Received: 30 January 2020 | Revised: 12 August 2021 | Accepted: 13 September 2021

DOI: 10.1002/jae.2885

RESEARCH ARTICLE

WILEY Journal of APPLIED ECONOMETRICS

Does model complexity add value to asset allocation? Evidence from machine learning forecasting models

Iason Kynigakis[1] | **Ekaterini Panopoulou**[2]

[1]Smurfit Graduate Business School, University College Dublin, Dublin, Ireland

[2]Essex Business School, University of Essex, Colchester, UK

Correspondence
Ekaterini Panopoulou, Essex Business School, University of Essex, Colchester, UK.
Email: a.panopoulou@essex.ac.uk

Summary

This study evaluates the benefits of integrating return forecasts from a variety of machine learning and forecast combination methods into an out-of-sample asset allocation framework. The economic evaluation of the forecasts shows that model complexity translates to improved results in the majority of cases considered, with shrinkage methods and shallow neural networks generating the highest individual performance. Overall, an investor would consistently realize superior out-of-sample gains by incorporating forecast combinations of machine learning models in the portfolio formation process.

KEYWORDS

forecast combination, machine learning, portfolio optimization, return predictability

1 | INTRODUCTION

The allocation of wealth among risky assets is one of the most important problems faced by investors. Investors' objectives and constraints along with estimating expected returns play a crucial role in constructing optimal portfolios. Since forecasting returns is quite challenging, the historical average is often used as an input in portfolio optimization. However, existing literature shows that out-of-sample return predictability adds economic value to asset allocation. This study sets out to examine whether return forecasts generated by linear and nonlinear machine learning methods and their combinations benefit portfolios consisting of stocks, bonds, and commodities, when compared with simple forecast combinations, the equal-weighted portfolio or portfolios based on the historical average.

Our study contributes primarily to two strands of literature. First, our study adds to the literature of asset allocation and portfolio formation that exploits the predictability of asset returns. There exists a rich literature in finance, such as DeMiguel et al. (2009), Duchin and Levy (2009), Kritzman et al. (2010), Kirby and Ostdiek (2012), Bianchi and Guidolin (2014), and Gao and Nardari (2018), who evaluate the out-of-sample performance of asset portfolios relative to simple benchmarks such as the equal-weighted portfolio. Specifically, we relate to the literature that incorporates machine learning in an asset allocation framework (see Callot et al., 2019; D'Hondt et al., 2020). Our contribution to this strand of literature arises from investigating the benefits of integrating return forecasts from machine learning methodologies into an out-of-sample portfolio optimization framework, by comparing the alternative portfolios to the widely used benchmarks of the equal-weighted portfolio and portfolios based on the historical average forecast.

Second, it contributes to the growing literature that uses machine learning methodologies to forecast economic and financial variables. The methodologies we employ have been applied in the context of macroeconomic forecasting using a large number of predictors. Specifically, studies using shrinkage methods to examine the predictability of key macroeconomic indicators include Bai and Ng (2008), De Mol et al. (2008), Stock and Watson (2012), Carrasco and Rossi (2016), Kotchoni et al. (2019), and Babii et al. (2019). The advantages of machine learning in the context of asset

pricing and return predictability have been explored among others by Rapach et al. (2013), Neely et al. (2014), Lima and Meng (2017), Kelly et al. (2019), Rapach et al. (2019), and Kozak (2019). Gu et al. (2020), Bianchi et al. (2020), and Kim and Swanson (2014) provide a comprehensive comparison of the predictive accuracy of machine learning methodologies for the equity premium, bond risk premia, and key macroeconomic variables, respectively. Our contribution to this literature stems from exploring the economic value of a wide range of machine learning methods when used to model stock, bond, and commodity returns used as inputs to asset allocation.

We employ a variety of machine learning methods along with forecast combination schemes to generate the return forecasts for individual stocks, bonds, and commodities. Specifically, we consider shrinkage methods with a wide range of convex and non-convex penalties, supervised and unsupervised dimensionality reduction techniques, ensembles of regression trees, support vector machines, artificial neural networks, and methods that combine forecasts not only from single predictor models, but also those from the machine learning approaches.

