also describe the data-preprocessing steps, the machine-learning methods employed, the forecasting procedure and the performance-evaluation methodology.

3.1.1. Sample construction and mutual fund characteristics

The construction of the data set follows the selection criteria outlined in DGNS and focuses on US domestic-equity mutual funds at the share class level.[9] The sample includes institutional as well as retail share classes and spans from January 1980 to December 2020. Since the authors do not provide detailed information on how to select the sample of equity funds, we follow Kaniel et al. (2023) and Doshi, Elkamhi, and Simutin (2015) by using the CRSP objective code to identify these funds.[10] In line with DGNS, we restrict the analysis to share classes that charge no front-end or back-end loads to ensure that portfolio rebalancing does not incur any costs. If no information on loads is provided by CRSP, we follow Roussanov, Ruan, and Wei (2021) and treat any missing values as zero. In addition, to select the relevant share classes, we apply the same filters as DGNS, which are commonly used in the mutual fund literature. First, we restrict the sample to share classes

[9]Given the absence of detailed guidance in DGNS and the numerous design choices involved in building the sample, Section II.A of the Appendix presents a comprehensive overview of our procedure.

[10]In line with Doshi, Elkamhi, and Simutin (2015) we require that the CRSP objective code begins with the letters "ED" followed by a "C", "Y", or "S" and that the third and fourth letter are not equal to "YH" or "YS". In contrast to Doshi, Elkamhi, and Simutin (2015), we also include sector funds (third letter "S") to be consistent with DGNS.

of actively managed mutual funds, thereby excluding ETFs and passive mutual funds.[11] Second, we consider only share classes that invest more than 70 percent of their total net assets (TNA) in equities. Third, to avoid incubation bias, we exclude observations of a share class before it reaches 36 months of age and before the first observation with at least USD 5 million of TNA (Elton, Gruber, and Blake, 2001; Evans, 2010).

We follow DGNS (refer to their subsection 2.2 for an in-depth comprehension of the details) and construct the 17 share-class characteristics. Table II.2 reports descriptive statistics (mean, median, standard deviation and the number of class-month observations) for the replication sample for each characteristic. We find our sample size to be larger than that of DGNS despite following their reported filtering steps as closely as possible.[12] However, a comparison with the data used in DGNS reveals that, in terms of descriptive statistics, these characteristics are fairly similar.

[Insert Table II.2 here]

The main analysis is performed at the share class level and the sample period spans from January 1980 to December 2020.[13]

3.1.2. Data preprocessing

Prior to the application of machine learning methods to predict future mutual fund alpha, we have to preprocess the 17 mutual fund characteristics to generate the target and predictor variables. In this context, we use the same procedure as outlined in DGNS. We start by converting the sample from monthly to annual frequency. For each year, we compute annual realized alpha[14], value added and flows as the average of their monthly values multiplied by twelve. As monthly flow volatility is already defined for each year, we multiply it by the square root of twelve to annualize it. For the remaining characteristics, the values in December of each year are used. At this stage, DGNS state that each characteristic is standardized cross-sectionally. However, inspection of the dataset provided in their replication

[11] In line with DGNS, we use the index fund identifier from CRSP and, in cases where this identifier is missing, we refer to fund names following the approach suggested by Gil-Bazo and Ruiz-Verdú (2009).

[12] In the absence of data processing code and observations per each filtering step applied by DGNS, it appears impossible to identify the reasons for this difference. However, we would expect our larger sample size to benefit the performance of any machine learning algorithms.

[13] In Subsection 4.1 we extend the sample through the end of 2024 to assess the robustness of our findings using additional data.

[14] The monthly realized alpha is computed as the share class excess return minus the expected return based on the FF5+MOM model, with factor loadings estimated over a rolling window from month $t-36$ to month $t-1$ (see equation (2) in DGNS).

package indicates that the standardization is not performed separately within each cross-section, but rather using the entire panel of observations available through December 2020. Standardization within each cross-section is a reasonable choice, as is standardization within the entire training set. While DGNS incorrectly state that they do the former, their standardization within the entire dataset (not only training set) does not introduce additional bias and leads, in fact, to results equivalent to those obtained when standardizing the training set. This is due to the inherent characteristics of the methods and implementation details. First, tree-based methods (gradient boosting and random forest) and OLS are entirely insensitive to feature scaling. Second, while scaling matters for elastic net due to the regularization imposed on the estimated coefficients, the *R* package used by DGNS defaults to standardizing features within the training set and they do not overwrite this default behavior. Thus, their procedure effectively standardizes within the entire training set rather than within each cross-section and we adopt the same approach to remain as close as possible to their implementation while correcting actual coding errors.

Next, we define the target variable as the one-year-ahead realized share-class alpha. In DGNS, share classes containing no observation for the target variable are excluded.[15] However, excluding these share classes at the time of prediction introduces a look-ahead bias (survivorship bias), since investors cannot observe whether the fund continues to exist over the subsequent 12 months at the time the forecast is made. This approach has important implications: funds that cease operations during the year are excluded from the analysis. More specifically, as DGNS only compute predicted alphas for funds that remain active until the end of the prediction year, portfolio allocations are also solely based on surviving funds. Since portfolio construction then relies on information only available at the end of the year and as survival is positively correlated with performance, this procedure leads to an upward bias in estimated performance. To address this issue, we create a training set (in which we remove observations with a missing target value) and a prediction set, for which the target variable is not required and thus, these observations are preserved.[16]

Consistent with DGNS, the 17 characteristics we use as predictors are the following one-year-lagged standardized variables: realized alpha, alpha t-stat, TNA, expense ratio, age, flows, volatility of flows, manager tenure, value added, R^2 and the t-stats of the market, profitability, investment, size, value and momentum be-

[15] This conclusion is based on their dataset "scaled_data_data_JFE.csv", where the column for realized alpha (target variable) contains no imputed data, as indicated by the absence of zeros.

[16] To evaluate the importance of this error, we report results based on only fixing the weighting issue in Table II.C1.

tas. Figure II.2 shows the correlation matrix of the target and predictor variables. In line with DGNS, we find that the target variable exhibits low correlation with lagged predictors, while several predictor variables exhibit high correlations. In our sample, the strongest absolute correlation (0.56) is observed between market beta t-stat lagged and R^2 lagged. Overall, the magnitudes of the estimated correlation coefficients closely align with those reported in DGNS. The main exception is the correlation between lagged flows and lagged flow volatility, which is 0.26 in our sample compared with 0.61 in DGNS. Since DGNS do not provide the code for sample construction and data preprocessing in their replication package, it is not possible to further investigate this discrepancy.

[Insert Figure II.2 here]

3.1.3. Machine learning methods

The data are organized in a panel structure, with months indexed at $t = 1, 2, \ldots, T$ and share classes as $i = 1, 2, \ldots, N$. Following DGNS, the simple linear predictive regression model serves as a benchmark in the prediction of mutual fund performance. In addition, we follow DGNS and implement the same ML methods (elastic net, random forest and gradient boosting) to forecast fund performance. We train each of these models using fund characteristics until $t - 1$ and realized alpha in t of each year. To predict the alpha of each share class over an h-year forecasting horizon, we employ the corresponding fund characteristics of year t:

$$\alpha_{i,t+h} = f_t^h \left(z_{i,t} | \, \gamma \right), \tag{II.1}$$

where $\alpha_{i,t+h}$ denotes h-year ahead realized alpha for share class i (target variable) and f_t^h denotes the machine learning model that takes the set of fund characteristics z_t (feature values) with hyperparameter vector γ as input. Consistent with DGNS, the forecasting horizon h in the replication corresponds to one year.

3.1.4. Forecasting approach, cross-validation of hyperparameters and performance evaluation

Forecasting approach: The data set is divided into a training and prediction set. The first 10 years of data on one-year ahead realized alphas (from 1981 to 1990) and one-year-lagged fund characteristics (from 1980 to 1989) are used to train each algorithm. Values of fund characteristics in December 1990 (prediction set), which are not part of the training set, are used to predict fund performance

(alpha) in December 1991. For each successive year, the training set is expanded by one year, the ML algorithms and OLS are retrained on the expanded sample and a new prediction for the following year is made. This approach results in a predicted alpha distribution for each forecasting cycle.

Cross validation of hyperparameters: For each estimation window, the hyperparameters of the elastic net, random forest and gradient boosting are tuned using k-fold cross-validation (CV) based on grid search. As shown by Bergmeir, Hyndman, and Koo (2018) and Goulet Coulombe et al. (2022), k-fold CV performs favorably to alternative techniques that explicitly account for the time-series properties of the data. Following DGNS, we set $k = 5$ in this replication.[17]

Performance evaluation: The out-of-sample performance is evaluated using a portfolio approach combining the top decile share classes of the predicted alpha distribution for each calendar year into an equally weighted portfolio and tracking its net returns in the 12 months of the same year. In addition to DGNS, we also form portfolios in the bottom decile of the alpha distribution and a spanning portfolio (long-short). This allows for a direct comparison of our results with those reported by Kaniel et al. (2023). If a portfolio constituent has a missing return observation in a given month, its portfolio weight is redistributed equally among the remaining constituents with available return observations for that month. This procedure results in a time series of monthly out-of-sample net returns of the top and bottom decile portfolios from January 1991 to December 2020 (360 months). As in DGNS and the reproduction, the risk-adjusted out-of-sample performance of these portfolios is estimated by running a time-series regression of the 360 out-of-sample monthly portfolio excess returns (net returns minus the risk-free rate) on the FF5+MOM factor returns.[18]

3.2. Empirical results

This subsection presents the empirical results based on the approach described in Subsection 3.1, along with a comparison to the findings reported by DGNS

[17] Hyperparameter tuning for all ML models in DGNS follows the default tuning grids provided by the *caret* package in R. For random forest, the replication package fixes the number of trees at `ntree = 1000`. Consequently, *caret* fits multiple random forest models, each with 1000 trees, while varying only the `mtry` parameter (the number of candidate variables considered at each split). Because neither a custom tuning grid nor a `tuneLength` value is specified, *caret* uses its default `tuneLength = 3` and therefore generates a default tuning grid containing three `mtry` values to find the optimal value.

[18] According to DGNS, in all cases, fund selection is based on performance predicted according to the FF5+MOM model.

and the reproduced results discussed earlier. We first evaluate the performance of the machine learning portfolios, then analyze feature importance and finally the underlying economic mechanism. Although the empirical analysis is conducted at the share-class level, consistent with DGNS, we refer to share classes as funds for simplicity throughout the remainder of this paper.

3.2.1. Replication of main result: Performance of machine learning portfolios

The replication results for the long-only top decile portfolio obtained relative to the FF5+MOM model are presented in column 2 of Table II.3. As in the reproduction, none of the algorithms are able to select long-only portfolios of funds that generate statistically significant positive net alpha.[19] The best-performing machine learning algorithm, gradient boosting, delivers a top-decile portfolio alpha of 7.1 bps per month (0.85% per year), followed by elastic net (0.61% per year), the benchmark OLS model (0.55% per year) and random forest (0.30% per year). The equally weighted portfolio exhibits a negative alpha, closely matching the result reported in the reproduction. Overall, the alphas in the replication differ in magnitude from those obtained in the reproduction but qualitatively confirm these findings (see Table II.1). In contrast to the reproduction results, the linear methods OLS and elastic net perform almost as well as gradient boosting, whereas random forest exhibits inferior forecasting performance, though none of these top decile results are statistically significant. To evaluate whether ML algorithms can identify low-performing funds ex-ante, column 3 of Table II.3 focuses on the bottom decile of the predicted alpha distribution. The results indicate that both nonlinear (gradient boosting, random forest) and linear (elastic net) ML algorithms, as well as the benchmark OLS model, reliably identify poorly performing funds, forming portfolios that deliver statistically significant negative net alpha at the 1% level. The algorithm associated with the lowest monthly bottom decile portfolio alpha is the benchmark OLS model (-25.0 bps), followed by elastic net (-24.7 bps), random forest (-22.5 bps) and gradient boosting (-18.3 bps).

Finally, column 4 of Table II.3 follows Kaniel et al. (2023) and presents the results for the long-short portfolios formed from the predictions of the ML algorithms and OLS. The results demonstrate that predictions based on machine-learning methods exploiting nonlinearities and interactions (random forest and gradient boosting) indeed capture true abnormal returns rather than simply compensation for risk. More specifically, gradient boosting and random forest select long-short

[19] This conclusion remains unchanged when alternative factor models are employed, as documented in Table II.B2.

portfolios of funds that deliver statistically significant (at the 1%-level) monthly net alphas of 25.4 bps and 25.0 bps (3.05% and 3.00% per year). These results are consistent with the findings of Kaniel et al. (2023), who use feed-forward neural networks to predict mutual-fund alpha based on a comprehensive set of predictors spanning the period from January 1980 to January 2019. In their forecast using only fund characteristics as predictors, they report a monthly alpha of 16 bps (1.92% per year) that is significant at the 1% level. However, it is important to note that long-short portfolios based on linear models (elastic net and OLS) generate net alphas that are even slightly higher than those of nonlinear ML methods and are statistically different from zero (at the 1%-level). Consistent with Kaniel et al. (2023), most of the predictability in after-fee alpha arises from the short leg of the long–short fund portfolios.

[Insert Table II.3 here]

The time series of cumulative abnormal returns based on the replication is shown in Figure II.3 and reveals a substantial deviation from the results reported in DGNS. The best performing ML algorithm in our replication, gradient boosting, achieves a cumulative net alpha of 22.76%. The equally weighted and asset-weighted portfolios earn a negative cumulative net alpha of -29.86% and -26.92%, respectively. While considerably below the magnitudes reported in the original study of DGNS, these values are largely consistent with those observed in the reproduction (see Figure II.1), except for random forest. In the reproduction (based on the dataset from the DGNS data repository), the cumulative net alpha for this ML algorithm is 29 percentage points higher.

[Insert Figure II.3 here]

Besides the forward-looking bias related to the construction of portfolio returns described in Section 2, the study of DGNS contains a survivorship bias (see Subsection 3.1.2 and 3.1.4), which we address in our replication. To assess the impact of this bias, we carry out a supplementary analysis in which it is intentionally left uncorrected. More specifically, we exclude all observations without a target variable ahead of predicting the decile portfolios. The results for the top-decile portfolios affected by this bias are reported in Table II.C1 and indicate that none of the algorithms generate a statistically significant alpha. For random forest, elastic net and OLS, the estimated coefficients are somewhat larger than in the look-ahead bias–corrected version (Table II.3), with elastic net emerging as the best-performing

algorithm, yielding an annual alpha of 0.88%. This indicates that the look-ahead bias indeed leads to an upward bias in alphas except for gradient boosting.

Overall, the empirical results in this replication reveal that the applied ML methods and OLS fail to identify long-only portfolios with statistically significant positive alpha but have predictive ability in selecting poorly performing funds. Furthermore, the main source of outperformance of long-short portfolios of funds arises from the latter, as reflected in the significant spanning-portfolio alphas delivered by all algorithms. This finding is consistent with the discussion related to the reproduction in Section 2, showing that once the forward-looking bias in portfolio return construction is removed, no ML algorithm generates a statistically significant long-only alpha.

3.2.2. Replication of other analyses

For completeness, we also replicate the remaining tables (tables 4 to 7) in the study of DGNS. The results are briefly summarized in this subsection and the tables are delegated to Appendix II.D.[20] Table 4 in DGNS reports significant outperformance of gradient boosting and random forest relative to OLS. In contrast, we find that the alphas of portfolios formed using gradient boosting, random forest and elastic net are not statistically different from those formed using OLS (Table II.D1). Furthermore, in contrast to Table 5 in DGNS, we find that the magnitudes of all ML algorithms and OLS in terms of Sharpe, Sortino and Information ratios are smaller (see Table II.D2). Additionally, portfolios constructed using double-sorting based on past performance and activeness (Table 6 in DGNS) do not yield significantly positive alphas (Table II.D3) and gradient boosting and random forest portfolios do not achieve positive alphas under any particular market conditions (Table II.D4, replication of their Table 7).

3.2.3. Feature importance

To evaluate how fund characteristics and their interactions affect the prediction of machine learning algorithms, DGNS compute model-agnostic SHAP (SHapley Additive exPlanations) values (Lundberg and Lee, 2017; Lundberg et al., 2020). SHAP values quantify the marginal contribution of each feature to the model's predictions, thereby improving the interpretability of the algorithms.[21] For the

[20] In most of these tables, we already include results for our extended sample period until 2024 that will be presented in Subsection 4.1.

[21] The basic idea of SHAP originates from the Shapley value in cooperative game theory and is used to fairly allocate the contributions of multiple participants in a collaboration. A general discussion of interpretable machine learning can be found in Molnar (2019).

purposes of replicating the analysis, we focus on the random forest model, as it performs best according to the results reported in DGNS. In their implementation, DGNS restrict the computation of SHAP values to the final estimation window covering the period from 1980 to 2019. Moreover, while the main results of the study rely on machine learning algorithms implemented in *R*, the SHAP analysis is conducted in *Python*. These two methodological choices might cause misleading interpretations.

The first issue concerns the data used to compute SHAP values. Based on the code provided by DGNS, it appears that SHAP values were computed using both training and test data. This is further suggested by the reporting of the time evolution of feature importance for gradient boosting and random forest based on a single estimation window. However, SHAP values are designed to explain a model's prediction on unseen data, rather than on the data used for training (Lundberg and Lee, 2017). To assess the predictive relevance of fund characteristics, SHAP values should therefore be computed on the prediction set for each estimation window using the corresponding trained model. The approach implemented in DGNS instead provides insights into how the ML model learned from the training data and which features had a strong predictive signal on future alpha, rather than the interpretability of its out-of-sample predictions. This issue is apparent in Figures 8 and 9 of their paper, where feature importance is reported starting in 1981, even though the first out-of-sample prediction occurs at the end of December 1990 for 1991. This distinction is important because, in a forecasting context, interpretability should reflect how the model generalizes to unseen data, where predictive performance and economic relevance are ultimately evaluated. By applying SHAP to both training and test sets, the DGNS approach risks overstating the importance of features that only contribute marginally to the out-of-sample predictions.

The second issue relates to inconsistencies between software implementations. While the primary results are based on machine learning models estimated in *R*, the SHAP values are computed separately in *Python*. Although DGNS explicitly specify the random seed and two hyperparameters (the number of trees and the number of features considered at each split) to match the *R* implementation when computing SHAP values, differences between the two implementations remain. First, their *R* model uses 1,000 trees, whereas their *Python* model uses only 100. Moreover, when some hyperparameters are not explicitly specified, each package applies its own default values, which differ across implementations (e.g., the minimum node size). Second, even when the same random seed is applied, it is unclear whether the random number generator behaves identically across the two languages. Overall,

these issues imply that the models used in this part of the analysis in DGNS are not equivalent.

To address these limitations, we compute SHAP values for each prediction window using the prediction set, implementing the entire procedure in *R* using the *randomForest* and *treeshap* libraries. Figure II.4 reports the average feature importance values. The importance of each characteristic is quantified by computing the mean of its absolute SHAP values across all observations and forecasting periods. Panel (a) presents the results reported in DGNS, which assess feature importance within the last estimation window spanning the 1980 to 2019 period. Panel (b) shows the corresponding results from the replication, which uses the same last estimation window as DGNS. However, unlike DGNS, where SHAP values are computed on the training and test set, we calculate SHAP values only on the test set (i.e., the last prediction iteration). Finally, in panel (c), we compute SHAP values for each one-year forecasting period. The reported feature importance represents the average of the absolute SHAP values across all out-of-sample predictions and forecasting iterations.

The main finding from Figure II.4 is that the results reported in DGNS and the replication differ in characteristic importance. The most notable difference is the importance of value added, which ranks first in DGNS but decreases to thirteenth in panel (b) and twelfth in panel (c), suggesting only limited relevance for predicting future alpha in a true out-of-sample setting. This difference may be due to the data leakage identified in the original analysis, where the feature's apparent predictive ability reflects information implicitly derived from the training data rather than actual forecasting ability.

In terms of measures of past performance, the alpha (intercept t-stat) turns out to be a more robust predictor in forecasting future alpha and is ranked among the two most important characteristics in both panels (first in panel b and second in panel c) and is largely consistent with the results reported in DGNS. The most important feature in the full out-of-sample replication is value beta t-stat, which ranks first in panel (c), but does not fall within the top five characteristics in DGNS. Additionally, unlike in DGNS, the momentum beta t-stat also appears among the top five features in panel (c). In contrast, fund characteristics such as manager tenure, flows[22] and volatility of flows consistently rank near the bottom across all panels, indicating limited predictive relevance. Overall, the replication results in panels (b) and (c), which are based on true out-of-sample data, suggest that the

[22] Kaniel et al. (2023) find in their study that fund momentum and fund flow are the most important predictors of future alpha.

most important predictors are alpha (intercept t-stat) and the variables that proxy for a fund's exposure to systematic risk factors, as measured by their t-statistics.

[Insert Figure II.4 here]

3.2.4. Economic mechanism

Our critique of DGNS concerns mainly their conclusion that machine learning methods can identify long-only mutual fund portfolios with statistically significant positive alpha. According to DGNS, the underlying economic mechanism of this finding is related to capital misallocation in the mutual fund market. More specifically, the application of machine learning methods allows the identification of fund managers whose skill is not sufficiently offset by diseconomies of scale (Berk and Green, 2004), consistent with the presence of informational frictions preventing investors from identifying outperforming funds (Roussanov, Ruan, and Wei, 2021). In this section, we assess this finding of DGNS based on the decile portfolios generated by the four prediction methods (random forest, gradient boosting, elastic net, OLS) in our replication.

We follow DGNS and assume that the net alpha of a fund can be decomposed into skill, diseconomies of scale, expense ratio and a zero-mean idiosyncratic shock. The expected net alpha of fund i is written as

$$E\left(\alpha_{i,t+1}|\,\mathcal{F}_t\right) = \hat{a}_{i,t+1} - D\left(Q_{i,t}\right) - p_{i,t}, \tag{II.2}$$

where $\hat{a}_{i,t+1} = E\left(a_{i,t+1}|\,\mathcal{F}_t\right)$ is the expected skill conditional on the information set $\mathcal{F}$, $D\left(Q_{i,t}\right)$ captures diseconomies of scale given the size of fund i in period t, $Q_{i,t}$ and $p_{i,t}$ denotes the expense ratio of fund i at time t. Based on the observed persistence of fund expense ratios, DGNS assume that the expense ratio at time t is a reliable predictor of the expense ratio at time $t+1$. Moreover, as in DGNS, we follow Roussanov, Ruan, and Wei (2021) and assume that the diseconomies of scale are logarithmic, $D\left(Q_{i,t}\right) = \eta \log(Q_{i,t})$ where $\eta = 0.0048$ and $Q_{i,t}$ equals the assets under management of all of the fund's share classes at the end of year t, expressed in 2015 dollars.[23] To estimate the expected skill for fund i in year t, $\hat{a}_{i,t+1}$, we apply the estimation procedure outlined in DGNS and average the fund's (annual) realized alphas before fees and diseconomies of scale from the fund's inception as initially suggested by Zhu (2018).

[23]The data to adjust assets under management for inflation are taken from the replication package of DGNS. The Consumer Price Index (CPI) corresponds to the FRED series "Consumer Price Index for All Urban Consumers: All Items in U.S. City Average, Index 1982-1984=100, Monthly, Seasonally Adjusted."

Figure II.5 illustrates the key economic mechanism underlying the empirical findings of DGNS, showing how capital misallocation (Roussanov, Ruan, and Wei, 2021) in the mutual-fund market may explain the performance of both OLS and machine-learning-based portfolios. Panel (a) presents the results from DGNS, while panel (b) shows the corresponding results from the replication. In each panel, the horizontal axis reports the mean net skill, $E\left(\hat{a}_{i,t+1} - p_{i,t} \middle| i \in D_j\right)$ for the jth decile portfolio, where funds are sorted by predicted alpha and D_j denotes the set of funds in the jth decile. The vertical axis shows the mean log size, $E\left(\log(Q_{i,t}) \middle| i \in D_j\right)$, for funds in D_j. The colored lines represent the mean log fund size across decile portfolios, corresponding to predictions from OLS (red), elastic net (yellow), gradient boosting (purple) and random forest (green). For each level of net skill, we also plot the efficient log size, $\log(Q_{i,t}^{BG})$, as implied by the perfectly competitive equilibrium of Berk and Green (2004), which is obtained by dividing net skill by the diseconomies of scale parameter, η (black).

Both panels show a positive relationship between mean net skill and fund size across the decile portfolios for all prediction methods, suggesting that these methods successfully identify managers with higher net skill. Panel (a), which illustrates the results of DGNS, shows that for the first eight decile portfolios, higher mean net skill is associated with larger fund size. Moreover, the average size of these eight portfolios exceeds the efficient size implied by the frictionless model of Berk and Green (2004). In the replication, shown in panel (b) of Figure II.5, we observe a qualitatively similar but less pronounced skill–size relationship, with substantially lower dispersion across ML methods. Consistent with the findings of DGNS, we observe that funds in the top decile of the predicted alpha distribution generated by nonlinear ML methods (gradient boosting and random forest) manage, on average, smaller portfolios than funds in the second-highest decile. However, in the replication, the decrease in mean log size observed for these methods is much less marked than in DGNS. Moreover, the results for linear models (elastic net and OLS) indicate larger average fund size in the top decile, given the level of mean net skill, but remain substantially below their efficient size suggested by the Berk and Green (2004) model. Overall, our results support the conclusion of DGNS that nonlinear machine learning methods improve mutual fund selection not only by identifying skilled managers but also by detecting those whose skill is not fully offset by diseconomies of scale. Nonetheless, this also holds for linear models (in DGNS as well as the replication), as the mean log size of their portfolios is below the efficient size implied by the model of Berk and Green (2004).

[Insert Figure II.5 here]

4. Extensions and additional insights

In the final part of this paper, we extend our analyses in several dimensions following the methodology outlined in Section 3. First, we update the sample through the end of 2024 to test the robustness of our findings with additional data. Second, motivated by the long-term focus of institutional investors[24], we extend the forecasting horizon and corresponding holding period to 36 months. Third, to examine whether a single sorting variable can predict future risk-adjusted returns (e.g., Carhart, 1997; Cremers and Petajisto, 2009; Amihud and Goyenko, 2013), we employ a simple univariate portfolio sorting approach based on the 17 predictors used in DGNS.

4.1. Expansion of sample period until 2024

In this part of the paper, we extend the sample period through the end of 2024 to evaluate the robustness of our findings with additional data. The results reported in Table II.4 confirm our earlier conclusion that ML-selected mutual fund portfolios do not produce statistically significant long-only alphas. At the same time, consistent with Kaniel et al. (2023), they reinforce the notion that nonlinear ML methods are able to benefit fund selection due to their ability to separate high- from low-performing funds ex ante, thereby generating a statistically significant long–short portfolio alpha. However, as in the original DGNS sample period spanning January 1991 to December 2020, we also find that OLS and elastic net are able to separate high- from low-performing funds and generate statistically significant long–short portfolio alphas that are even slightly higher than those produced by nonlinear machine-learning methods. These results confirm our previous findings presented in Table II.3.

[Insert Table II.4 here]

4.2. Forecasting and investment horizon

Most of the literature that studies mutual fund performance has a short-term focus (for a review, see Cremers, Fulkerson, and Riley, 2019). Two notable exceptions are the studies by Bessembinder, Cooper, and Zhang (2023, 2025), which adopt a

[24] In the survey sample of McCahery, Sautner, and Starks (2016), 71% of the respondents indicate a holding period of more than two years of their investments and suggest that they hold most of their investments (75% on average) in active rather than purely passive (indexed) positions.

long-horizon perspective. In this section, we examine the effects of extending the forecasting horizon to three years, in line with the long-term investment focus of institutional investors (McCahery, Sautner, and Starks, 2016). In this context, we repeat the analysis for all prediction methods using the cumulative alpha over a period of 36 months as the target variable and the same predictors as in Section 3, resulting in an extended forecasting horizon of three years.[25] More specifically, consistent with the forecasting approach outlined in Section 3, we use the first 10 years of data for the initial training, employing three-year-ahead realized alphas (three-year periods ending from 1983 to 1990) and lagged fund characteristics (from 1980 to 1987) to train two nonlinear (gradient boosting, random forest) and one linear ML algorithm (elastic net) as well as the benchmark model (OLS). With a 36-month forecasting horizon, model training and prediction are conducted every third year using annual data and an expanding window, ensuring that all available information is incorporated in model calibration. Values of fund characteristics in December 1990, not employed in the training process, are used to predict the cumulative fund performance from January 1991 to December 1993. This procedure is consistent with the return predictability literature, in which the target variable is cumulated over the h-period forecasting horizon, while the predictor is taken at the time of the forecast (e.g., Fama and French, 1989; Cooper and Priestley, 2009; Atanasov, Møller, and Priestley, 2020; Goyal, Welch, and Zafirov, 2024). The out-of-sample performance is again evaluated based on a portfolio approach as in Section 3 but with a holding period of 36 months. We form equally weighted portfolios based on share classes in the top and bottom deciles of the predicted alpha distribution and track the net returns over the subsequent 36 months of the forecasting horizon. The prediction and portfolio formation procedure is repeated every three years until the end of the sample, yielding a time series of monthly out-of-sample net returns for the top and bottom-decile portfolios. The risk-adjusted out-of-sample performance of the top and bottom decile portfolios is estimated by running a time-series regression of out-of-sample monthly portfolio excess returns (net returns minus the risk-free rate) on the FF5+MOM factor model.

The results for the sample period spanning January 1991 through December 2020 (360 months) are shown in panel (a) of Table II.5 and reveal three main insights. First, ML methods do not identify long-only portfolios with a statistically significant positive alpha. Second, all ML algorithms (gradient boosting, random forest and elastic net) as well as the benchmark OLS model reliably identify the

[25] DGNS examine the effects of reducing the portfolio rebalancing frequency from 12 to 24 and 36 months (reported in their internet appendix), but they do not consider the implications of extending the forecasting horizon beyond 12 months.

worst-performing funds in the predicted alpha distribution by selecting mutual fund portfolios that generate statistically significant negative net alpha at the 10% level or higher relative to the FF5+MOM factor model. Third, long–short portfolios constructed from the top and bottom deciles of the predicted alpha distribution yield statistically significant net alphas only for the nonlinear ML algorithms. The best-performing algorithm, random forest, delivers a highly significant monthly out-of-sample alpha of 25.5 bps (equivalent to 3.06% annually), which is slightly higher than the 25.0 bps based on a one-year prediction horizon as reported in Table II.3. In panel (b) of Table II.5, we extend the sample period through the end of 2024 and confirm the robustness of our results.

Overall, the results suggest that mutual fund performance predictability persists over a three-year forecasting horizon by separating high from low-performing funds. In contrast to a one-year forecasting horizon, OLS and elastic net do not produce any significant long-short portfolio alpha, reinforcing the relative advantage of ML at longer horizons. Again, most of the economic value arises from avoiding funds that the applied models predict to be the worst performers.

[Insert Table II.5 here]

4.3. Univariate sorts

To further examine the relation between mutual fund characteristics and subsequent performance, we implement univariate portfolio sorts following standard practice in the empirical asset pricing literature (see, e.g., Carhart, 1997; Cremers and Petajisto, 2009; Amihud and Goyenko, 2013). We construct mutual fund portfolios using the 17 share class characteristics that serve as predictors in our forecasting exercise in Section 3. At each annual formation date (December of each year), funds are ranked based on the magnitude of these characteristics, $z_{i,t}$. Following Li and Rossi (2021), high (low) decile portfolios consist of the share classes with the highest (lowest) values of $z_{i,t}$ and are equally weighted. We then construct a long–short portfolio by taking a long position in the highest and a short position in the lowest decile portfolio and hold it for the subsequent year $(t+1)$. We repeat the analysis for a forecasting and holding period of 3 years (until $t+3$). As in Section 3, we compute the risk-adjusted out-of-sample performance for each portfolio using the FF5+MOM factor model and statistical inference is based on HAC-robust standard errors (Newey and West, 1987) with 12 lags.

Table II.6 reports the results for each of the 17 characteristics, sorted based on the significance of the spanning portfolio for a forecasting horizon of 12 months.

Among these characteristics, five deliver statistically significant alpha at the 5% level or higher (spanning portfolio). We find that a strategy long in funds with a high alpha t-stat and short in funds with a low alpha t-stat yields a monthly outperformance of 20.4 bps (2.45% annually), which is statistically significant at the 1% level. A similar pattern emerges for value added, which yields a highly significant spanning-portfolio alpha of 15.8 bps (1.90% annually). For realized alpha, we also find a significant long–short portfolio alpha of 19.4 bps (2.33% annually), statistically significant at the 5% level. Moreover, risk-adjusted returns of such long-short strategies are also significant at the 10% level or higher for age, value beta t-stat, profitability beta t-stat and investment beta t-stat. For all these fund characteristics, the negative alpha of the long-short portfolio indicates that funds with higher values of the characteristic tend to earn lower future risk-adjusted returns. The long-short strategies based on the remaining 10 characteristics do not deliver a statistically significant outperformance. Comparing the univariate sorts in this section with the more sophisticated ML approach in Table II.3 reveals that the best ex-post univariate regressor, alpha t-stat, generates a monthly out-of-sample alpha of 20.4 bps for the long–short portfolio, whereas the best-performing ML algorithm, elastic net, delivers a higher monthly out-of-sample risk-adjusted return of 29.8 bps.

[Insert Table II.6 here]

For a 36-month horizon (see Table II.7), a similar pattern emerges. The best performing univariate sort, investment beta t-stat, yields an absolute alpha of 22.3 bps (2.68% annualized), which is comparable to the best ML algorithm, random forest (25.5 bps; 3.06% annualized)

[Insert Table II.7 here]

5. Conclusion

DGNS argue that machine learning methods using fund characteristics as predictors can be employed to construct tradable, long-only equity mutual fund portfolios that generate economically and statistically significant out-of-sample alpha. In this study, we show that these conclusions are invalid and arise from a coding error related to the construction of portfolio returns. Correcting this error eliminates the statistical significance and reduces the annual alpha by 1.42 percentage points for the best-performing ML algorithm.

We corroborate this finding in an independent replication in which we correct another, less impactful error leading to survivorship bias. While long-only portfolios fail to deliver significant alpha, we demonstrate that nonlinear ML methods can distinguish between high- and low-performing funds ex-ante based on long–short portfolios formed from the top and bottom deciles of the predicted alpha distribution. However, it is important to note that over a one-year forecasting horizon, OLS and elastic net also yield statistically significant long–short alphas that slightly exceed those from nonlinear ML algorithms. Given that mutual funds cannot be shorted directly, the economic interpretation differs substantially from that in DGNS, since the benefits of machine learning methods and OLS mainly accrue from avoiding mutual funds that are likely to underperform in the future.

Finally, we extend the analysis of DGNS along three dimensions. First, we show that our findings remain robust when the sample period is extended through the end of 2024. Second, we demonstrate that the replication results continue to hold at a 36-month forecasting horizon, consistent with the longer-term investment horizon of institutional investors. In this setting, statistically significant net alphas from long–short portfolios formed on predicted alpha deciles arise only for the nonlinear ML algorithms. Hence, our results imply that the advantages of more sophisticated ML prediction methods arise primarily over longer forecasting horizons, though in neither case do they generate statistically significant long-only alphas. Third, we examine the relationship between mutual fund characteristics and subsequent performance using univariate portfolio sorts. For the strongest univariate predictors, the alphas of long-short portfolios are comparable to the best-performing machine learning algorithms. This suggests that relatively simple methods can be sufficient to identify future underperforming funds, independent of the forecasting horizon.

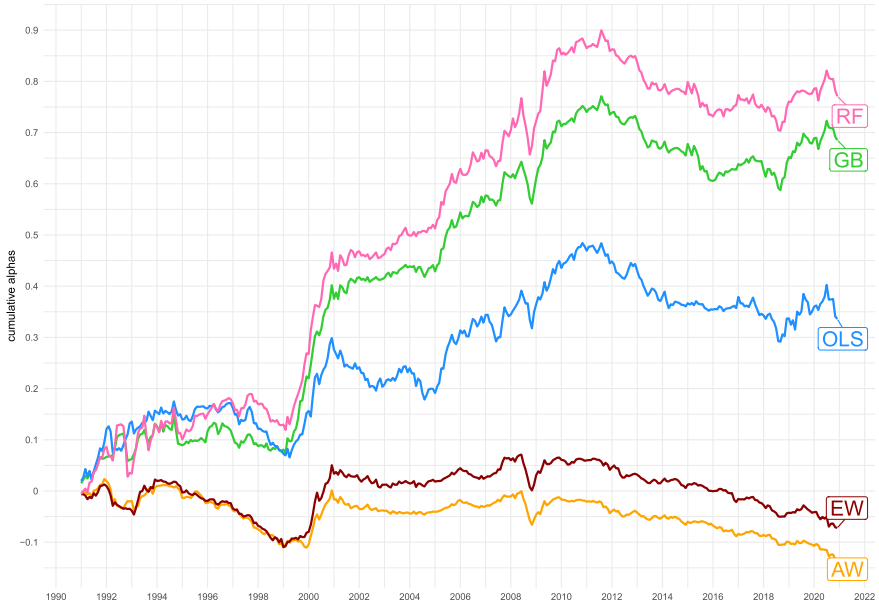
Table II.1: **Monthly out-of-sample alpha (in %) - Reproduction**

Column 1 of this table reports the monthly out-of-sample alphas (in %) net of all costs of the equally weighted top decile portfolios from Table 3 in DGNS. Column 2 presents the estimates based on their data repository (which contains added noise due to the proprietary nature of the CRSP data). The results based on the repository after correcting for the forward-looking bias induced in the portfolio weights are shown in column 3. The out-of-sample alphas are obtained by regressing excess monthly portfolio returns on the main factor model used in DGNS, i.e., the Fama and French (2015) 5-factor model (FF5) extended by the momentum factor of Carhart (1997) (FF5+MOM). The out-of-sample period spans from January 1991 to December 2020 (360 observations). HAC-robust standard errors, as suggested by Newey and West (1987), are reported in parentheses. As in DGNS, the number of lags is set to 12. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5% and 1% level, respectively.

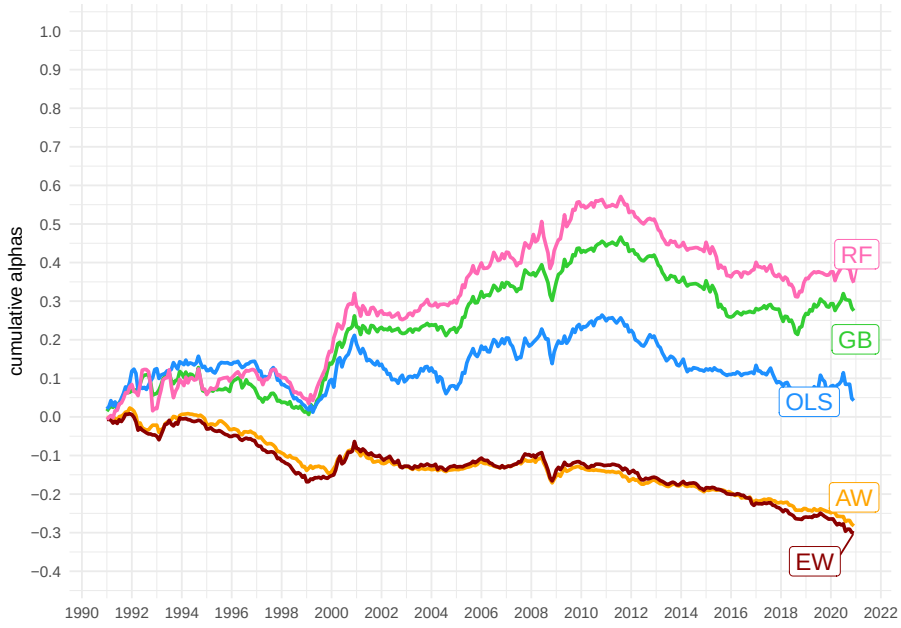
	DGNS published	**Data repository**	**Reproduction**
Gradient boosting	0.197**	0.199**	0.083
	(0.080)	(0.080)	(0.073)
Random forest	0.224**	0.224**	0.106
	(0.087)	(0.087)	(0.082)
Elastic net	0.091	0.091	0.009
	(0.069)	(0.069)	(0.065)
OLS	0.101	0.100	0.019
	(0.066)	(0.066)	(0.062)
Equally Weighted	-0.018	-0.018	-0.082**
	(0.044)	(0.044)	(0.039)
Asset Weighted	-0.037	-0.034	-0.077**
	(0.035)	(0.035)	(0.032)

Figure II.1: **Cumulative portfolio alpha of DeMiguel et al. (2023) and the reproduction**

This figure plots the time series of cumulative out-of-sample realized alphas of the excess returns net of all costs of the top decile portfolios. The realized portfolio alphas are obtained using Fama and French (2015) five-factor regressions augmented with the momentum factor of Carhart (1997). The portfolios are constructed using random forest (RF), gradient boosting (GB), OLS and with two naive strategies (equally weighted (EW) and asset-weighted (AW) portfolios of all available funds). Panel (a) reports the results from DGNS and panel (b) presents those from the reproduction. The out-of-sample period spans from January 1991 to December 2020.



(a) Cumulative out-of-sample portfolio alpha - DGNS



(b) Cumulative out-of-sample portfolio alpha - reproduction

Table II.2: **Share class characteristics: Descriptive statistics**

This table reports monthly descriptive statistics (mean, median, standard deviation and number of class-month observations) of all share class characteristics considered. Panel (a) shows the statistics reported in DGNS and Panel (b) presents the corresponding values from the replication. Expense ratio, manager tenure, turnover ratio and volatility of flows are reported based on annual data. All variables are measured at the share class level and correspond to actively managed US domestic equity share classes from January 1980 to December 2020.

	Mean	Median	Standard deviation	Class-month observations
Panel (a): DGNS				
monthly return	0.86%	1.25%	5.23%	718,928
monthly realized alpha	-0.14%	-0.13%	2.22%	676,147
alpha *t*-stat	-0.431	-0.430	1.1209	676,475
TNA (USD mill.)	679.9	97.4	2,593	719,398
expense ratio	1.11%	1.04%	0.52%	712,564
age (months)	145.7	117.0	109.8	719,398
flows	0.002	-0.004	0.094	718,734
manager tenure (years)	8.219	7.005	5.352	656,418
turnover ratio	0.790	0.550	1.141	711,568
volatility of flows	0.173	0.091	0.240	704,945
value added	-0.295	-0.016	37.233	669,727
market beta *t*-stat	16.667	15.064	10.591	676,475
profitability beta *t*-stat	-0.125	-0.125	1.463	676,475
investment beta *t*-stat	-0.444	-0.495	1.544	676,475
size beta *t*-stat	1.460	0.617	3.801	676,475
value beta *t*-stat	0.022	-0.081	2.195	676,475
momentum beta *t*-stat	0.009	0.026	1.878	676,475
R^2	0.907	0.944	0.122	676,475
Panel (b): Replication				
monthly return	0.88%	1.29%	5.30%	999,959
monthly realized alpha	-0.14%	-0.14%	2.15%	936,600
alpha *t*-stat	-0.436	-0.437	1.207	936,636
TNA (USD mill.)	628.2	99.3	2487.5	1,008,457
expense ratio	1.13%	1.04%	1.01%	826,264
age (months)	148.9	124.0	105.7	1,008,457
flows	-0.002	-0.005	0.059	999.022
manager tenure (years)	7.890	6.833	5.556	757,988
turnover ratio	0.748	0.530	0.953	825,887
volatility of flows	0.130	0.081	0.137	1,007,164
value added	-0.266	-0.008	39.879	800,972
market beta *t*-stat	16.847	15.457	9.715	936,636
profitability beta *t*-stat	-0.193	-0.183	1.456	936,636
investment beta *t*-stat	-0.486	-0.545	1.549	936,636
size beta *t*-stat	1.645	0.735	3.861	936,636
value beta *t*-stat	-0.007	-0.128	2.207	936,636
momentum beta *t*-stat	-0.013	0.010	1.822	936,636
R^2	0.911	0.947	0.117	936,636

Figure II.2: **Correlation matrix between the target variable and fund characteristics**

This figure reports the correlation coefficients between the target variable (realized alpha) and 17 fund characteristics used as predictors. Predictors are lagged one year with respect to the target variable. The sample period spans from January 1980 to December 2020.

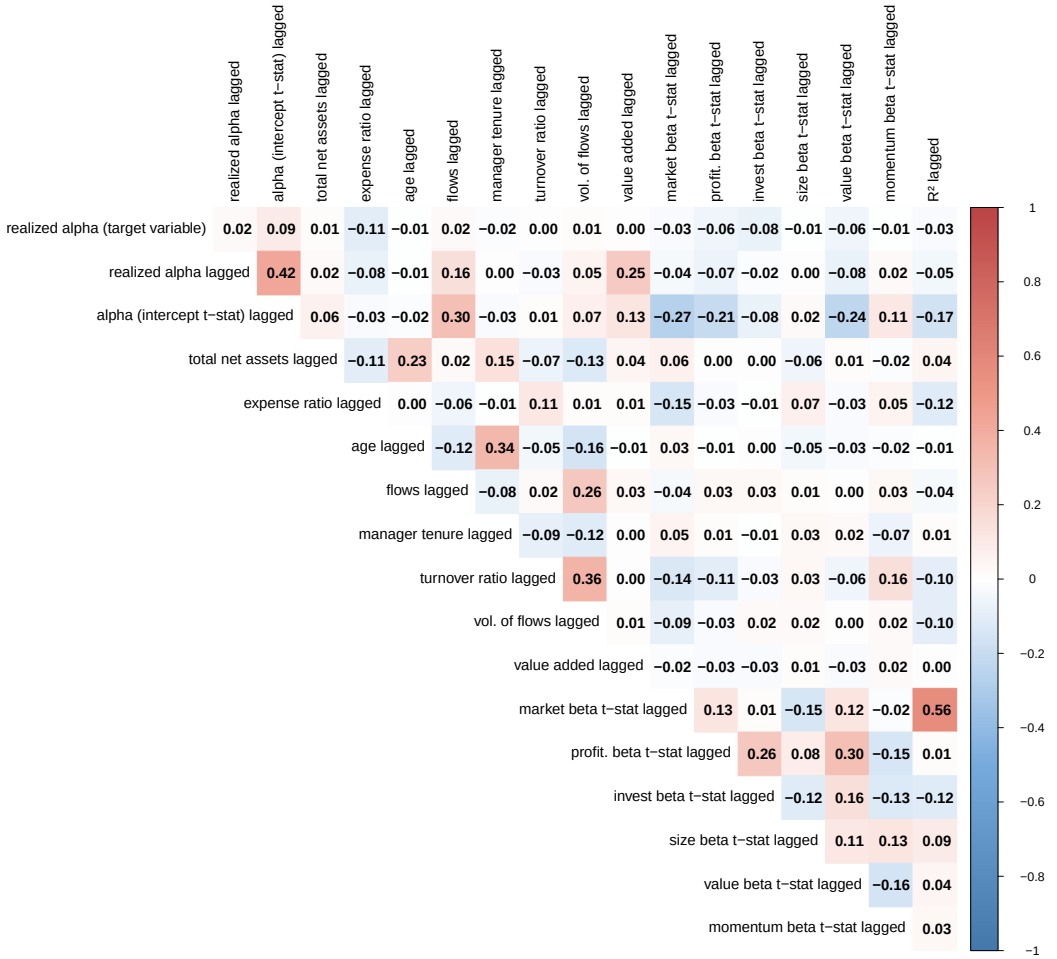


Table II.3: **Monthly out-of-sample alpha (in %) - Replication**

Column 1 of this table reports the monthly out-of-sample alphas (in %) net of all costs of the equally weighted top decile portfolios from Table 3 in DGNS and column 2 shows the results of the replication. Column 3 reports the monthly out-of-sample alphas (in %) net of all costs of the equally weighted bottom decile portfolios based on the predicted alpha distribution. Column 4 shows the monthly out-of-sample alpha of the spanning portfolio, which is long in the top decile and short in the bottom decile of the predicted alpha distribution. The out-of-sample alphas are obtained by regressing excess monthly portfolio returns on the main factor model used in DGNS, i.e., the Fama and French (2015) 5-factor model (FF5) extended by the momentum factor of Carhart (1997) (FF5+MOM). The out-of-sample period spans from January 1991 to December 2020 (360 observations). HAC-robust standard errors, as suggested by Newey and West (1987), are reported in parentheses. As in DGNS, the number of lags is set to 12. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5% and 1% level, respectively.

		Replication		
	DGNS published	**Top decile**	**Bottom decile**	**Long-short**
Gradient boosting	0.197**	0.071	-0.183***	0.254***
	(0.080)	(0.080)	(0.060)	(0.084)
Random forest	0.224**	0.025	-0.225***	0.250***
	(0.087)	(0.078)	(0.062)	(0.092)
Elastic net	0.091	0.051	-0.247***	0.298***
	(0.069)	(0.067)	(0.059)	(0.097)
OLS	0.101	0.046	-0.250***	0.296***
	(0.066)	(0.067)	(0.058)	(0.097)
Equally Weighted	-0.018	-0.081**	-0.081**	
	(0.044)	(0.037)	(0.037)	
Asset Weighted	-0.037	-0.074**	-0.074**	
	(0.035)	(0.030)	(0.030)	

Figure II.3: **Cumulative portfolio alpha of the replication**

This figure plots the time series of cumulative out-of-sample realized alphas of the excess returns net of all costs of the top decile portfolios from the replication. The realized portfolio alphas are obtained using Fama and French (2015) five-factor regressions augmented with the momentum factor of Carhart (1997). The portfolios are constructed using random forest (RF), gradient boosting (GB), OLS and with two naive strategies (equally weighted (EW) and asset-weighted (AW) portfolios of all available funds). The out-of-sample period spans from January 1991 to December 2020.

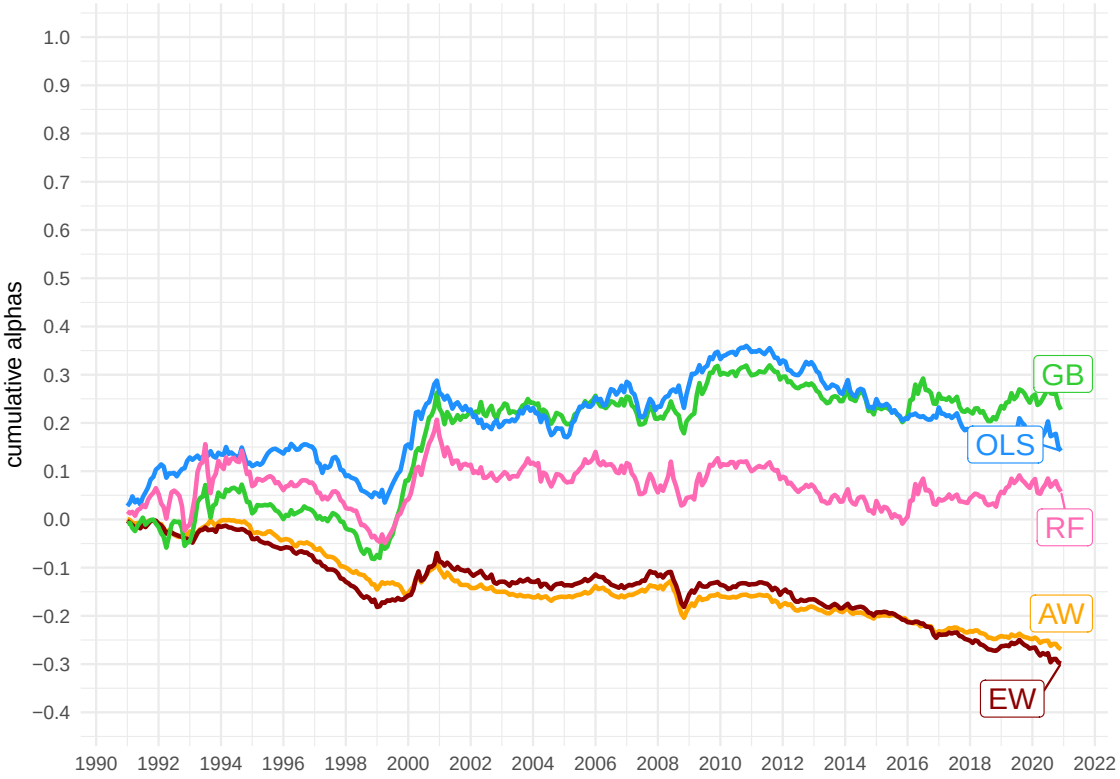
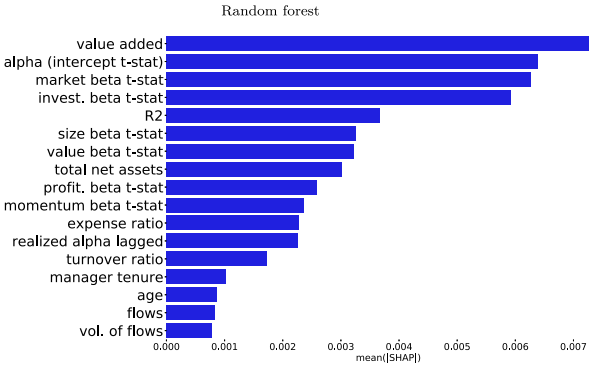
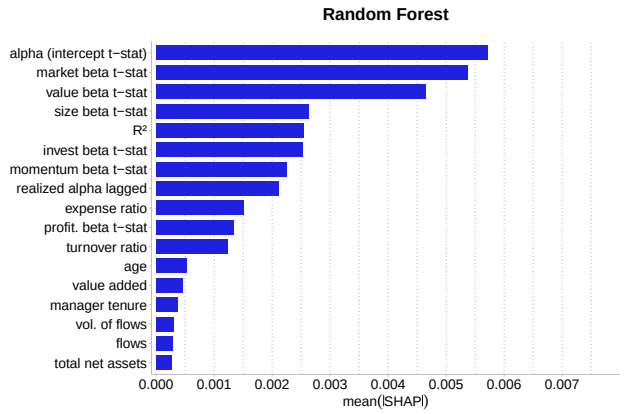


Figure II.4: **Characteristic importance**

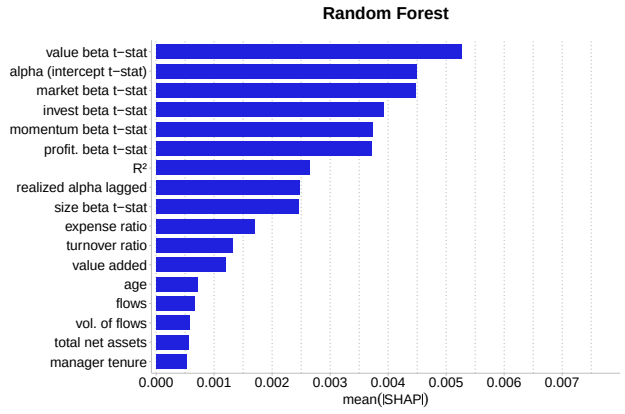
This figure presents the mean absolute SHAP values across all observations and forecasting periods for a one-year forecasting horizon for random forest. Panels (a) and (b) compare feature importance results from DGNS and the replication, respectively, both using the final estimation window from 1980 to 2019. While DGNS compute SHAP values on the training and test set, the replication uses the test set to explain the ML model's prediction on unseen data. Panel (c) extends this approach by averaging SHAP values across all one-year out-of-sample forecasting periods. SHAP values are calculated over the out-of-sample period from January 1991 to December 2020.



(a) Characteristic importance - DGNS



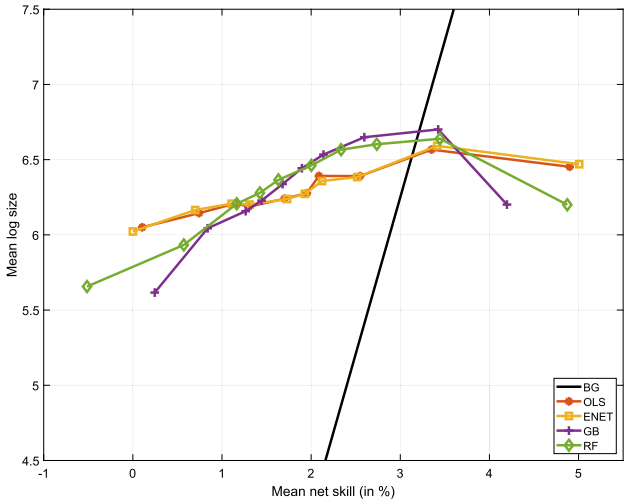
(b) Characteristic importance (last iteration) - Replication



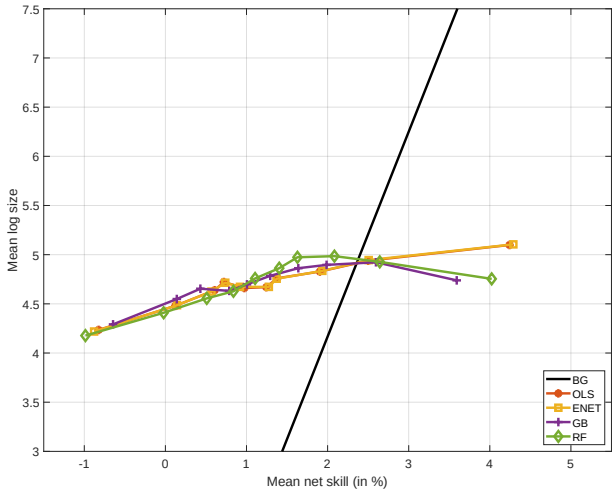
(c) Characteristic importance (full out-of-sample) - Replication

Figure II.5: **Capital misallocation and machine learning, DeMiguel et al. (2023) vs. Replication**

This figure illustrates capital misallocation for the decile portfolios generated by the four prediction methods considered in DGNS and the replication. In each panel, the horizontal axis reports the mean net skill, $E\left(\hat{a}_{i,t+1} - p_{i,t} | \, i \in D_j\right)$ for the jth decile portfolio, where funds are sorted by predicted alpha and D_j denotes the set of funds in the jth decile. The vertical axis shows the mean log size, $E\left(\log(Q_{i,t}) | \, i \in D_j\right)$, for funds in D_j. Net skill is the average of past realized alpha before fees and diseconomies of scale estimated using the approach in DGNS, initially suggested by Zhu (2018), minus the expense ratio. Diseconomies of scale are calculated following Roussanov, Ruan, and Wei (2021) as the logarithm of fund size multiplied by the estimated diseconomies of scale parameter $\eta = 0.0048$. For each level of net skill, we plot the efficient log size, $\log(Q_{i,t}^{BG})$, as implied by the perfectly competitive equilibrium of Berk and Green (2004), which is obtained by dividing net skill by the diseconomies of scale parameter, η (straight black line). The colored lines represent the mean log fund size across decile portfolios, corresponding to predictions from OLS (red circles), elastic net (yellow squares), gradient boosting (purple triangles) and random forest (green diamonds). Panel (a) illustrates the results from DGNS and panel (b) shows the results from the replication. The out-of-sample period spans from January 1991 to December 2020.



(a) Capital misallocation and machine learning - DGNS



(b) Capital misallocation and machine learning - Replication

Table II.4: **Monthly out-of-sample alpha (in %) - Extended data set through 2024**

This table reports monthly out-of-sample alphas (in percent), net of all costs, for the top and bottom deciles (columns 1 and 2) of the predicted alpha distribution obtained from three ML methods (gradient boosting, random forests and elastic net), Ordinary Least Squares (OLS) and two naive benchmarks (equally weighted and asset-weighted portfolios of all available funds). It also reports the spanning portfolio, which is long the top decile and short the bottom decile. The out-of-sample alphas are obtained by regressing excess monthly portfolio returns on the main factor model used in DGNS, i.e., the Fama and French (2015) 5-factor model (FF5) extended by the momentum factor of Carhart (1997) (FF5+MOM). The out-of-sample period spans from January 1991 to December 2024 (408 observations). HAC-robust standard errors, as suggested by Newey and West (1987), are reported in parentheses. As in DGNS, the number of lags is set to 12. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5% and 1% level, respectively.

	Top decile	**Bottom decile**	**Long-short**
Gradient boosting	0.010	-0.188***	0.198**
	(0.074)	(0.049)	(0.077)
Random forest	0.029	-0.204***	0.232***
	(0.074)	(0.053)	(0.087)
Elastic net	0.030	-0.242***	0.272***
	(0.065)	(0.051)	(0.090)
OLS	0.031	-0.254***	0.285***
	(0.066)	(0.051)	(0.092)
Equally Weighted	-0.096***	-0.096***	
	(0.034)	(0.034)	
Asset Weighted	-0.084***	-0.084***	
	(0.027)	(0.027)	

Table II.5: **Monthly out-of-sample alpha (in %) for a forecasting horizon of 36 months**

This table reports monthly out-of-sample alphas (in %), net of all costs, for the top-decile portfolio (column 1), the bottom-decile portfolio (column 2) and a spanning portfolio (column 3) that goes long in the top decile and short in the bottom decile of the predicted alpha distribution. Results are shown for gradient boosting, random forest, elastic net, OLS and two naive benchmarks (equally weighted and asset-weighted portfolios of all available funds), using a forecasting horizon of 36 months. The out-of-sample alphas are obtained by regressing excess monthly portfolio returns on the main factor model used in DGNS, i.e., the Fama and French (2015) 5-factor model (FF5) extended by the momentum factor of Carhart (1997) (FF5+MOM). Panel (a) covers an out-of-sample period from January 1991 to December 2020 (360 observations), whereas panel (b) extends this period through December 2024 (408 observations). HAC-robust standard errors, as suggested by Newey and West (1987), are reported in parentheses. As in DGNS, the number of lags is set to 12. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5% and 1% level, respectively.

	Top decile	**Bottom decile**	**Long-short**
	Panel (a): Sample through 2020		
Gradient Boosting	0.066	-0.144*	0.210**
	(0.083)	(0.074)	(0.087)
Random Forest	0.079	-0.176***	0.255***
	(0.081)	(0.059)	(0.071)
Elastic Net	-0.030	-0.149**	0.119
	(0.060)	(0.065)	(0.084)
OLS	-0.033	-0.150**	0.118
	(0.064)	(0.064)	(0.084)
Equally weighted	-0.098**	-0.098**	
	(0.039)	(0.039)	
Asset weighted	-0.082*	-0.082*	
	(0.043)	(0.043)	
	Panel (b): Sample through 2024		
Gradient Boosting	0.002	-0.184***	0.186**
	(0.074)	(0.052)	(0.080)
Random Forest	0.073	-0.185***	0.258***
	(0.089)	(0.055)	(0.088)
Elastic Net	-0.057	-0.153***	0.096
	(0.056)	(0.055)	(0.075)
OLS	-0.067	-0.159***	0.092
	(0.059)	(0.054)	(0.076)
Equally weighted	-0.114***	-0.114***	
	(0.036)	(0.036)	
Asset weighted	-0.085**	-0.085**	
	(0.038)	(0.038)	

Table II.6: **Monthly out-of-sample alpha (in %) of univariate sorts - 12 months**

This table presents univariate sorts on 17 fund characteristics used in DGNS. For year $t+1$, funds are sorted into deciles based on the characteristic measured in December of year t and equally weighted portfolio returns are calculated. A long–short portfolio is formed by taking a long position in the tenth decile and a short position in the first decile with a holding period of 12 months. The table reports the top decile (column 1), bottom decile (column 2) and the spanning portfolio (column 3). As in Section 3, the out-of-sample alphas are obtained by regressing excess monthly portfolio returns on the Fama and French (2015) 5-factor model (FF5) extended by the momentum factor of Carhart (1997) (FF5+MOM). The out-of-sample period spans from January 1991 to December 2020 (360 observations). HAC-robust standard errors, as suggested by Newey and West (1987), are reported in parentheses. The number of lags is set to 12. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5% and 1% level, respectively. To improve the readability of the table, we sort fund characteristics according to their significance level of the spanning portfolio (long-short).

	top decile	**bottom decile**	**Long-short**
alpha t-stat	0.036	-0.168***	0.204***
	(0.049)	(0.039)	(0.056)
value added	-0.004	-0.162***	0.158***
	(0.056)	(0.047)	(0.065)
age (months)	-0.109***	-0.053	-0.056**
	(0.039)	(0.042)	(0.025)
realized alpha	0.021	-0.173***	0.194**
	(0.073)	(0.066)	(0.089)
value beta t-stat	-0.160***	0.038	-0.198**
	(0.057)	(0.077)	(0.099)
profitability beta t-stat	-0.144***	-0.029	-0.115*
	(0.032)	(0.057)	(0.065)
investment beta t-stat	-0.182**	-0.001	-0.181*
	(0.079)	(0.070)	(0.106)
market beta t-stat	-0.145***	0.004	-0.149
	(0.025)	(0.116)	(0.113)
size beta t-stat	-0.119**	-0.072***	-0.047
	(0.047)	(0.026)	(0.046)
R^2	-0.151***	-0.042	-0.109
	(0.027)	(0.122)	(0.122)
manager tenure (years)	-0.102**	-0.076	-0.025
	(0.042)	(0.061)	(0.048)
volatility of flows	-0.115*	-0.089***	-0.025
	(0.061)	(0.032)	(0.051)
flows	-0.088*	-0.062	-0.026
	(0.051)	(0.059)	(0.068)
turnover ratio	-0.074	-0.051	-0.023
	(0.087)	(0.033)	(0.089)
momentum beta t-stat	-0.088	-0.104**	0.016
	(0.074)	(0.043)	(0.077)
expense ratio	-0.085	-0.075**	-0.010
	(0.066)	(0.031)	(0.056)
TNA (USD mill.)	-0.074**	-0.073	-0.001
	(0.030)	(0.049)	(0.034)

Table II.7: **Monthly out-of-sample alpha (in %) of univariate sorts - 36 months**

This table presents univariate sorts on 17 fund characteristics used in DGNS. For the three-year period ending in year $t+3$, funds are sorted into deciles based on the characteristic measured in December of year t and equally weighted portfolio returns are calculated. A long–short portfolio is formed by taking a long position in the tenth decile and a short position in the first decile with a holding period of 36 months. The table reports the top decile (column 1), bottom decile (column 2) and the spanning portfolio (column 3). As in Section 3 the out-of-sample alphas are obtained by regressing excess monthly portfolio returns on the Fama and French (2015) 5-factor model (FF5) extended by the momentum factor of Carhart (1997) (FF5+MOM). The out-of-sample period spans from January 1991 to December 2020 (360 observations). HAC-robust standard errors, as suggested by Newey and West (1987), are reported in parentheses. The number of lags is set to 12. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5% and 1% level, respectively. To improve the readability of the table, we sort fund characteristics according to their significance level of the spanning portfolio (long-short).

	top decile	**bottom decile**	**Long-short**
alpha t-stat	-0.017	-0.151***	0.134**
	(0.054)	(0.041)	(0.059)
investment beta t-stat	-0.246***	-0.023	-0.223**
	(0.070)	(0.068)	(0.107)
age (months)	-0.115***	-0.070**	-0.046*
	(0.040)	(0.034)	(0.025)
TNA (USD mill.)	-0.051*	-0.098**	0.047*
	(0.029)	(0.047)	(0.028)
profitability beta t-stat	-0.133***	-0.026	-0.107
	(0.028)	(0.068)	(0.070)
value beta t-stat	-0.149***	-0.011	-0.138
	(0.050)	(0.071)	(0.091)
realized alpha	-0.013	-0.139*	0.126
	(0.072)	(0.073)	(0.091)
market beta t-stat	-0.166***	-0.026	-0.140
	(0.029)	(0.111)	(0.102)
value added	-0.037	-0.124**	0.088
	(0.053)	(0.058)	(0.069)
R^2	-0.172***	-0.035	-0.137
	(0.028)	(0.132)	(0.124)
volatility of flows	-0.127**	-0.078**	-0.048
	(0.064)	(0.036)	(0.060)
turnover ratio	-0.109*	-0.063*	-0.046
	(0.065)	(0.033)	(0.062)
manager tenure (years)	-0.099**	-0.126***	0.027
	(0.040)	(0.047)	(0.037)
expense ratio	0.032	-0.063**	0.095
	(0.158)	(0.028)	(0.152)
momentum beta t-stat	-0.136*	-0.123***	-0.013
	(0.074)	(0.046)	(0.087)
flows	-0.101*	-0.107**	0.007
	(0.051)	(0.048)	(0.053)
size beta t-stat	-0.114**	-0.115***	0.001
	(0.044)	(0.026)	(0.047)

Appendix to

"Does machine learning really help to select mutual funds with positive alpha"

by

Jürg Fausch Moreno Frigg Thomas Johann Emil Mussbach
Christian Westheide Wolfgang Drobetz

Appendix II.A. Sample construction

To align our work as closely as possible with DGNS, we follow their documented procedures to the greatest extent feasible to create the sample. Nonetheless, an exact replication is not possible as the repository lacks both the raw CRSP data and the code used to build the sample. It only contains the final analysis files derived from data that has already been filtered, merged and preprocessed. Consequently, we must make our own design choices in cases where no unique implementation of a filtering step is specified. Because reproducing the sample requires additional assumptions, we describe the relevant steps in greater detail in this appendix. For further information on the DGNS data and methodology, we refer readers to the original paper.

Consistent with DGNS, we use data from the CRSP Survivor-Bias-Free US Mutual Fund database. We access the various data tables from CRSP to create a panel dataset of monthly US mutual funds. Following DGNS, the main sample spans January 1980 through December 2020. In our extension, we expand the data set to December 2024.

Starting with the tables provided by CRSP, we implement several adjustments and proceed with specific decisions, outlined below:

Monthly Total Return: This table contains monthly returns (mret). There are a small number of data errors in the CRSP return data. We examine the most extreme returns (top and bottom 1% per year-month) by comparing them with data from Morningstar and Lipper. In 64 instances (less than 0.002% of our sample), this process reveals a false return (oftentimes related to fund mergers) and we replace those returns accordingly. In 11 instances, it is unclear which of the two data sources (if any) reports correct monthly returns. We set those returns to missing. Finally, we drop 4 funds[1] as they were involved in a complex merger that lead to uninformative monthly returns over several months.

Monthly Total Net Assets: This table contains monthly total net assets (mtna). Considering that DGNS's sample starts in 1980 and that some variables require monthly net asset data, it is important to address the issue that regular monthly data on net assets only became available in the CRSP database starting in 1991. DGNS do not specify how they deal with this issue. We fill in missing observations of TNA using the latest available value of TNA for each share class.

[1] Their CRSP fundnos are 22169, 26380, 26378 and 26379.

Fund Summary: This table contains quarterly or annual data on several fund characteristics, including the fund name (fund_name), the date a manager assumed responsibility (mgr_dt), the expense ratio (exp_ratio), the turnover ratio (turn_ratio), the CRSP objective code (crsp_obj_cd) and the ratio of TNA invested in equities (per_com and per_pref). First, since DGNS do not provide specific details on how they implemented the equity ratio filter, we calculate the equity ratio as the total percentage invested in common (per_com) and preferred stocks (per_pref). To derive a monthly time series, we assume that the allocation stays the same from the effective date of the asset allocations (asset_dt) until the asset date of the next observation. If there are no following observations, we assume that the allocation stays the same until the termination date of the share class. Second, as we create a monthly panel and the fund summary information is reported on an annual basis, we fill missing monthly observations of the respective year. If more than 11 consecutive observations are missing, we assign "NA" starting from the 12^{th} missing observation onward.

Front Load & Rear Load: These tables report the corresponding loads (front_load and rear_load) along with their effective date ranges (begdt and enddt). Front loads may vary depending on the investment amount, while rear loads depend on the holding period. However, DGNS do not elaborate on the specific methods they use to select no-load funds from the database.

As our analysis covers both retail and institutional share classes, we adopt the highest available front load, as retail investors are unlikely to invest above thresholds that would result in lower front loads.

For 74.8% of fund-month observations, only a single rear load is available. For the remaining observations, we pick the load reported for a holding period of twelve months (10.6% of observations). This is consistent with the investment horizon of twelve months used by DGNS in their investment strategies. However, if this specific period is not available, we pick the value reported for the next longer holding period (2% of observations). If this is not available, we pick the observation with missing investment horizon[2] (12.6% of observations). Finally, if this is not available, we pick the holding period below 12 months, but closest to 12 months (0.003% of observations). Based on the effective date ranges, we construct monthly time series of loads for each fund.

The CRSP database contains a substantial number of missing load values, making the assumptions and implementation choices in this regard particularly important. During the sample period of DGNS, 84.5% of monthly observations have

[2]time_period equals -99, 9999, or 99999.

either a missing front or rear load. Carrying forward the last non-missing value still results in 67.5% of monthly observations having either a missing front or rear load. Therefore, as, e.g., Roussanov, Ruan, and Wei (2021), we treat any missing load values as zero, effectively classifying funds without reported loads as no-load funds.

Factor Loadings: In addition to the CRSP Mutual Fund data, we collect factors for the alpha estimation. For the original sample period, we use the factors provided by DGNS in their data repository. For the extended time period until 2024, Fama/French 3 factors and the momentum factor are retrieved from Ken French's data library[3]; the liquidity factor is retrieved from Robert Stambaugh's homepage.[4] It has been shown that these factor returns can vary considerably over different versions (Akey, Robertson, and Simutin, 2026). We decide to use the most recently available information for the 2024 sample.

Table II.A1: **Sample selection**

This table gives an overview of the different steps taken to replicate DGNS's sample of active U.S. equity mutual funds for the period from January 1980 to December 2020 [December 2024]. The figures show the remaining monthly observations and unique share classes after each data cleaning step.

Data filter	Observations		Share Classes	
	2020	**2024**	**2020**	**2024**
U.S. mutual funds	7,936,442	9,412,047	67,428	72,037
U.S. equity mutual funds	2,941,619	3,543,599	28,948	31,425
Exclude funds with loads	2,007,854	2,516,235	22,182	24,410
Exclude passively managed funds and ETFs	1,731,711	2,138,206	19,213	20,726
Exclude funds with equity ratio not greater than 70%	1,433,027	1,771,315	15,896	16,889
Exclude funds not at least 36 months of age	1,128,238	1,438,567	12,885	14,120
Exclude fund observations before at least USD 5m in assets	1,008,457	1,288,466	10,669	11,733

Filtering Procedure: The construction of the data set follows the selection criteria outlined in DGNS and focuses on US domestic-equity mutual funds at the share class level. The sample includes institutional as well as retail share classes and spans from January 1980 to December 2020. Since the authors do not provide detailed information on how to select equity funds, we follow Kaniel et al. (2023) and Doshi, Elkamhi, and Simutin (2015) by using the CRSP objective code to identify these funds.[5] In line with DGNS, we restrict the analysis to share classes that charge

[3] https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html - here we download the July 2025 version of the data

[4] https://finance.wharton.upenn.edu/~stambaug/ - retrieved in March 25

[5] In line with Doshi, Elkamhi, and Simutin (2015) we require that the CRSP objective code begins with the letters "ED" followed by a "C", "Y", or "S", and that the third and fourth letter are not equal to "YH" or "YS". In contrast to Doshi, Elkamhi, and Simutin (2015), we also include sector funds (third letter "S") to be consistent with DGNS. Additionally, we exclude funds with Strategic Insight Objective Code "OPI".

no front-end or back-end loads to ensure that portfolio rebalancing does not incur any costs. If no information on loads is provided by CRSP, we follow Roussanov, Ruan, and Wei (2021) and treat any missing values as zero. In addition, to select the relevant share classes, we apply the same filters as DGNS, which are commonly used in the mutual fund literature. First, we restrict the sample to share classes of actively managed mutual funds, thereby excluding ETFs and passive mutual funds.[6] Finally, we consider a fund an index fund if its CRPS objective code equals "EDCL". Second, we consider only share classes that invest more than 70 percent of their total net assets (TNA) in equities. Third, to avoid incubation bias, we exclude observations of a share class before it reaches 36 months of age and before the first observation with at least USD 5 million of TNA (Elton, Gruber, and Blake, 2001; Evans, 2010).

The resulting dataset includes 10,669 [11,733] share classes for the period between 1980 and 2020 [2024]. For the 2020 sample, 74.6% of those funds are style-based, 21.1% cap-based and 4.4% sector-based (in terms of total TNA), which is very close to the distribution found by DGNS.

[6] In line with DGNS, we use the index fund identifier from CRSP and, in cases where this identifier is missing, we refer to fund names following the approach suggested by Gil-Bazo and Ruiz-Verdú (2009): We consider a fund an index fund if its name contains "Index","Idx","Ix","Indx","NASDAQ","Nasdaq","Dow","Mkt","DJ","S&P","500" or "BARRA".

Appendix II.B. Reproduction and Replication using alternative factor models

Table II.B1: **Monthly out-of-sample alpha (in %) - Reproduction**

This table reports monthly out-of-sample alphas (in %), net of all costs, for the equally weighted top-decile portfolios in Table 3 of DGNS for the asset pricing models not shown in the main text. The out-of-sample alphas are obtained by regressing excess monthly portfolio returns net of all costs against the Fama and French (1993) three-factor model augmented with the momentum factor of Carhart (1997) (FF3+MOM), Fama and French (2015) five factors (FF5) and the FF5 model augmented with momentum and with the liquidity risk factor of Pástor and Stambaugh (2003) (FF5+MOM+LIQ). The out-of-sample period spans from January 1991 to December 2020 (360 observations). HAC-robust standard errors, as suggested by Newey and West (1987), are reported in parentheses. As in DGNS, the number of lags is set to 12. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5% and 1% level, respectively.

	DGNS published	**Data repository**	**Reproduction**
		FF3+MOM	
Gradient boosting	0.178**	0.105	-0.020
	(0.077)	(0.076)	(0.073)
Random forest	0.210**	0.098	-0.033
	(0.086)	(0.083)	(0.082)
Elastic net	0.044	0.037	-0.052
	(0.065)	(0.074)	(0.072)
OLS	0.056	0.046	-0.043
	(0.063)	(0.073)	(0.071)
Equally Weighted	-0.018	-0.009	-0.075*
	(0.045)	(0.047)	(0.041)
Asset Weighted	-0.043	-0.046	-0.093***
	(0.036)	(0.036)	(0.033)
		FF5	
Gradient boosting	0.222***	0.224***	0.105
	(0.085)	(0.086)	(0.077)
Random forest	0.263***	0.264***	0.145
	(0.097)	(0.097)	(0.091)
Elastic net	0.075	0.075	-0.005
	(0.067)	(0.067)	(0.064)
OLS	0.085	0.084	0.005
	(0.065)	(0.065)	(0.062)
Equally Weighted	-0.007	-0.006	-0.072*
	(0.045)	(0.045)	(0.039)
Asset Weighted	-0.033	-0.030	-0.074**
	(0.035)	(0.035)	(0.031)
		FF5+MOM+LIQ	
Gradient boosting	0.198**	0.200**	0.083
	(0.081)	(0.082)	(0.074)
Random forest	0.226**	0.22**	0.107
	(0.089)	(0.088)	(0.083)
Elastic net	0.098	0.098	0.016
	(0.068)	(0.068)	(0.065)
OLS	0.109*	0.108	0.027
	(0.066)	(0.066)	(0.062)
Equally Weighted	-0.017	-0.016	-0.081**
	(0.045)	(0.045)	(0.039)
Asset Weighted	-0.036	-0.033	-0.077**
	(0.036)	(0.036)	(0.033)

Table II.B2: **Monthly out-of-sample alpha (in %) - Replication**

This table reports monthly out-of-sample alphas (in %), net of all costs, for the equally weighted top-decile portfolios in Table 3 of DGNS for the asset pricing models not shown in the main text. The out-of-sample alphas are obtained by regressing excess monthly portfolio returns net of all costs against the Fama and French (1993) three-factor model augmented with the momentum factor of Carhart (1997) (FF3+MOM), Fama and French (2015) five factors (FF5) and the FF5 model augmented with momentum and with the liquidity risk factor of Pástor and Stambaugh (2003) (FF5+MOM+LIQ). The out-of-sample period spans from January 1991 to December 2020 (360 observations). HAC-robust standard errors, as suggested by Newey and West (1987), are reported in parentheses. As in DGNS, the number of lags is set to 12. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5% and 1% level, respectively.

	DGNS published	**Data repository**	**Replication**
		FF3+MOM	
Gradient boosting	0.178**	0.105	-0.049
	(0.077)	(0.076)	(0.081)
Random forest	0.210**	0.098	-0.090
	(0.086)	(0.083)	(0.083)
Elastic net	0.044	0.037	-0.061
	(0.065)	(0.074)	(0.074)
OLS	0.056	0.046	-0.061
	(0.063)	(0.073)	(0.073)
Equally Weighted	-0.018	-0.009	-0.079**
	(0.045)	(0.047)	(0.040)
Asset Weighted	-0.043	-0.046	-0.092**
	(0.036)	(0.036)	(0.031)
		FF5	
Gradient boosting	0.222***	0.224***	0.105
	(0.085)	(0.086)	(0.087)
Random forest	0.263***	0.264***	0.050
	(0.097)	(0.097)	(0.083)
Elastic net	0.075	0.075	0.051
	(0.067)	(0.067)	(0.066)
OLS	0.085	0.084	0.046
	(0.065)	(0.065)	(0.066)
Equally Weighted	-0.007	-0.006	-0.071*
	(0.045)	(0.045)	(0.037)
Asset Weighted	-0.033	-0.030	-0.073**
	(0.035)	(0.035)	(0.030)
		FF5+MOM+LIQ	
Gradient boosting	0.198**	0.200**	0.074
	(0.081)	(0.082)	(0.081)
Random forest	0.226**	0.22**	0.025
	(0.089)	(0.088)	(0.079)
Elastic net	0.098	0.098	0.060
	(0.068)	(0.068)	(0.065)
OLS	0.109*	0.108	0.055
	(0.066)	(0.066)	(0.065)
Equally Weighted	-0.017	-0.016	-0.081**
	(0.045)	(0.045)	(0.038)
Asset Weighted	-0.036	-0.033	-0.074**
	(0.036)	(0.036)	(0.031)

Appendix II.C. Replication based on intentionally retaining the survivorship bias present in DeMiguel et al. (2023)

Table II.C1: **Monthly out-of-sample alpha (in %) - Replication with survivorship bias**

Column 1 of this table reports the monthly out-of-sample alphas (in %) net of all costs of the equally weighted top decile portfolios from Table 3 in DGNS. Column 2 shows the results of the reproduction. Column 3 reports the monthly out-of-sample alphas (in %) net of all costs of the equally weighted top decile portfolios based on the predicted alpha distribution. The results presented in this table are obtained using the replication methodology outlined in Section 3, while retaining the survivorship bias present in DGNS. The out-of-sample alphas are obtained by regressing excess monthly portfolio returns on the main factor model used in DGNS, i.e., the Fama and French (2015) 5-factor model (FF5) extended by the momentum factor of Carhart (1997) (FF5+MOM). The out-of-sample period spans from January 1991 to December 2020 (360 observations). HAC-robust standard errors, as suggested by Newey and West (1987), are reported in parentheses. As in DGNS, the number of lags is set to 12. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5% and 1% level, respectively.

	DGNS published	**Reproduction**	**Replication (biased)**
Gradient boosting	0.197**	0.083	0.046
	(0.080)	(0.073)	(0.072)
Random forest	0.224**	0.106	0.042
	(0.087)	(0.082)	(0.083)
Elastic net	0.091	0.009	0.073
	(0.069)	(0.065)	(0.071)
OLS	0.101	0.019	0.066
	(0.066)	(0.062)	(0.070)
Equally Weighted	-0.018	-0.082**	-0.084**
	(0.045)	(0.039)	(0.038)
Asset Weighted	-0.037	-0.077**	-0.077**
	(0.035)	(0.032)	(0.032)

Appendix II.D. Replication of table 4 to 7 in DeMiguel et al. (2023)

This appendix contains the tables discussed in Subsection 3.2.2 in the main text.

Table II.D1: **Out-of-sample alpha with respect to OLS**

This table reports monthly out-of-sample alphas (in %), net of all costs, for portfolios that are long in funds selected by one of the methods considered (gradient boosting, random forests, elastic net, equally weighted, or asset weighted) and short in funds selected by OLS. For example, "gradient boosting minus OLS" denotes a long-short portfolio that is long in the gradient-boosting top-decile portfolio and short in the OLS top-decile portfolio. The out-of-sample alphas are obtained by regressing monthly portfolio returns on the main factor model used in DGNS, i.e., the Fama and French (2015) 5-factor model (FF5) extended by the momentum factor of Carhart (1997) (FF5+MOM). The out-of-sample period spans January 1991 to December 2020 (360 observations) and, using the extended data set, January 1991 to December 2024 (408 observations), as reported in the last column. HAC-robust standard errors, as suggested by Newey and West (1987), are reported in parentheses. As in DGNS, the number of lags is set to 12. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5% and 1% level, respectively.

	DGNS	**Replication**	**Replication (2024)**
GB minus OLS	0.096**	0.025	-0.021
	(0.043)	(0.052)	(0.057)
RF minus OLS	0.123**	-0.021	-0.002
	(0.050)	(0.061)	(0.053)
EN minus OLS	-0.010	0.006	-0.001
	(0.011)	(0.005)	(0.005)
EW minus OLS	-0.119**	-0.127**	-0.127**
	(0.048)	(0.056)	(0.058)
AW minus OLS	-0.137***	-0.119*	-0.115*
	(0.052)	(0.063)	(0.063)

Table II.D2: **Out-of-sample mean excess return and risk**

This table reports for each fund portfolio the following monthly out-of-sample performance metrics: mean excess returns net of all costs; standard deviation; Sharpe ratio (mean excess return divided by the standard deviation); Sortino ratio (mean excess return divided by the semi-deviation); information ratio (alpha net of all costs relative to the FF5+MOM model divided by idiosyncratic volatility); maximum drawdown; and value-at-risk (VaR) based on the historical simulation method at the 99% confidence level. The final column reports the average annual portfolio turnover. Panel (a) reports the results published in DGNS. Panel (b) presents the replication results based on an out-of-sample period spanning January 1991 to December 2020 (360 observations) and panel (c) shows the results for the extended out-of-sample period through December 2024 (408 observations).

Panel (a): DGNS - Published

	Mean	Standard deviation	Sharpe ratio	Sortino ratio	Information ratio	Maximum drawdown	VaR 99%	Turnover
Gradient boosting	0.90%	4.71%	0.192	0.292	0.174	50.3%	12.0%	1.476
Random forest	0.93%	4.96%	0.188	0.290	0.163	55.4%	13.4%	1.410
Elastic net	0.81%	4.81%	0.168	0.249	0.075	58.3%	12.4%	1.219
OLS	0.82%	4.80%	0.170	0.253	0.083	58.5%	12.3%	1.218
Equally weighted	0.78%	4.39%	0.178	0.263	-0.029	51.4%	10.2%	0.414
Asset weighted	0.73%	4.42%	0.166	0.243	-0.069	52.8%	10.7%	0.369

Panel (b): Replication - Sample through 2020

	Mean	Standard deviation	Sharpe ratio	Sortino ratio	Information ratio	Maximum drawdown	VaR 99%	Turnover
Gradient boosting	0.81%	5.08%	0.159	0.236	0.057	60.9%	14.4%	1.446
Random forest	0.77%	5.20%	0.148	0.217	0.019	67.1%	15.3%	1.383
Elastic net	0.74%	4.94%	0.150	0.220	0.045	63.0%	14.1%	1.123
OLS	0.74%	4.92%	0.151	0.221	0.040	62.4%	13.9%	1.137
Equally weighted	0.72%	4.44%	0.163	0.236	-0.142	52.7%	10.5%	0.354
Asset weighted	0.69%	4.41%	0.156	0.225	-0.148	55.5%	11.0%	0.308

Panel (c): Replication - Sample through 2024

	Mean	Standard deviation	Sharpe ratio	Sortino ratio	Information ratio	Maximum drawdown	VaR 99%	Turnover
Gradient boosting	0.75%	4.98%	0.150	0.220	0.007	59.1%	12.7%	1.422
Random forest	0.77%	5.17%	0.149	0.220	0.022	63.7%	14.3%	1.401
Elastic net	0.71%	4.98%	0.143	0.210	0.026	61.9%	13.5%	1.139
OLS	0.72%	4.98%	0.144	0.211	0.027	61.6%	13.4%	1.149
Equally weighted	0.70%	4.47%	0.158	0.230	-0.167	52.7%	10.4%	0.326
Asset weighted	0.68%	4.42%	0.153	0.221	-0.169	55.5%	10.8%	0.297

Table II.D3: **Out-of-sample alpha of double-sorted portfolios**

This table reports monthly out-of-sample alphas (in %), net of all costs, for the portfolios obtained by double sorting funds on pairs of measures of past performance and fund activeness, following the procedure outlined in DGNS. The portfolio alphas reported in the table are obtained by regressing excess monthly portfolio returns on the main factor model used in DGNS, i.e., the Fama and French (2015) 5-factor model (FF5) extended by the momentum factor of Carhart (1997) (FF5+MOM). The out-of-sample period spans January 1991 to December 2020 (360 observations) and, using the extended data set, January 1991 to December 2024 (408 observations), as reported in the last column. HAC-robust standard errors, as suggested by Newey and West (1987), are reported in parentheses. As in DGNS, the number of lags is set to 12. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5% and 1% level, respectively.

Double sort on	**DGNS**	**Replication**	**Replication (2024)**
Alpha t-stat and R^2	0.177*	0.071	0.016
	(0.090)	(0.077)	(0.073)
Alpha t-stat and market beta t-stat	0.207**	0.080	0.031
	(0.099)	(0.087)	(0.079)
Value added and R^2	0.113	0.029	-0.018
	(0.095)	(0.082)	(0.078)
Value added and market beta t-stat	0.137	0.047	0.009
	(0.102)	(0.086)	(0.081)

Table II.D4: **Out-of-sample alpha of fund portfolios under different market conditions**

This table reports monthly out-of-sample alphas (in %), net of all costs, for the top-decile fund portfolios obtained using gradient boosting and random forests under different market conditions. The portfolio alphas reported in the table are obtained by regressing excess monthly portfolio returns on the main factor model used in DGNS, i.e., the Fama and French (2015) 5-factor model (FF5) extended by the momentum factor of Carhart (1997) (FF5+MOM) as well as indicator variables for expansions and recessions. Each panel shows the results for high and low investor sentiment. Expansions and recessions are defined according to the NBER convention. The high (low) investor sentiment indicator equals one if investor sentiment, as defined by (Baker and Wurgler, 2006, 2007), is above (below) the median over the period July 1965 to December 2020. For the data extended to 2024 combined with investor sentiment, we use data only through the end of 2023, as the Baker and Wurgler (2006, 2007) data availability ends in 2023. The out-of-sample period spans January 1991 to December 2020 (360 observations) and, using the extended data set, January 1991 to December 2023 (396 observations). HAC-robust standard errors, as suggested by Newey and West (1987), are reported in parentheses. As in DGNS, the number of lags is set to 12. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5% and 1% level, respectively.

	Business Cycle			**Investor Sentiment**		
	Expansion	Recession	Exp.–Rec.	High	Low	High–Low
Panel (a): DGNS						
Gradient boosting	0.179**	0.375	-0.196	0.233***	0.150	0.083
	(0.082)	(0.228)	(0.226)	(0.085)	(0.109)	(0.106)
Random forest	0.202**	0.445*	-0.243	0.266***	0.169	0.097
	(0.087)	(0.248)	(0.236)	(0.102)	(0.118)	(0.131)
Panel (b): Replication						
Gradient boosting	0.067	0.113	-0.047	0.052	0.095	-0.043
	(0.082)	(0.183)	(0.186)	(0.094)	(0.098)	(0.108)
Random forest	0.038	-0.080	0.118	-0.009	0.074	-0.083
	(0.079)	(0.191)	(0.194)	(0.104)	(0.078)	(0.105)
Panel (c): Replication 2024						
Gradient boosting	0.006	-0.097	0.103	-0.036	0.045	-0.081
	(0.079)	(0.163)	(0.168)	(0.102)	(0.079)	(0.110)
Random forest	0.019	0.053	-0.035	-0.033	0.097	-0.129
	(0.078)	(0.196)	(0.197)	(0.099)	(0.082)	(0.106)

Chapter III

Machine Learning Design Choices and Equity Mutual Fund Performance

1. Introduction

In recent years, machine learning (ML) methods have become increasingly popular in empirical finance research. A growing number of studies apply these techniques to predict financial variables such as equity risk premiums (Gu, Kelly, and Xiu, 2020; Leippold, Wang, and Zhou, 2022), bond risk premiums (Bianchi, Büchner, and Tamoni, 2021), option returns (Bali et al., 2023), mutual fund performance (DeMiguel et al., 2023; Kaniel et al., 2023; Fausch et al., 2025a) and mutual fund flows (Fausch et al., 2025b). In contrast to traditional linear frameworks such as OLS, the increasing adoption of ML is driven by their ability to capture nonlinearities and complex interactions among predictors without requiring ex ante functional form specification. As a result, this modeling flexibility may enhance out-of-sample forecasting accuracy.

At the same time, the increasing use of ML methods introduces a fundamental challenge: the expanding set of design choices. Implementing these models requires researchers to make numerous methodological decisions[1], each of which may materially influence empirical results. In this context, Chen, Hanauer, and Kalsbach (2024) show that alternative ML design choices can lead to substantial variation in predictive performance in the cross-section of stock returns, highlighting the sensitivity of ML-based findings to the specific modeling choices made.

This sensitivity is particularly relevant in settings where the underlying evidence is already inconclusive, such as the literature on equity mutual fund performance (alpha) predictability using ML methods. In this context, DeMiguel et al. (2023) report that portfolios based on fund selection using ML generate statistically significant long-only alpha of up to 2.69% (on an annual basis). In contrast, Kaniel

[1]For example, researchers must select among a wide range of candidate algorithms.

et al. (2023) find a significant alpha of 2.04%, but only for a long–short strategy. In a recent study, Fausch et al. (2025a) show that the results presented by DeMiguel et al. (2023) are affected by a forward-looking bias, leading to an overestimation of portfolio alpha. After correcting this bias, the long-only alpha becomes insignificant, while long–short strategies remain significantly profitable. Importantly, this finding is not limited to nonlinear ML methods but also holds for linear models such as OLS and elastic net.

Building on this sensitivity and the mixed evidence in the existing literature, I systematically evaluate how design choices affect the performance predictability of U.S. equity mutual funds. To this end, I analyze 2,052 ML model specifications across seven design choice categories. Specifically, I consider the choice of algorithm, data imputation method, training window length, training frequency as well as three decisions related to hyperparameter tuning (search strategy, tuning procedure and tuning frequency), and assess their impact on the alpha of ML-generated mutual fund portfolios. Overall, my empirical results show that only two specifications (0.1% of all possible combinations) produce a statistically significant positive long-only alpha at the 10% level over the out-of-sample period from January 1994 to December 2024. This finding suggests limited evidence for a systematic and robust long-only alpha predictability, consistent with the results reported in Fausch et al. (2025a). Economically, the dispersion in performance is substantial, ranging from an annual alpha of 1.76% for the best-performing specification to -2.51% for the worst one, highlighting the pronounced sensitivity of estimated alphas to ML design choices.

Of the seven categories analyzed, the choice of algorithm and the training window were found to be the most influential determinants of ML-based portfolio performance. Nonlinear methods, such as random forest and gradient boosting, achieve positive median annualized alphas of 0.55% and 0.45%, respectively, whereas linear models (OLS and elastic net) produce negative median alphas of -0.19% and -0.43%. In addition, performance generally improves with the amount of training data: the expanding window approach delivers the highest median alpha (0.81%), while a rolling window with 12 months of training data results in a negative median alpha of -0.64%.

In order to quantify the uncertainty stemming from the researchers' choices in a systematic way, I draw on the growing body of literature on non-standard errors (NSE) in finance (see, among others, Menkveld et al., 2024; Soebhag, Van Vliet, and Verwijmeren, 2024; Walter, Weber, and Weiss, 2024; Chen, Hanauer, and Kalsbach, 2024). Whereas traditional statistical inference focuses on standard errors

(SEs) driven by the stochastic nature of the data-generating process, recent studies have highlighted that empirical findings can vary considerably depending on the decisions made in the research design. Within this framework, the NSE measures the dispersion of outcomes across alternative specifications and thus provides a direct measure of specification uncertainty. Across the 2,052 specifications used in this study, the overall NSE amounts to a sizeable 0.83% annualized alpha, highlighting that conclusions about ML-based predictability are sensitive to the chosen research design.

Building on this, I compare the uncertainty induced by research design choices with that arising from the data-generating process by computing the ratio of NSE to SE. The resulting ratio of 0.86 indicates that sampling variation remains slightly more important than specification uncertainty in this setting. This suggests that, although researcher choices are important, they do not completely dominate statistical uncertainty in the context of mutual fund performance prediction using ML. The estimate lies within the range reported by Menkveld et al. (2024) (0.59 to 1.99), but is lower than the values documented by Soebhag, Van Vliet, and Verwijmeren (2024) and Walter, Weber, and Weiss (2024) (ratios of 1.06 and 1.10, respectively).[2] Finally, the study most closely related to my work, Chen, Hanauer, and Kalsbach (2024), reports a substantially higher ratio of 1.59 in the context of machine learning-based stock return prediction.

Contrary to the findings of Chen, Hanauer, and Kalsbach (2024), who concluded that no single design choice primarily drives the high NSE, the empirical evidence presented in this paper suggests the opposite. Specifically, I show that linear models, such as elastic net and OLS, exhibit increased sensitivity to alternative specifications. Consequently, these models appear to be less robust in comparison to nonlinear algorithms (random forest and gradient boosting). Moreover, employing a validation set instead of a cross-validation approach for hyperparameter tuning leads to a higher NSE-to-SE ratio. This finding is particularly relevant, as both linear models and validation-based approaches are among the most commonly used methods in the financial machine learning literature.

Overall, my paper contributes to two strands of the literature. First, I contribute to the growing body of research on financial machine learning (Gu, Kelly, and Xiu, 2020; Bianchi, Büchner, and Tamoni, 2021; Leippold, Wang, and Zhou, 2022; Bali et al., 2022, 2023; Fausch et al., 2025b), and more specifically to the literature on predicting mutual fund performance using ML methods (DeMiguel et al., 2023; Kaniel et al., 2023; Fausch et al., 2025a). Building on this work, I systematically evaluate

[2]Using conventional SEs instead of HAC-robust SEs increases the ratio in this paper to 1.06.

the impact of more than 2,000 ML specifications on alpha prediction, thereby showing that identifying long-only alpha ex-ante remains a highly challenging task, even with access to modern ML techniques. At the same time, I document substantial dispersion between the best- and worst-performing specifications, suggesting that future research might benefit from reporting results across a wider range of model configurations. Second, the paper contributes to the emerging literature on non-standard errors (Menkveld et al., 2024; Soebhag, Van Vliet, and Verwijmeren, 2024; Walter, Weber, and Weiss, 2024; Chen, Hanauer, and Kalsbach, 2024) in finance. To the best of my knowledge, this is the first study to assess researcher-induced uncertainty at the intersection of machine learning and mutual fund performance predictability. Although closely related, my work differs from Chen, Hanauer, and Kalsbach (2024) with regards to two important aspects. First, while Chen, Hanauer, and Kalsbach (2024) examine stock return predictability using ML, I focus on mutual fund performance predictability, a setting where the underlying evidence for finding long-only alpha ex-ante is mixed and inconclusive. Second, I extend their framework by including additional design choice categories that are of considerable importance when implementing machine learning techniques. In this context, I evaluate choices related to data imputation, training frequencies and hyperparameter tuning. I further contribute to this literature by demonstrating that commonly used ML choices tend to exhibit particularly high NSE-to-SE ratios. This finding emphasizes that even widely adopted modeling decisions can be a key source of uncertainty induced by researchers and can therefore materially influence empirical outcomes.

The remainder of the paper is structured as follows. Section 2 describes the data, mutual fund characteristics and the prediction framework. Section 3 outlines the seven categories of machine learning design choices and the corresponding specifications. Section 4 presents the empirical results, including the impact of design choices on portfolio alpha and the estimation of non-standard errors. Finally, Section 5 concludes.

2. Data, fund characteristics and prediction framework

This section provides a detailed description of the sample and the characteristics used to predict mutual fund performance. It also describes the data preprocessing steps and the forecasting procedure. Furthermore, it outlines the methodology used to assess the impact of ML design choices on portfolio performance as well

as researcher-induced uncertainty, which is measured via non-standard errors. The data used in this analysis are obtained from the Center for Research in Security Prices (CRSP) Survivor-Bias-Free US Mutual Fund database.

2.1. Sample data

To examine the impact of various machine learning design choices on fund performance prediction, I rely on monthly data of U.S. equity mutual funds that primarily invest in the domestic market. Following DeMiguel et al. (2023) and Fausch et al. (2025a), the analysis is conducted at the share class level, as investors ultimately allocate capital to share classes rather than to funds. The sample spans the period from January 1980 to December 2024 and includes both institutional and retail share classes. Similar to these papers, I restrict the sample to actively managed equity mutual funds, thereby excluding both ETFs and passive funds.[3] In addition, I include only share classes that invest at least 70 percent of their portfolio in equities and that charge no front- or back-end loads (DeMiguel et al., 2023; Fausch et al., 2025a). Finally, to address the well-documented incubation bias in mutual fund research, I further exclude observations of share classes prior to reaching 36 months of age and before surpassing USD 15 million in total net assets for the first time (Elton, Gruber, and Blake, 2001; Evans, 2010; Doshi, Elkamhi, and Simutin, 2015; Kaniel et al., 2023; Fausch et al., 2025b). The final data set consists of 11,687 unique share classes.

2.2. Mutual fund characteristics

To predict the performance of equity mutual funds, I use the 17 characteristics proposed by DeMiguel et al. (2023) which are either readily accessible or easy to compute based on CRSP data.

For each share class, I retrieve information on monthly net returns and total net assets (TNA) from CRSP. Based on the return data, I estimate time-series regressions using a rolling estimation window of 36 months[4], regressing excess returns[5] on the five-factor asset pricing model proposed by Fama and French (2015) augmented

[3]Since the relevant identifiers in CRSP for these share classes are available starting in 2003, I follow the procedure outlined in Fausch et al. (2025a) and exclude passive share classes based on specific keywords in the fund name, as suggested by Gil-Bazo and Ruiz-Verdú (2009).

[4]Following DeMiguel et al. (2023), a minimum of 30 consecutive return observations is required to estimate the regression.

[5]The excess return is computed by subtracting the one-month T-bill yield obtained from the data library of Kenneth French from the fund's net return.

with the momentum factor (Jegadeesh and Titman, 1993; Carhart, 1997).[6] This yields monthly t-statistics for alpha, as well as for the market, size, value, profitability, investment, and momentum betas, along with R^2 as a measure of activeness (Amihud and Goyenko, 2013). The realized alpha of share class i, denoted by $\alpha^r_{i,t}$, is computed as

$$\begin{aligned}\alpha^r_{i,t} = R^e_{i,t} - \hat{b}_{i,t-1}(R_{M,t} - R_{f,t}) - \hat{s}_{i,t-1}SMB_t - \hat{h}_{i,t-1}HML_t \\ -\hat{r}_{i,t-1}RMW_t - \hat{c}_{i,t-1}CMA_t - \hat{m}_{i,t-1}MoM_t,\end{aligned} \tag{III.1}$$

where the factor loadings ($\hat{b}_{i,t-1}$, $\hat{s}_{i,t-1}$, $\hat{h}_{i,t-1}$, $\hat{r}_{i,t-1}$, $\hat{c}_{i,t-1}$ and $\hat{m}_{i,t-1}$) correspond to the estimates obtained from the rolling window regression from $t-36$ to $t-1$.

Monthly flows are computed as the relative change in total net assets (TNA) for share class i, adjusted for returns in month t. In contrast to DeMiguel et al. (2023), I estimate the standard deviation of flows using a 36-month rolling window to obtain monthly flow volatility, ensuring consistency with the time horizon used to obtain t-statistics.[7] In addition, I compute the fund age in months as the difference between the month-end date and the fund's inception date. Moreover, I calculate the manager's tenure in months as the difference between the month-end date and the date on which the respective manager assumed responsibility for the portfolio according to CRSP.

Finally, I retrieve information on expense and turnover ratios from CRSP. Unlike the other characteristics, these two variables are reported at an annual frequency and therefore do not vary across months within a given year. Based on the expense ratio, I compute value added following the definition of DeMiguel et al. (2023):

$$va_{i,t} = (\alpha_{i,t} + ER_{i,t}/12) \times (TNA_{i,t-1}). \tag{III.2}$$

This measure captures the monthly value added by the fund manager, expressed in dollars and based on abnormal performance, gross of fees.

2.3. Data preprocessing and prediction framework

Before applying ML algorithms to predict future mutual fund performance, several preprocessing steps are required. First, I define the target variable, i.e., future performance, as the one-month-ahead realized alpha ($\alpha^r_{i,t+1}$) obtained from the FF6 asset pricing model. This introduces a lag structure in which fund characteristics in

[6] Throughout this paper, this specification is referred to as FF6.

[7] DeMiguel et al. (2023) use the standard deviation of flows within a given year, which is consistent with the subsequent transformation to annual data.

month t are used as predictors, ensuring that only information available up to month t is utilized and thereby preventing any forward-looking bias. Second, I construct two samples, one for training the algorithms and one for prediction. As no imputation is performed for the target variable and most ML algorithms cannot handle missing data, this step is necessary in order to avoid look-ahead bias (Fausch et al., 2025a). More specifically, for the training sample, all observations for which the one-month-ahead alpha is not available are removed, resulting in a reduced sample of 11,414 unique share classes (1,049,376 observations).[8] The summary statistics for the training sample are reported in Table III.1.

[Insert Table III.1 here]

As I evaluate different design choices concerning ML algorithms, imputation methods, training data windows, frequency adjustments and hyperparameter tuning (see Section 3), the prediction framework adapts accordingly. Nevertheless, across all configurations, the out-of-sample period consistently starts in January 1994 and ends in December 2024.[9] After training the model according to the respective design choices, fund characteristics in month t are used to predict fund performance in $t+1$, yielding a predicted alpha distributions for all share classes in the prediction set at time t. In the case of the most frequent training data update (monthly; see Subsection 3.4), only characteristics up to month $t-1$ are included in the training set, thereby ensuring a strict separation between the training and prediction sample. Across all specifications and time-series estimation iterations, I standardize the characteristics following DeMiguel et al. (2023) and Fausch et al. (2025a). Specifically, each characteristic is scaled within the training sample (panel) to have zero mean and unit variance, and the estimated parameters are then applied to the prediction set.

2.4. Design choices, performance implications and non-standard errors

To assess the performance implication of individual design choices, I use a portfolio approach similar to Kaniel et al. (2023), DeMiguel et al. (2023) and Fausch et al. (2025a). More specifically, for each of the 2,052 paths[10] and in each month,

[8] These observations are retained in the prediction sample, as the target variable is not required for out-of-sample forecasting.

[9] Since CRSP does not report turnover ratios for 1992 and one design choice involves a non-imputation strategy, no prediction can be generated for the year 1993.

[10] All possible paths are explained in detail in Section 3.

I form an equally weighted top decile portfolio based on the predicted alpha distribution. The portfolio is rebalanced monthly using updated predictions, and its net return is tracked over the out-of-sample period (January 1994 to December 2024). Using the resulting 372 monthly return observations, I estimate the average alpha ($\hat{\alpha}_i$) based on the FF6 model for each specification:

$$\begin{aligned} R^e_{i,t} = \alpha_i + b_i(R_{M,t} - R_{f,t}) + s_i SMB_t + h_i HML_t \\ + r_i RMW_t + c_i CMA_t + m_i MoM_t + \varepsilon_{i,t}, \end{aligned} \tag{III.3}$$

where $R^e_{i,t}$ denotes the excess return of share class i in month t, $R_{f,t}$ is the risk-free rate, and $R_{M,t}$ is the market return. The factors SMB_t and HML_t capture the size and value–growth effects, respectively. The profitability and investment factors are represented by RMW_t and CMA_t. The momentum factor is given by MoM_t. The intercept α_i measures the abnormal return not explained by the asset pricing model and $\varepsilon_{i,t}$ denotes the regression error term. The 2,052 estimated alphas ($\hat{\alpha}_i$) are subsequently used to examine both the overall performance distribution and the distributions corresponding to individual design choices.

To measure and quantify the non-standard error (NSE) in this study, I follow the approach suggested by Walter, Weber, and Weiss (2024):

$$\text{NSE} = \sqrt{\frac{1}{I-1}\sum_{i=1}^{I}(\hat{\alpha}_i - \bar{\hat{\alpha}})^2}, \tag{III.4}$$

where $\hat{\alpha}_i$ represents the estimated alpha for path i based on Equation III.3. Thus, the NSE corresponds to the cross-sectional standard deviation of estimated alphas across $I = 2,052$ specifications and captures the sensitivity of results to alternative research design choices.

To relate the magnitude of the NSE to that of the standard error (SE), I estimate HAC-robust standard errors[11] for $\hat{\alpha}_i$ (based on Equation III.3) and compute the average standard error ($\overline{SE}$) across all possible paths:

$$\overline{\text{SE}} = \frac{1}{I}\sum_{i=1}^{I}\text{SE}(\hat{\alpha}_i). \tag{III.5}$$

Finally, I compute the ratio of NSE-to-SE, which quantifies the extent to which methodological uncertainty (NSE) contributes relative to sampling uncertainty (SE).

[11] The number of lags for the HAC estimator (Newey and West, 1987) is set to 12.

3. Machine learning design choices

The increasing use of ML methods for forecasting financial variables such as equity risk premiums and mutual fund performance has led to a growing range of design choices. Implementing these models requires researchers to make numerous methodological decisions, which can materially affect empirical results (Chen, Hanauer, and Kalsbach, 2024).

Building on this insight and following Chen, Hanauer, and Kalsbach (2024), I analyze decisions related to the choice of algorithm and the training window. In addition, I extend their framework by incorporating various imputation methods, training frequency and three choices with respect to hyperparameter tuning (search strategies, tuning procedures and frequency adjustment). However, I do not consider alternative target variables[12], the choice of transforming the target variable[13], feature selection prior to training[14], different subsamples[15] as well as post-publication treatment[16].

The set of categories and their corresponding design choices is illustrated in Figure III.1, with Panel (a) corresponding to OLS, Panel (b) to elastic net and Panel (c) to gradient boosting and random forest. Given the heterogeneity of the algorithms used in this paper, not all design choices apply to every method.[17] In total, I evaluate 2,052 distinct specifications to predict mutual fund alpha, comprising 36 for OLS, 648 for elastic net, and 684 for gradient boosting and random forest each. The main categories and their corresponding choices are described in detail in Subsection 3.1 through Subsection 3.7.

[Insert Figure III.1 here]

[12]The analysis focuses exclusively on predicting FF6 alpha. Alternative targets, such as raw returns or excess returns are not considered, as model performance is difficult to compare across fundamentally different prediction objectives in a portfolio-based evaluation framework.

[13]Chen, Hanauer, and Kalsbach (2024) consider both the raw target and a binary transformation. As my analysis relies on sorting share classes based on predicted alphas, a binary target would remove the cross-sectional variation required for portfolio formation.

[14]Chen, Hanauer, and Kalsbach (2024) use LASSO for feature selection prior to estimation. However, sparse methods such as LASSO can be unstable and may lead to arbitrary feature selection (Xu, Caramanis, and Mannor, 2011).

[15]I follow DeMiguel et al. (2023) and Kaniel et al. (2023) and rely on the sample of U.S. actively managed funds based on CRSP.

[16]This is in line with Kaniel et al. (2023) and DeMiguel et al. (2023).

[17]For example, OLS does not involve hyperparameters, and thus no decisions arise in those categories.

3.1. Algorithm (N=4)

As the literature on financial machine learning continues to grow, the range of algorithms employed has expanded accordingly. Nevertheless, the majority of studies still use OLS as the baseline model against which more advanced methods are benchmarked. Besides OLS, the most commonly used algorithms are elastic net (EN), random forest (RF), gradient boosting (GB), sometimes also referred as boosted regression trees, as well as feed-forward neural networks.[18]

In this study, I closely follow DeMiguel et al. (2023) and Fausch et al. (2025a) and employ a range of forecasting models, including OLS, elastic net, gradient boosting, and random forest, to predict mutual fund alpha.[19] OLS and elastic net are linear models that differ in their regularization properties, whereas gradient boosting and random forest are flexible machine learning methods capable of capturing nonlinearities and complex interactions among predictors without requiring prior specification.[20]

3.2. Imputation method (N=3)

Missing values are common in financial datasets, including mutual fund data from CRSP (see, e.g., DeMiguel et al., 2023; Fausch et al., 2025a,b). When using such data to predict fund performance, researchers must therefore decide how to handle missing observations, either by excluding them or by applying an imputation method. Accordingly, the literature on financial machine learning has adopted a variety of approaches to address this issue. For instance, Li and Rossi (2021) rely exclusively on boosted regression trees, which can inherently handle missing values. In contrast, DeMiguel et al. (2023) use the mean, while Gu, Kelly, and Xiu (2020) and Bali et al. (2022) use the median. Kaniel et al. (2023) rely on a latent factor model for imputation, whereas Fausch et al. (2025b) exclude observations with missing values altogether to avoid dependence on a specific imputation technique.

The importance of this methodological choice is highlighted by Filippou et al. (2025), who show that the selection of a particular imputation strategy can sub-

[18] See, e.g., Gu, Kelly, and Xiu (2020), Bianchi, Büchner, and Tamoni (2021), Leippold, Wang, and Zhou (2022), Bali et al. (2022), Bali et al. (2023), DeMiguel et al. (2023), Kaniel et al. (2023), Fausch et al. (2025a) and Fausch et al. (2025b).

[19] Consistent with DeMiguel et al. (2023) and Fausch et al. (2025a), I do not rely on feed-forward neural networks, which are used in Kaniel et al. (2023). Their implementation typically requires substantially higher computational resources, particularly in the presence of multiple hyperparameter tuning dimensions. Given the large number of model specifications considered, incorporating neural networks would render the analysis computationally infeasible. All models are implemented in *Python* using the *scikit-learn* package.

[20] A detailed description of these models can be found in Gu, Kelly, and Xiu (2020).

stantially affect the accuracy of hedge fund return predictions. Although Chen, Hanauer, and Kalsbach (2024) acknowledge multiple imputation techniques, they do not treat this as a distinct design choice. Instead, they default to cross-sectional mean imputation in their application. In contrast, I explicitly examine three simple imputation strategies and their implications for mutual fund performance prediction: (i) excluding all observations with missing values in any of the predictors, (ii) mean imputation and (iii) median imputation.

3.3. Training window (N=4)

To separate signal from noise in machine learning predictions, it is often argued that larger training samples improve forecasting performance. While this may hold in purely cross-sectional settings, its validity is less clear in time-series applications. In financial machine learning, a common approach is the expanding window, where new observations are continuously added while all past data is retained.[21] However, this design choice presents a fundamental trade-off. Older observations may be less informative for predicting current mutual fund performance, so that including them can introduce noise rather than improve predictive accuracy. In addition, expanding windows substantially increase computational costs, as both model estimation and hyperparameter tuning become more demanding over time. In contrast, rolling windows limit the training sample to more recent observations. This results in smaller and computationally more tractable datasets that may better capture current market conditions by adapting to structural changes in financial markets. In this context, Bali et al. (2023) and Fausch et al. (2025b) consider both expanding and rolling windows. They generally find slightly higher predictive accuracy for expanding windows, though the differences are small and model-dependent.

This design choice is also acknowledged by Chen, Hanauer, and Kalsbach (2024), who apply either an expanding window or a fixed 10-year rolling window, but do not explicitly examine window length as a separate modeling decision. Building on this, I explicitly study the effects of different training horizons to allow for a systematic comparison between longer windows, which incorporate more historical information, and shorter windows emphasizing more recent market dynamics. Additionally, the length of the window may affect the sensitivity of the estimates and thus the non-standard error. To capture this dimension, I follow Fausch et al. (2025b) and implement an expanding window as well as rolling windows with training lengths

[21] The expanding window approach is used by, among others, Gu, Kelly, and Xiu (2020); Bianchi, Büchner, and Tamoni (2021); Leippold, Wang, and Zhou (2022); Bali et al. (2023); DeMiguel et al. (2023); Fausch et al. (2025b).

of 12, 36 and 60 months.

3.4. Training frequency (N=3)

When working with time-series data, researchers must decide how often to retrain models. Although more frequent updates allow models to incorporate the latest information, they also substantially increase computational costs. This creates a trade-off between model responsiveness and computational feasibility.

In the financial machine learning literature, both frequent and infrequent retraining schemes are employed. Some studies align retraining with the data frequency, updating models monthly or annually, depending on the application (DeMiguel et al., 2023; Fausch et al., 2025a,b). In contrast, other studies update models less frequently due to computational constraints, despite producing monthly forecasts (Gu, Kelly, and Xiu, 2020; Leippold, Wang, and Zhou, 2022). While Chen, Hanauer, and Kalsbach (2024) fix the retraining frequency at an annual level, I explicitly treat it as a design choice and analyze its impact on predictive performance. Accordingly, all models are retrained at monthly, semiannual or annual frequency.

3.5. Hyperparameter search strategy (N=4)

To improve out-of-sample prediction accuracy and mitigate overfitting, researchers must choose an appropriate hyperparameter search strategy. Accordingly, the literature on financial machine learning has adopted a variety of approaches to address this issue. For instance, Li and Rossi (2021) rely primarily on default hyperparameters provided by the *LightGBM* package. In contrast, grid search is the most commonly used method for identifying optimal configurations (see, e.g., Gu, Kelly, and Xiu, 2020; Leippold, Wang, and Zhou, 2022; DeMiguel et al., 2023; Fausch et al., 2025a).[22] To overcome the computational limitations and narrow grid restrictions of this approach, Bali et al. (2023) use random search[23] (Bergstra and Bengio, 2012), whereas Goulet Coulombe et al. (2025) and Fausch et al. (2025b) rely on the Tree-structured Parzen Estimator (TPE) approach based on Bayesian optimization[24] (Bergstra et al., 2011).

[22]This approach evaluates all possible combinations of predefined hyperparameter values to identify the optimal configuration.

[23]Instead of exhaustively exploring all possible combinations (grid search), this approach specifies the number of trials in advance and samples hyperparameter configurations randomly from predefined distributions. This strategy enables a broader exploration of the hyperparameter space while holding computational cost fixed.

[24]Bayesian optimization further improves efficiency by adaptively focusing on promising regions of the search space.

Despite the variety of approaches, Chen, Hanauer, and Kalsbach (2024) do not consider these design choices in their study. In contrast, I evaluate the four strategies (default settings, grid search, random search and Bayesian optimization) and assess their impact on predictive performance. However, as OLS does not require hyperparameter tuning, this category is not applicable. For elastic net, default settings yield identical predictions within a given month and are therefore excluded.[25] Table III.2 reports the hyperparameters and their corresponding search spaces for each algorithm. To ensure comparability, the number of iterations in random and Bayesian search is set equal to the grid size, following Bergstra and Bengio (2012). Hyperparameters are selected based on validation or cross-validated root mean squared error (see Subsection 3.6).

[Insert Table III.2 here]

3.6. Hyperparameter tuning procedure (N=2)

In addition to the choice of search strategy, hyperparameter tuning requires selecting a procedure for evaluating candidate configurations. In the financial machine learning literature, a common approach is to split the training sample into a training and validation set (see, e.g., Gu, Kelly, and Xiu, 2020; Bianchi, Büchner, and Tamoni, 2021; Leippold, Wang, and Zhou, 2022; Bali et al., 2022, 2023; Kaniel et al., 2023), thereby preserving the temporal structure of the data. However, Bergmeir, Hyndman, and Koo (2018) argue that maintaining this structure is not always necessary, even in the presence of serial correlation or non-stationarity, and show that k-fold cross-validation can improve forecasting performance. Consistent with this evidence, DeMiguel et al. (2023) document that cross-validation improves predictive performance in mutual fund prediction, and related studies adopt this approach in similar settings (Fausch et al., 2025a,b).

Chen, Hanauer, and Kalsbach (2024) do not consider this design choice and instead rely on a fixed validation split. In contrast, I examine its impact on predictive performance by comparing the two approaches. Specifically, I implement (i) a validation split, in which 75% of the training sample is used for model estimation and 25% for validation, following Kaniel et al. (2023), and (ii) a 5-fold cross-validation procedure that disregards the chronological ordering of the data (DeMiguel et al., 2023; Fausch et al., 2025a,b).

[25] This occurs because the default shrinkage parameter in *scikit-learn* is set to 1, shrinking all coefficients effectively to zero and leaving only the intercept.

3.7. Hyperparameter frequency adjustment (N=3)

After specifying the search strategy and tuning procedure, an additional design choice in time-series settings concerns the frequency of hyperparameters updates. When generating monthly predictions over an extended out-of-sample period, researchers must decide whether to re-tune hyperparameters at each iteration or only at lower frequencies (e.g., annually). While more frequent updates may improve predictive accuracy by incorporating the most recent information, they also substantially increase computational costs, as hyperparameter tuning is typically the most time-intensive part in the ML workflow. Accordingly, the financial machine learning literature often updates hyperparameters less frequently than predictions are generated. For example, although Gu, Kelly, and Xiu (2020), Leippold, Wang, and Zhou (2022), Bali et al. (2023) and Fausch et al. (2025b) generate monthly forecasts, hyperparameters are updated only annually (Gu, Kelly, and Xiu, 2020; Leippold, Wang, and Zhou, 2022; Bali et al., 2023) or even every other year (Fausch et al., 2025b). This variation highlights that the frequency of hyperparameter tuning is a modeling choice without a clear consensus in the literature, with both annual and multi-year update schemes being commonly used.

Chen, Hanauer, and Kalsbach (2024) do not consider this dimension as a separate design choice and adopt the approach of updating hyperparameters annually. In contrast, I examine the impact of alternative update intervals by comparing annual tuning with longer update frequencies of three and five years. More frequent updates are not considered due to their prohibitive computational cost.

4. Empirical analysis

In the empirical analysis, I investigate how different ML design choices impact the out-of-sample alpha of mutual fund portfolios by analyzing performance across a large set of model specifications. In addition, I assess the extent to which these choices contribute to estimation uncertainty by comparing non-standard errors to conventional sampling variation.

4.1. Design choice impact on portfolio alpha

To assess how ML design choices affect out-of-sample portfolio alpha, I begin by examining the best and worst performing configurations for each algorithm (see Table III.3). Overall, there is little evidence of robust positive long-only alpha across specifications.

Among the best configurations, all algorithms deliver positive alpha estimates. However, only elastic net achieves statistical significance, with an alpha of 10.8 bps per month (1.30% annually), and even this result is only marginally significant at the 10% level. Although gradient boosting and random forest generate larger alphas of 11.6 bps and 14.7 bps (1.39% and 1.76% annually), respectively, these estimates are not statistically different from zero, while OLS produces the lowest alpha at 4.7 bps (0.56% annually). Across all 2,052 models, only two portfolios (both based on elastic net) deliver statistically significant positive alpha. Given the large number of specifications tested, such limited evidence is likely attributable to data snooping rather than genuine predictability. This interpretation is consistent with the results of Fausch et al. (2025a), who find no evidence of significant long-only alpha in equity mutual funds after correcting for the forward-looking bias in DeMiguel et al. (2023). In line with prior empirical evidence (see, e.g., Jensen, 1968; Malkiel, 1995; Gruber, 1996; Fama and French, 2010), equally weighted and asset-weighted portfolios across all share classes exhibit statistically significant negative alphas of -12.1 bps and -9.2 bps (-1.45% and -1.10% annually), respectively.

In contrast, negative alpha is both more frequent and more pronounced in the worst configurations. With the exception of gradient boosting, all algorithms produce statistically significant negative alphas at the 10% level or higher. The magnitude of underperformance is particularly large for linear models, with elastic net reaching -20.9 bps (-2.51% annually, significant at the 1% level) and OLS -16.1 bps (-1.93% annually, significant at the 5% level). By comparison, the nonlinear methods perform somewhat better, with gradient boosting at -11.8 bps (-1.42% annually) and random forest at -13.1 bps (-1.57% annually, significant at the 10% level), both broadly comparable to the equally weighted benchmark. Overall, 153 out of 2,052 specifications generate statistically significant negative alpha. This corresponds to 11.11% of OLS models (4 out of 36), 22.84% of elastic net models (148 out of 648) and only a single specification for random forest. This pattern suggests that performance is highly sensitive to design choices, particularly for linear models, which appear more prone to possible overfitting or unstable out-of-sample performance under certain specifications.

[Insert Table III.3 here]

Although the previously discussed extreme configurations illustrate the potential performance range, they provide only limited insight into the role of individual design choices. To address this, I examine the distribution of annualized alphas across the seven design choice categories over the out-of-sample period from January 1994

to December 2024. Figure III.2 presents the corresponding boxplots, where the horizontal line within each box denotes the median and the dot the mean. The median alpha serves as the main reference point for comparing performance across specifications.[26] The figure reveals substantial within-category heterogeneity, particularly for algorithm choice, training window, training frequency and hyperparameter search strategy.

Across algorithms, nonlinear methods seem to outperform linear models. Random forest performs best with an alpha of 0.55%, followed by gradient boosting (0.45%). In contrast, OLS and elastic net both exhibit negative alphas (-0.19% and -0.43%, respectively). Consistent with Table III.3, elastic net also shows the largest dispersion, spanning 3.8 percentage points (from 1.30% to -2.51%), indicating high sensitivity to other design choices.

Turning to the training window reveals that performance improves with longer training histories. The expanding window delivers the highest alpha (0.81%), followed by rolling windows of 36 and 60 months (0.41% and 0.36%, respectively) while the 12-month specification performs worst with a negative alpha of -0.64%. Overall, this suggests that, in the context of mutual fund performance prediction, access to a longer data history is more valuable than focusing on short-term fluctuations, indicating that even older observations retain predictive content rather than introducing noise. In contrast to the training window results, for the training frequency, the most computationally intensive approach, monthly retraining, yields the weakest performance with an alpha of 0.06%. The strongest results are obtained with semiannual updates (0.51%), followed by annual retraining scheme (0.37%).

Finally, hyperparameter tuning improves performance relative to default settings. While default parameters yield slightly negative results (-0.05%), all tuning approaches generate positive alphas, ranging from 0.27% (grid search) to 0.39% (random search), with Bayesian optimization yielding 0.34%. Overall, differences across tuning methods are relatively small, suggesting that defining a sensible hyperparameter space is more important than employing increasingly sophisticated optimization techniques.

The remaining design choices, i.e., imputation method, hyperparameter tuning procedure and hyperparameter adjustment frequency, exhibit comparatively low within-category variation. However, contrary to expectations, more computationally intensive approaches, such as cross-validation or frequent hyperparameter updates, do not improve performance compared to simpler alternatives.

[26] Across all categories, means are generally slightly below medians, suggesting mild left-skewness in the distribution of alphas.

[Insert Figure III.2 here]

Using the distribution of alphas across the 2,052 top-decile portfolios, the previous analysis showed that design choices have a substantial impact on performance. The largest variation, however, arises from the choice of algorithm, highlighting it as the most important decision dimension. To further explore this heterogeneity, I examine differences within categories for each algorithm. Figure III.3 extends Figure III.2 by presenting boxplots of design choices separately for each algorithm and reveals several additional insights. For OLS (panel a), the general pattern that more training data improves performance does not hold. The best result is obtained with a 36-month rolling window (0.36%), which outperforms the expanding window approach (0.24%). For elastic net (panel b), cross-validation yields substantially higher alpha than the commonly used validation approach (-0.21% versus -0.75%) and also appears more robust, as indicated by a lower range (3.06% versus 3.80%). For gradient boosting (panel c), the expanding window performs particularly well, delivering a median alpha close to 1% and a minimum alpha of 0.38%, while also exhibiting the lowest dispersion among all configurations (range of 1.01%), implying robustness to alternative design choice categories. For random forest (panel d), hyperparameter tuning appears to be crucial: while the default specification yields a median alpha of -0.34%, alternative search strategies achieve substantially higher values, ranging from 0.46% to 0.54%.[27] Overall, the results indicate that there is no universally optimal set of design choices across algorithms, highlighting the importance of careful, method-specific design for achieving strong out-of-sample portfolio performance.

[Insert Figure III.3 here]

4.2. Non-standard errors

Building on the analysis of how design choices affect out-of-sample portfolio alpha, this section examines the extent to which these choices contribute to estimation uncertainty. In particular, I compare non-standard errors, which capture variation induced by researcher choices, with standard errors, reflecting pure sampling variation.

As outlined in Subsection 2.4, I estimate the overall NSE as the cross-sectional standard deviation of the 2,052 portfolio alphas, capturing sensitivity to plausible

[27] This may partly reflect the implementation of the random forest algorithm in the *scikit-learn* package, where the default setting considers all features at each split, potentially increasing tree correlation and reducing out-of-sample performance relative to more conventional random forest specifications.

research design choices and relate this metric to the SE. Across all specifications, the overall NSE amounts to an economically sizeable annualized alpha of 0.83%. However, the NSE-to-SE ratio amounts to 0.86, indicating that the sampling error still exceeds research design-induced variation in this study. This ratio is lower than those reported in Chen, Hanauer, and Kalsbach (2024), Walter, Weber, and Weiss (2024) and Soebhag, Van Vliet, and Verwijmeren (2024), with values of 1.58, 1.10 and 1.06, respectively, but lies within the range documented by Menkveld et al. (2024) (0.59 to 1.99). One explanation for the lower ratio is the use of HAC-robust standard errors rather than conventional standard errors as in e.g., Walter, Weber, and Weiss (2024), which increases the average SE and thereby reduces the ratio.[28]

Next, to assess whether certain design choices are particularly sensitive to alternative specifications, I compute conditional non-standard errors (NSE), standard errors (SE) and their ratio for each design choice $c \in \mathcal{C}$, where $\mathcal{C}$ denotes the full set of the 23 design choices (i.e., OLS, elastic net, . . ., 60-month hyperparameter adjustment). Table III.4 reports the results of this conditional analysis. The NSE (column 2) is highest for the validation-based tuning procedure (0.92), elastic net (0.90), no data imputation (0.87) and monthly retraining frequency (0.87), indicating that these choices are most sensitive to alternative specifications. In contrast, the expanding training window (0.44), gradient boosting (0.60), random forest (0.62) and default hyperparameters (0.62) exhibit the lowest NSE values and therefore appear most robust.

As standard errors are broadly comparable across choices, these patterns carry over to the NSE-to-SE ratios (column 4 of Table III.4). Within the *Algorithm* category, elastic net exhibits the highest ratio (1.12), making it the only specification in this study for which NSE exceeds SE, indicating that design-induced uncertainty outweighs sampling variation. The second linear model, OLS, also shows a relatively high ratio (0.90). Notably, both algorithms are among the most commonly used in the financial machine learning literature.[29] By contrast, the nonlinear methods (random forest and gradient boosting) exhibit comparatively low ratios (both 0.59), which are among the lowest across all specifications. While for *Imputation*, differences are small, with ratios ranging from 0.84 to 0.89, the opposite is true for *Training window*. The expanding window yields the lowest ratio (0.47), both within this category and overall, indicating that it adds relatively little uncertainty

[28] Using conventional standard errors increases the ratio to 1.06, aligning it with the estimates in Soebhag, Van Vliet, and Verwijmeren (2024) and Walter, Weber, and Weiss (2024).

[29] For instance, Gu, Kelly, and Xiu (2020), Bianchi, Büchner, and Tamoni (2021), Leippold, Wang, and Zhou (2022), DeMiguel et al. (2023), and Fausch et al. (2025a,b) employ either elastic net, OLS, or both.

beyond sampling variation. By comparison, rolling windows (12, 36, and 60 months) produce higher ratios between 0.70 and 0.73. For training frequency, monthly retraining results in the highest ratio (0.92), whereas semiannual and annual updates yield lower values (0.79).

Considerable within-category variation is also observed for design choices related to hyperparameter tuning. For the *Hyperparameter search strategy*, using default parameters introduces relatively little additional uncertainty (0.63). However, despite this, most studies in the financial machine learning literature employ some sort of hyperparameter tuning.[30] Across the three tuning approaches considered, differences are small, with ratios ranging from 0.86 to 0.88. For the *Hyperparameter tuning procedure*, the commonly used validation set yields a considerable higher ratio (0.97) than 5-fold cross-validation (0.75), suggesting that it introduces nearly as much uncertainty from design choices as from the sampling variation.[31] Finally, ratios for *Hyperparameter frequency adjustment* are broadly similar, ranging from 0.82 to 0.89.

[Insert Table III.4 here]

Overall, elastic net, validation-based hyperparameter tuning and monthly retraining are associated with the highest levels of design-induced uncertainty relative to sampling variation. By contrast, the expanding training window, gradient boosting, random forest and default hyperparameters appear comparatively robust.

5. Conclusion

This paper examines how machine learning design choices affect the predictability of equity mutual fund performance and quantifies the extent to which these choices contribute to researcher-induced uncertainty. Using 2,052 model specifications across seven design choice categories, I provide a systematic evaluation of both performance implications and non-standard errors.

The results show that identifying outperforming funds ex ante remains a challenging task. Across all specifications, only two models generate statistically significant positive alpha, which is likely attributable to data snooping rather than genuine predictability. Overall, the findings are consistent with recent evidence

[30] Apart from Li and Rossi (2021), nearly all studies rely on a tuning strategy.

[31] The studies of Gu, Kelly, and Xiu (2020), Bianchi, Büchner, and Tamoni (2021), Leippold, Wang, and Zhou (2022), Bali et al. (2022, 2023) and Kaniel et al. (2023) all rely on a validation set for tuning hyperparameters.

suggesting that machine learning is more effective at identifying underperforming funds than consistently selecting outperforming funds.

At the same time, design choices have economically meaningful effects on ML-based portfolio outcomes. The choice of algorithm emerges as particularly important, with nonlinear methods generally outperforming linear models. In addition, expanding training windows tend to deliver stronger and more stable results, indicating that long-term patterns are informative for fund performance prediction. However, these effects are not uniform across models, highlighting the importance of model-specific design choices.

Beyond performance, the results highlight the role of researcher-induced uncertainty. Non-standard errors are economically sizeable and, with an NSE-to-SE ratio of 0.86, are of similar magnitude to sampling variation. This uncertainty varies substantially across specifications and linear models, validation-based tuning, and frequent retraining are particularly sensitive to design choices, whereas nonlinear methods and expanding windows appear more robust.

Overall, the findings suggest that machine learning does not fundamentally alter the difficulty of identifying equity mutual funds that generate positive alpha. More importantly, they demonstrate that empirical results are highly sensitive to methodological choices, underscoring the need for transparency and systematic robustness analysis in financial machine learning research.

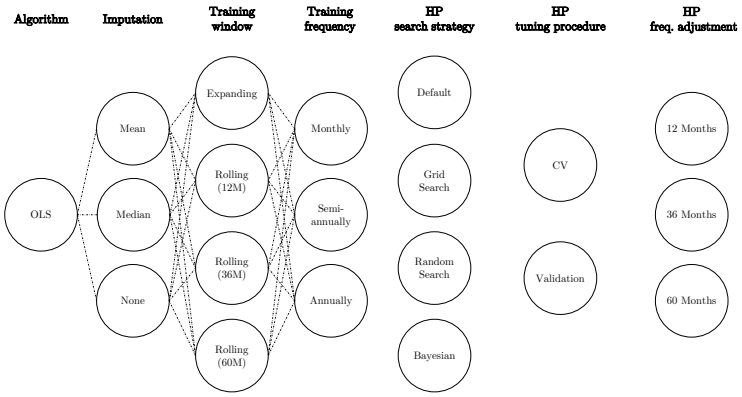
Table III.1: **Share class characteristics: Descriptive statistics**

This table reports monthly descriptive statistics (mean, median, standard deviation, and number of class-month observations) of all share class characteristics considered in the training sample. Expense ratio, turnover ratio and manager tenure are reported based on annual data. All variables are measured at the share class level and correspond to actively managed US domestic equity share classes from January 1980 to December 2024.

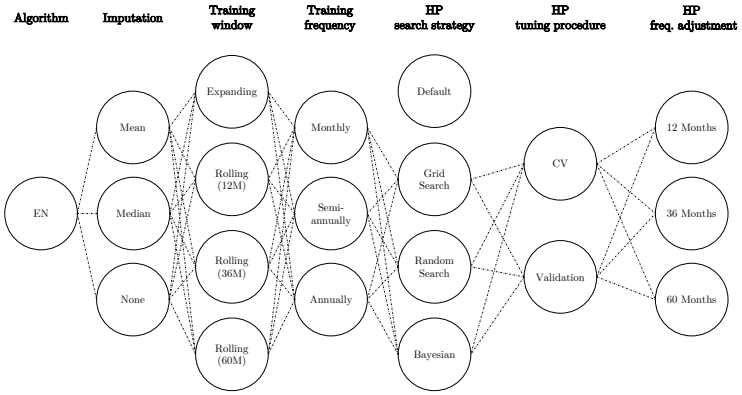
	Mean	Median	Standard deviation	Class-month observations
monthly return	0.912%	1.305%	5.268%	1,095,717
monthly realized alpha	-0.166%	-0.156%	2.078%	1,058,232
alpha t-stat	-0.523	-0.539	1.174	1,059,620
TNA (USD mill.)	826.567	131.300	3351.314	1,085,155
expense ratio	1.015%	0.970%	0.968%	844,905
age (months)	168.868	144.000	113.675	1,095,894
flows	-0.002	-0.006	0.067	1,084,709
manager tenure (years)	103.602	88.000	70.399	645,773
turnover ratio	0.680	0.470	0.928	845,084
volatility of flows	0.053	0.034	0.079	1,043,670
value added	-0.430	-0.024	49.664	840,724
market beta t-stat	18.768	17.218	10.610	1,059,620
profitability beta t-stat	-0.058	-0.055	1.502	1,059,620
investment beta t-stat	-0.455	-0.486	1.655	1,059,620
size beta t-stat	1.430	0.624	3.651	1,059,620
value beta t-stat	0.325	0.099	2.518	1,059,620
momentum beta t-stat	-0.102	-0.079	1.743	1,059,620
R^2	0.908	0.943	0.122	1,059,620

Figure III.1: **Design choices**

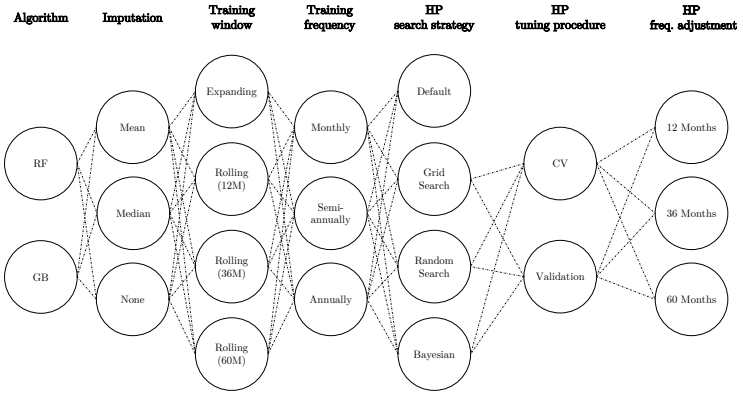
This figure plots the seven categories and their respective options applied in the empirical analysis to predict mutual fund alpha.



(a) Design choices for OLS



(b) Design choices for elastic net



(c) Design choices for gradient boosting and random forest

Table III.2: **Hyperparameter search spaces**

This table reports the hyperparameter search spaces for elastic net, random forest and gradient boosting. For grid search, each hyperparameter is tuned over a predefined set of discrete grid values, with the total grid size reported in column four. For random search and Bayesian optimization, hyperparameters are sampled from continuous or integer-valued search spaces over a specified number of iterations, also reported in column four.

	Grid search		
Model	Hyperparameter	Grid values	Grid size
Elastic net	Shrinkage strength	$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0.5\}$	150
	penalty mix (L_1, L_2)	$\{0.01, 0.25, 0.5, 0.75, 0.99\}$	
	Maximum number of iterations	$\{50, 100, 200, 500, 1000, 2000\}$	
Random Forest	Number of trees	$\{100, 500, 1000\}$	144
	Maximum depth of tree	$\{2, 10, 20, 30\}$	
	Subsample fraction	$\{0.6, 0.8, 1\}$	
	Number features per split	$\{\sqrt{17}, \log_2(17)\}$	
	Minimum samples per leaf	$\{2, 1000\}$	
Gradient boosting	Number of trees	$\{50, 500, 2000, 4000\}$	144
	Maximum depth of tree	$\{2, 5, 10, 20\}$	
	Learning rate	$\{10^{-3}, 10^{-2}, 10^{-1}\}$	
	Minimum samples per leaf	$\{1, 5, 20\}$	
	Random search and Bayesian optimization		
Model	Hyperparameter	Search space	Iterations
Elastic net	Shrinkage strength	$[10^{-4}, 0.5]$	150
	penalty mix (L_1, L_2)	$[0.01, 0.99]$	
	Maximum number of iterations	integer: $[50, 2000]$	
Random Forest	Number of trees	integer: $[100, 1000]$	144
	Maximum depth of tree	integer: $[2, 30]$	
	Subsample fraction	$[0.6, 1]$	
	Number features per split	integer: $[1, 17]$	
	Minimum samples per leaf	integer: $[2, 1000]$	
Gradient boosting	Number of trees	integer: $[50, 4000]$	144
	Maximum depth of tree	integer: $[2, 20]$	
	Learning rate	$[10^{-3}, 10^{-1}]$	
	Minimum samples per leaf	integer: $[1, 20]$	

Table III.3: **Monthly out-of-sample alpha (in %) of best and worst configuration per algorithm**

Column 1 of this table reports the monthly out-of-sample alphas (in %) net of all costs of the equally weighted top decile portfolios for the best configuration per algorithm and column 2 for the worst configuration. The out-of-sample alphas are obtained by regressing excess monthly portfolio returns on the Fama and French (2015) 5-factor model extended by the momentum factor proposed by Carhart (1997) (FF6). The out-of-sample period spans from January 1994 to December 2024 (372 observations). HAC-robust standard errors, as suggested by Newey and West (1987), are reported in parentheses. Following DeMiguel et al. (2023), Kaniel et al. (2023) and Fausch et al. (2025a), the number of lags is set to 12. The level of significance is displayed as *, ** and ***, indicating statistical significance at the 10%, 5% and 1% level, respectively.

	Best configuration	**Worst configuration**
OLS	0.047	-0.161**
	(0.083)	(0.070)
Elastic net	0.108*	-0.209***
	(0.062)	(0.065)
Gradient boosting	0.116	-0.118
	(0.081)	(0.079)
Random forest	0.147	-0.131*
	(0.093)	(0.079)
Equally weighted	-0.121***	-0.121***
	(0.034)	(0.034)
Asset weighted	-0.092***	-0.092***
	(0.028)	(0.028)

Figure III.2: **Distribution of annualized alphas across alternative design choices**

This figure plots annualized alphas grouped by algorithm, imputation method, training window length, training frequency, hyperparameter search strategy, validation approach and hyperparameter tuning frequency. The out-of-sample alphas are obtained by regressing excess monthly portfolio returns on the Fama and French (2015) 5-factor model extended by the momentum factor proposed by Carhart (1997) (FF6). The out-of-sample period spans from January 1994 to December 2024 (372 observations). Each box summarizes the cross-specification distribution of alpha. The horizontal line within each box denotes the median and the dot indicates the mean.

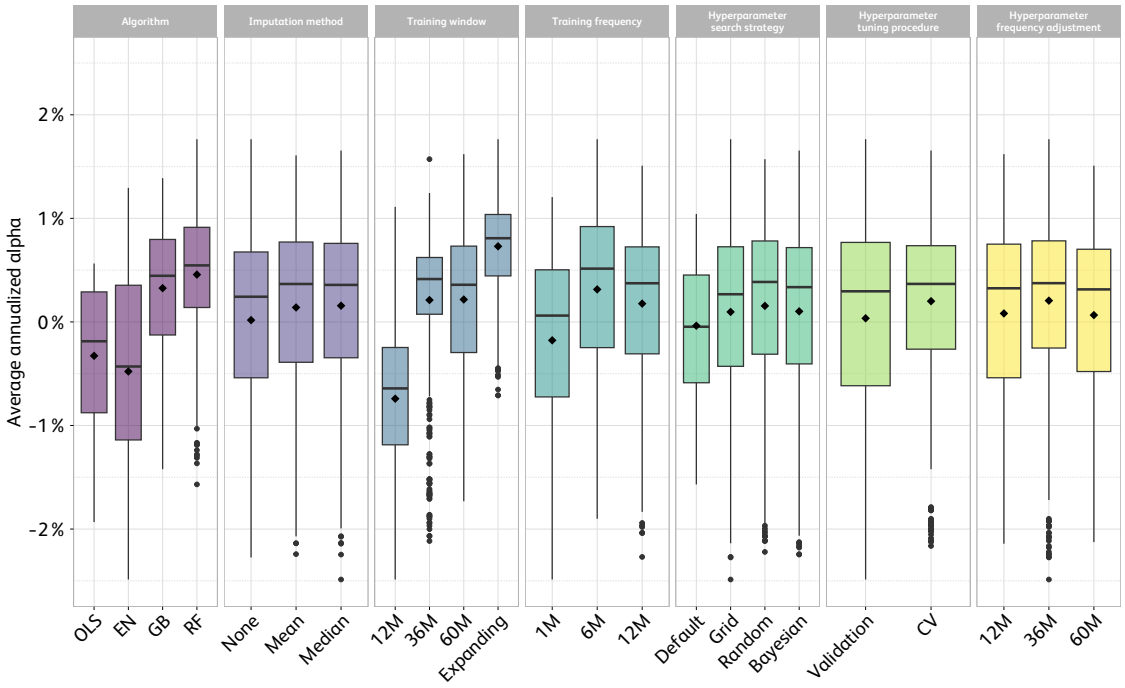
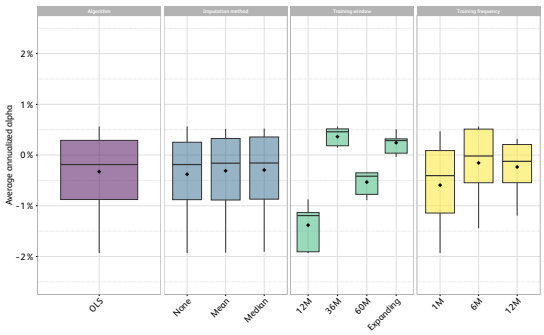
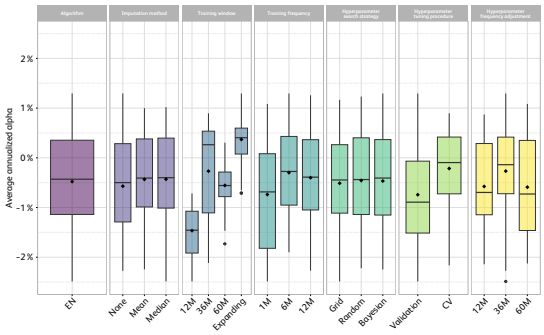


Figure III.3: **Distribution of alphas across alternative design choices by algorithm**

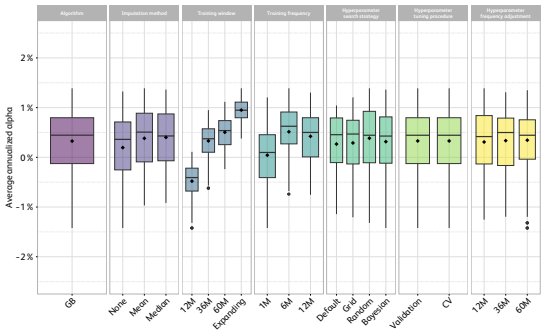
This figure plots annualized alphas across design choices, separately for each algorithm (OLS, elastic net, gradient boosting and random forest). The out-of-sample alphas are obtained by regressing excess monthly portfolio returns on the Fama and French (2015) 5-factor model extended by the momentum factor proposed by Carhart (1997) (FF6). The out-of-sample period spans from January 1994 to December 2024 (372 observations). Each box summarizes the cross-specification distribution of alpha. The horizontal line within each box denotes the median and the dot indicates the mean.



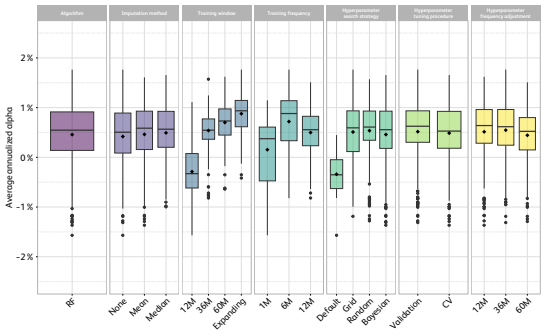
(a) Distribution for OLS



(b) Distribution for elastic net



(c) Distribution for gradient boosting



(d) Distribution for random forest

Table III.4: **NSE-to-SE ratios across research design choices**

This table reports non-standard errors (NSE) for each research design choice within a given category, along with HAC-robust standard errors (SE) (Newey and West, 1987) with 12 lags, the corresponding NSE-to-SE ratios and the number of observations. The NSE is defined as the cross-sectional standard deviation of portfolio alphas within each design choice (Walter, Weber, and Weiss, 2024). The out-of-sample period spans January 1994 to December 2024.

1. Algorithm	**NSE**	**SE**	**Ratio**	**Obs.**
OLS	0.76	0.84	0.90	36
Elastic net	0.90	0.80	1.12	648
Gradient boosting	0.60	1.02	0.59	684
Random forest	0.62	1.05	0.59	684
2. Imputation	**NSE**	**SE**	**Ratio**	**Obs.**
No imputation	0.87	0.98	0.89	684
Mean	0.81	0.95	0.85	684
Median	0.80	0.94	0.84	684
3. Training window	**NSE**	**SE**	**Ratio**	**Obs.**
Expanding	0.44	0.94	0.47	513
Rolling (12M)	0.69	0.94	0.73	513
Rolling (36M)	0.70	1.00	0.70	513
Rolling (60M)	0.66	0.95	0.70	513
4. Training frequency	**NSE**	**SE**	**Ratio**	**Obs.**
Monthly	0.87	0.95	0.92	684
Semiannually	0.78	0.99	0.79	684
Annually	0.74	0.93	0.79	684
5. HP search strategy	**NSE**	**SE**	**Ratio**	**Obs.**
Default	0.62	0.98	0.63	72
Grid search	0.83	0.96	0.86	648
Random search	0.84	0.95	0.88	648
Bayesian optimization	0.83	0.96	0.87	648
6. HP tuning procedure	**NSE**	**SE**	**Ratio**	**Obs.**
Validation	0.92	0.94	0.97	972
Cross-validation	0.73	0.98	0.75	972
7. HP freq. adjustment	**NSE**	**SE**	**Ratio**	**Obs.**
12 months	0.84	0.95	0.89	648
36 months	0.80	0.98	0.82	648
60 months	0.85	0.95	0.89	648

Bibliography

Akey, P., A. Z. Robertson, and M. Simutin. 2026. Noisy factors? the retroactive impact of methodological changes on the fama-french factors. *Review of Finance* rfag002.

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*, KDD '19, 2623–2631. Association for Computing Machinery. ISBN 9781450362016. doi:10.1145/3292500.3330701.

Amihud, Y., and R. Goyenko. 2013. Mutual fund's r^2 as predictor of performance. *Review of Financial Studies* 26:667–94.

Aragon, G. O., and M. S. Kim. 2023. Fire sale risk and expected stock returns. *Journal of Financial Economics* 149:578–609.

Atanasov, V., S. V. Møller, and R. Priestley. 2020. Consumption fluctuations and expected returns. *Journal of Finance* 75:1677–713.

Athey, S., and G. W. Imbens. 2019. Machine learning methods that economists should know about. *Annual Review of Economics* 11:685–725.

Baker, M., and J. Wurgler. 2006. Investor sentiment and the cross-section of stock returns. *Journal of Finance* 61:1645–80.

———. 2007. Investor sentiment in the stock market. *Journal of Economic Perspectives* 21:129–51.

Bali, T. G., H. Beckmeyer, M. Moerke, and F. Weigert. 2023. Option return predictability with machine learning and big data. *Review of Financial Studies* 36:3548–602.

Bali, T. G., A. Goyal, D. Huang, F. Jiang, and Q. Wen. 2022. Predicting corporate bond returns: Merton meets machine learning. *Unpublished Working Paper, Georgetown University, University of Lausanne, Singapore Management University, Central University of Finance and Economics* .

Barber, B. M., X. Huang, and T. Odean. 2016. Which factors matter to investors? evidence from mutual fund flows. *Review of Financial Studies* 29:2600–42.

Barras, L., O. Scaillet, and R. Wermers. 2010. False discoveries in mutual fund performance: Measuring luck in estimated alphas. *Journal of Finance* 65:179–216.

Bekaert, G., E. C. Engstrom, and N. R. Xu. 2022. The time variation in risk appetite and uncertainty. *Management Science* 68:3975–4004.

Ben-David, I., J. Li, A. Rossi, and Y. Song. 2022. What do mutual fund investors really care about? *Review of Financial Studies* 35:1723–74.

Ben-Rephael, A., S. Kandel, and A. Wohl. 2012. Measuring investor sentiment with mutual fund flows. *Journal of Financial Economics* 104:363–82.

Bergmeir, C., R. J. Hyndman, and B. Koo. 2018. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis* 120:70–83.

Bergstra, J., R. Bardenet, Y. Bengio, and B. Kégl. 2011. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, eds., *Advances in Neural Information Processing Systems*, vol. 24. Curran Associates, Inc.

Bergstra, J., and Y. Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* 13:281–305.

Berk, J. B., and R. C. Green. 2004. Mutual fund flows and performance in rational markets. *Journal of Political Economy* 112:1269–95.

Berk, J. B., and J. H. van Binsbergen. 2015. Measuring skill in the mutual fund industry. *Journal of Financial Economics* 118:1–20.

———. 2016. Assessing asset pricing models using revealed preference. *Journal of Financial Economics* 119:1–23.

Bessembinder, H., M. J. Cooper, and F. Zhang. 2023. Mutual fund performance at long horizons. *Journal of Financial Economics* 147:132–58.

———. 2025. The (large) effect of return horizon on fund alpha. *Critical Finance Review* 14:389–424.

Bianchi, D., M. Büchner, and A. Tamoni. 2021. Bond risk premiums with machine learning. *Review of Financial Studies* 34:1046–89.

Breiman, L. 2001. Random forests. *Machine Learning* 45:5–32.

Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen. 1984. *Classification and regression trees*. Chapman and Hall/CRC.

Cao, S., W. Jiang, J. Wang, and B. Yang. 2024. From man vs. machine to man + machine: The art and ai of stock analyses. *Journal of Financial Economics* 160:103910–.

Carhart, M. M. 1997. On persistence in mutual fund performance. *Journal of Finance* 52:57–82.

Cen, X., W. W. Dou, L. Kogan, and W. Wu. 2024. Fund flows and income risk of fund managers. *Unpublished Working Paper, National Bureau of Economic Research* .

Chen, L., M. Pelger, and J. Zhu. 2024. Deep learning in asset pricing. *Management Science* 70:714–50.

Chen, M., M. X. Hanauer, and T. Kalsbach. 2024. Design choices, machine learning, and the cross-section of stock returns. *Unpublished Working Paper, Technical University of Munich* .

Chen, Q., I. Goldstein, and W. Jiang. 2010. Payoff complementarities and financial fragility: Evidence from mutual fund outflows. *Journal of Financial Economics* 97:239–62.

Chen, T., and C. Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '16, 785–94. Association for Computing Machinery.

Chevalier, J., and G. Ellison. 1997. Risk taking by mutual funds as a response to incentives. *Journal of Political Economy* 105:1167–200.

Clark, T. E., and K. D. West. 2007. Approximately normal tests for equal predictive accuracy in nested models. *Journal of Econometrics* 138:291–311.

Cooper, I., and R. Priestley. 2009. Time-varying risk premiums and the output gap. *Review of Financial Studies* 22:2801–33.

Coval, J., and E. Stafford. 2007. Asset fire sales (and purchases) in equity markets. *Journal of Financial Economics* 86:479–512.

Cremers, K. M., J. A. Fulkerson, and T. B. Riley. 2019. Challenging the conventional wisdom on active management: A review of the past 20 years of academic literature on actively managed mutual funds. *Financial Analysts Journal* 75:8–35.

Cremers, K. M., and A. Petajisto. 2009. How active is your fund manager? a new measure that predicts performance. *Review of Financial Studies* 22:3329–65.

Cremers, M., M. A. Ferreira, P. Matos, and L. Starks. 2016. Indexing and active fund management: International evidence. *Journal of Financial Economics* 120:539–60.

Dass, N., V. Nanda, and Q. Wang. 2013. Allocation of decision rights and the investment strategy of mutual funds. *Journal of Financial Economics* 110:254–77.

Del Guercio, D., and P. A. Tkac. 2002. The determinants of the flow of funds of managed portfolios: Mutual funds vs. pension funds. *Journal of Financial and Quantitative Analysis* 37:523–57.

———. 2008. Star power: The effect of monrningstar ratings on mutual fund flow. *Journal of Financial and Quantitative Analysis* 43:907–36.

DeMiguel, V., J. Gil-Bazo, F. J. Nogales, and A. A. Santos. 2023. Machine learning and fund characteristics help to select mutual funds with positive alpha. *Journal of Financial Economics* 150:103737–.

Diebold, F. X., and R. S. Mariano. 1995. Comparing predictive accuracy. *Journal of Business and Economic Statistics* 13:253–63.

Diebold, F. X., and M. Shin. 2019. Machine learning for regularized survey forecast combination: Partially-egalitarian lasso and its derivatives. *International Journal of Forecasting* 35:1679–91.

Doshi, H., R. Elkamhi, and M. Simutin. 2015. Managerial activeness and mutual fund performance. *Review of Asset Pricing Studies* 5:156–84.

Dou, W. W., L. Kogan, and W. Wu. 2024. Common fund flows: Flow hedging and factor pricing. *Journal of Finance (forthcoming)* .

Edelen, R. M. 1999. Investor flows and the assessed performance of open-end mutual funds. *Journal of Financial Economics* 53:439–66.

Elton, E. J., M. J. Gruber, and C. R. Blake. 2001. A first look at the accuracy of the crsp mutual fund database and a comparison of the crsp and morningstar mutual fund databases. *Journal of Finance* 56:2415–30.

Etula, E., K. Rinne, M. Suominen, and L. Vaittinen. 2020. Dash for cash: Monthly market impact of institutional liquidity needs. *Review of Financial Studies* 33:75–111.

Evans, R. B. 2010. Mutual fund incubation. *Journal of Finance* 65:1581–611.

Evans, R. B., and Y. Sun. 2021. Models or stars: The role of asset pricing models and heuristics in investor risk adjustment. *Review of Financial Studies* 34:67–107.

Fama, E. F., and K. R. French. 1989. Business conditions and expected returns on stocks and bonds. *Journal of Financial Economics* 25:23–49.

———. 1993. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics* 33:3–56.

———. 2010. Luck versus skill in the cross-section of mutual fund returns. *Journal of Finance* 65:1915–47.

———. 2015. A five-factor asset pricing model. *Journal of Financial Economics* 116:1–22.

Fausch, J., M. Frigg, T. Johann, E. Mussbach, C. Westheide, and W. Drobetz. 2025a. Does machine learning really help to select mutual funds with positive alpha? *Unpublished Working Paper, Lucerne University of Applied Sciences and Arts, University of Mannheim, University of Hamburg, Stockholm University* .

Fausch, J., M. Frigg, S. Ruenzi, and F. Weigert. 2025b. Machine learning mutual fund flows. *Unpublished Working Paper, Lucerne University of Applied Sciences and Arts, University of Neuchâtel, University of Mannheim* .

Ferreira, M. A., A. Keswani, A. F. Miguel, and S. B. Ramos. 2013. The determinants of mutual fund performance: A cross-country study. *Review of Finance* 17:483–525.

Ferson, W. E., and M. S. Kim. 2012. The factor structure of mutual fund flows. *International Journal of Portfolio Analysis and Management* 1:112–43.

Filippou, I., I. Psaradellis, D. Rapach, and L. Zografopoulos. 2025. Improving hedge fund return prediction: Dealing with missing data via deep learning. *Unpublished*

Working Paper, Florida State University, University of Edinburgh, Federal Reserve Bank of Atlanta, University of St. Andrews .

Franzoni, F., and M. C. Schmalz. 2017. Fund flows and market states. *Review of Financial Studies* 30:2621–73.

Frazzini, A., and O. A. Lamont. 2008. Dumb money: Mutual fund flows and the cross-section of stock returns. *Journal of Financial Economics* 88:299–322.

Giglio, S., B. Kelly, and D. Xiu. 2022. Factor models, machine learning, and asset pricing. *Annual Review of Financial Economics* 14:337–68.

Gil-Bazo, J., and P. Ruiz-Verdú. 2009. The relation between price and performance in the mutual fund industry. *Journal of Finance* 64:2153–83.

Goldstein, I., H. Jiang, and D. T. Ng. 2017. Investor flows and fragility in corporate bond funds. *Journal of Financial Economics* 126:592–613.

Goulet Coulombe, P., M. Leroux, D. Stevanovic, and S. Surprenant. 2022. How is machine learning useful for macroeconomic forecasting? *Journal of Applied Econometrics* 37:920–64.

Goulet Coulombe, P., D. Rapach, E. C. M. Schütte, and S. Schwenk-Nebbe. 2025. The anatomy of machine learning-based portfolio performance. *Unpublished Working Paper, Université du Québec à Montréal, Federal Reserve Bank of Atlanta, Aarhus University* .

Goyal, A., I. Welch, and A. Zafirov. 2024. A comprehensive 2022 look at the empirical performance of equity premium prediction. *Review of Financial Studies* 37:3490–557.

Green, J., J. R. Hand, and X. F. Zhang. 2017. The characteristics that provide independent information about average us monthly stock returns. *Review of Financial Studies* 30:4389–436.

Gruber, M. J. 1996. Another puzzle: The growth in actively managed mutual funds. *Journal of Finance* 51:783–810.

Gu, S., B. Kelly, and D. Xiu. 2020. Empirical asset pricing via machine learning. *Review of Financial Studies* 33:2223–73.

Harvey, D., S. Leybourne, and P. Newbold. 1997. Testing the equality of prediction mean squared errors. *International Journal of Forecasting* 13:281–91.

Hastie, T., R. Tibshirani, J. H. Friedman, and J. H. Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer.

Hauzenberger, N., F. Huber, and K. Klieber. 2023. Real-time inflation forecasting using non-linear dimension reduction techniques. *International Journal of Forecasting* 39:901–21.

Hillert, A., A. Niessen-Ruenzi, and S. Ruenzi. 2025. Mutual fund shareholder letters: Flows, performance, and managerial behavior. *Management Science* 71:4453–73.

Huang, J., K. D. Wei, and H. Yan. 2007. Participation costs and the sensitivity of fund flows to past performance. *Journal of Finance* 62:1273–311.

———. 2022. Investor learning and mutual fund flows. *Financial Management* 51:739–65.

Investment Company Institute. 2025. Long-term and money market mutual fund assets. https://www.ici.org/research/stats/trends_01_25.

Ippolito, R. A. 1992. Consumer reaction to measures of poor quality: Evidence from the mutual fund industry. *Journal of Law and Economics* 35:45–70.

Ivković, Z., and S. Weisbenner. 2009. Individual investor mutual fund flows. *Journal of Financial Economics* 92:223–37.

Jank, S. 2012. Mutual fund flows, expected returns, and the real economy. *Journal of Banking & Finance* 36:3060–70.

Jegadeesh, N., and C. S. Mangipudi. 2021. What do fund flows reveal about asset pricing models and investor sophistication? *Review of Financial Studies* 34:108–48.

Jegadeesh, N., and S. Titman. 1993. Returns to buying winners and selling losers: Implications for stock market efficiency. *Journal of Finance* 48:65–91.

Jensen, M. C. 1968. The performance of mutual funds in the period 1945-1964. *Journal of Finance* 23:389–416.

Kacperczyk, M., S. van Nieuwerburgh, and L. Veldkamp. 2014. Time-varying fund manager skill. *Journal of Finance* 69:1455–84.

Kamstra, M. J., L. A. Kramer, M. D. Levi, and R. Wermers. 2017. Seasonal asset allocation: Evidence from mutual fund flows. *Journal of Financial and Quantitative Analysis* 52:71–109.

Kaniel, R., Z. Lin, M. Pelger, and S. Van Nieuwerburgh. 2023. Machine-learning the skill of mutual fund managers. *Journal of Financial Economics* 150:94–138.

Kapoor, S., E. M. Cantrell, K. Peng, T. H. Pham, C. A. Bail, O. E. Gundersen, J. M. Hofman, J. Hullman, M. A. Lones, M. M. Malik, et al. 2024. Reforms: Consensus-based recommendations for machine-learning-based science. *Science Advances* 10:eadk3452–.

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*, vol. 30. Curran Associates, Inc.

Kelly, B., and D. Xiu. 2023. Financial machine learning. *Foundations and Trends® in Finance* 13:205–363.

Keswani, A., and D. Stolin. 2008. Which money is smart? mutual fund buys and sells of individual and institutional investors. *Journal of Finance* 63:85–118.

Khorana, A., and H. Servaes. 2012. What drives market share in the mutual fund industry? *Review of Finance* 16:81–113.

Khorana, A., H. Servaes, and P. Tufano. 2008. Mutual fund fees around the world. *Review of Financial Studies* 22:1279–310.

Kim, M. 2020. Cross-sectional asset pricing with fund flow and liquidity risk. *Unpublished working paper, University of Melbourne* .

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

Leippold, M., Q. Wang, and W. Zhou. 2022. Machine learning in the chinese stock market. *Journal of Financial Economics* 145:64–82.

Li, B., and A. G. Rossi. 2021. Selecting mutual funds from the stocks they hold: A machine learning approach. *Unpublished Working Paper, Wuhan University, Georgetown University* .

Li, L., K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. 2017. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research* 18:6765–816.

Lou, D. 2012. A flow-based explanation for return predictability. *Review of Financial Studies* 25:3457–89.

Lundberg, S. M., G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. 2020. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence* 2:56–67.

Lundberg, S. M., and S.-I. Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc.

Malkiel, B. G. 1995. Returns from investing in equity mutual funds 1971 to 1991. *Journal of Finance* 50:549–72.

Massa, M., J. Reuter, and E. Zitzewitz. 2010. When should firms share credit with employees? evidence from anonymously managed mutual funds. *Journal of Financial Economics* 95:400–24.

Masters, T. 1993. *Practical neural network recipes in c++*. Academic Press.

McCahery, J. A., Z. Sautner, and L. T. Starks. 2016. Behind the scenes: The corporate governance preferences of institutional investors. *Journal of Finance* 71:2905–32.

Medeiros, M. C., G. F. Vasconcelos, Á. Veiga, and E. Zilberman. 2021. Forecasting inflation in a data-rich environment: the benefits of machine learning methods. *Journal of Business & Economic Statistics* 39:98–119.

Menkveld, A. J., A. Dreber, F. Holzmeister, J. Huber, M. Johannesson, M. Kirchler, S. Neusüss, M. Razen, U. Weitzel, D. Abad-Díaz, et al. 2024. Nonstandard errors. *Journal of Finance* 79:2339–90.

Molnar, C. 2019. *Interpretable machine learning*. https://christophm.github.io/interpretable-ml-book/.

Murray, S., Y. Xia, and H. Xiao. 2024. Charting by machines. *Journal of Financial Economics* 153:103791–.

Nanda, V., Z. J. Wang, and L. Zheng. 2004. Family values and the star phenomenon: Strategies of mutual fund families. *Review of Financial Studies* 17:667–98.

Neuhierl, A., X. Tang, R. T. Varneskov, and G. Zhou. 2026. Do option characteristics predict the underlying stock returns in the cross-section? *Management Science* 72:2742–60.

Newey, W., and K. West. 1987. A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica* 55:703–08.

Pástor, L., and R. F. Stambaugh. 2003. Liquidity risk and expected stock returns. *Journal of Political Economy* 111:642–85.

Pástor, L., and M. B. Vorsatz. 2020. Mutual fund performance and flows during the covid-19 crisis. *Review of Asset Pricing Studies* 791–833.

Reuter, J., and E. Zitzewitz. 2021. How much does size erode mutual fund performance? a regression discontinuity approach. *Review of Finance* 25:1395–432.

Roussanov, N., H. Ruan, and Y. Wei. 2021. Marketing mutual funds. *Review of Financial Studies* 34:3045–94.

Roussanov, N. L., H. Ruan, and Y. Wei. 2020. Mutual fund flows and performance in (imperfectly) rational markets? *Unpublished Working Paper. Jacobs Levy Equity Management Center for Quantitative Financial Research Paper* .

Sharpe, W. F. 1966. Mutual fund performance. *Journal of Business* 39:119–38.

Sirri, E. R., and P. Tufano. 1998. Costly search and mutual fund flows. *Journal of Finance* 53:1589–622.

Soebhag, A., B. Van Vliet, and P. Verwijmeren. 2024. Non-standard errors in asset pricing: Mind your sorts. *Journal of Empirical Finance* 78:101517–.

Svozil, D., V. Kvasnicka, and J. Pospichal. 1997. Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems* 39:43–62.

Walter, D., R. Weber, and P. Weiss. 2024. Methodological uncertainty in portfolio sorts. *Unpublished Working Paper, University of Konstanz, Goethy University Frankfurt, Reykjavik University* .

Warther, V. A. 1995. Aggregate mutual fund flows and security returns. *Journal of Financial Economics* 39:209–35.

Welch, B. L. 1947. The generalization of 'student's'problem when several different population variances are involved. *Biometrika* 34:28–35.

Wermers, R. 2000. Mutual fund performance: An empirical decomposition into stock-picking talent, style, transactions costs, and expenses. *Journal of Finance* 55:1655–95.

West, K. D. 1996. Asymptotic inference about predictive ability. *Econometrica* 1067–84.

Xu, H., C. Caramanis, and S. Mannor. 2011. Sparse algorithms are not stable: A no-free-lunch theorem. *IEEE transactions on pattern analysis and machine intelligence* 34:187–93.

Yang, J. 2022. Fast treeshap: Accelerating shap value computation for trees. *Unpublished Working Paper, LinkedIn Corporation* .

Zheng, L. 1999. Is money smart? a study of mutual fund investors' fund selection ability. *Journal of Finance* 54:901–33.

Zhu, M. 2018. Informative fund size, managerial skill, and investor rationality. *Journal of Financial Economics* 130:114–34.

Zou, H., and T. Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67:301–20. doi:https://doi.org/10.1111/j.1467-9868.2005.00503.x.