To explore the potential benefits of using the machine learning methods in an asset allocation setting, we construct portfolios based on the return forecasts generated from the multivariate prediction models. We compare the out-of-sample performance of portfolios utilizing machine learning forecasts to that of the equal-weighed portfolio and a mean–variance portfolio based on the historical average forecast. The analysis is conducted for a conservative and an aggressive investor and for different levels of leverage. Additionally, we conduct robustness analysis and investigate how alternative estimates of the covariance matrix affect the performance of the portfolios. We explore the performance of the portfolios in terms of the average characteristics of the weight vectors and variable importance and evaluate the models for the full sample and around NBER-dated recessions and expansions. Finally, we examine the performance of the portfolios incorporating transaction costs based on a penalized mean–variance objective function that dampens the effects of transaction costs to portfolio returns.

Overall, our study shows that using machine learning techniques can be beneficial for the out-of-sample portfolio performance. Our asset allocation results show that the majority of the portfolios outperforms the equal-weighted and historical average portfolio benchmarks. When comparing portfolios across different combinations of weight constraints, our findings indicate that allocations that allow leverage further improve the performance of portfolios based on machine learning methods and that machine learning methods benefit more the portfolios of an aggressive investor. However, we observe that mean–variance portfolios exhibit similar out-of-sample performance across different specifications of the covariance matrix. Furthermore, we find that portfolios based on machine learning consist primarily of stocks and commodities, with bonds being a small part of portfolios belonging to a conservative investor. For the equal-weighted and historical average and most of the simple forecast combination portfolios, we find higher Sharpe ratios during expansionary periods. The pattern is similar for most of the dimensionality reduction and nonlinear machine learning methods, while for shrinkage methods and machine learning forecast combinations, the pattern is reversed. Finally, when introducing transaction costs, the performance of the portfolios deteriorates due to the high degree of turnover, even when using the modified mean–variance objective function. However, portfolios based on forecast combinations of machine learning models tend to yield the best performance.

The remainder of this study is organized as follows. Section 2 describes the methods used to generate the out-of-sample return forecasts used in portfolio optimization. Section 3 provides details on the data, sample splitting, and hyperparameter tuning. Section 4 presents the asset allocation framework and examines the economic value of the forecasts, Section 5 investigates further the portfolio performance, while Section 6 concludes.

2 | RETURN PREDICTION MODELS

Let r_t denote the return of an asset at time t and $\mathbf{r} = (r_1, r_2, ..., r_T)$ the $T \times 1$ vector of asset returns. We express by $x_{i,t-1}$ the i th predictor at time $t-1$, $\mathbf{x}_{t-1} = \left(x_{1,t-1}, x_{2,t-1}, ..., x_{p,t-1}\right)'$ is the $1 \times p$ vector of p candidate predictors and $\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_{T-1})$ the $T \times p$ matrix of predictors. The equation we use to model an asset's return takes the following general form:

$$r_t = f(\mathbf{x}_{t-1}) + \varepsilon_t. \tag{1}$$

Once the model $\widehat{f}$ has been estimated, the expected return of an asset at time $t+1$ using data available through t is $E_t(r_{t+1}) = \widehat{f}(\mathbf{x}_t)$. This prediction can then be used in asset allocation.

A commonly used method is the classic normal linear regression model, estimated by ordinary least squares (OLS). In this case, the form to approximate f is the following linear function:

$$f_{\boldsymbol{\theta}}(\mathbf{X}) = \alpha + \mathbf{X}\boldsymbol{\beta}, \tag{2}$$

where α is the intercept and $\boldsymbol{\beta} = \left(\beta_1, \beta_2, ..., \beta_p\right)$ is the coefficient vector. The estimates of the parameters $\boldsymbol{\theta} = (\alpha, \boldsymbol{\beta})$ are obtained by minimizing the residual sum of squares:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \mathcal{L}(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\mathbf{r} - f_{\boldsymbol{\theta}}(\mathbf{X})\|^2, \tag{3}$$

where $\mathcal{L}(\cdot)$ indicates the least squares loss and $\|\cdot\|$ denotes the l_2 norm. First, we consider the kitchen sink (KS) model, which is a multivariate prediction model utilizing all p predictors. The forecast at $t+1$ of a regression with p predictors is given by $\widehat{r}_{t+1} = \widehat{\alpha} + \mathbf{x}_t\widehat{\boldsymbol{\beta}}$.

It is well known that this model tends to have poor forecasting performance, as the estimated parameters have low bias but high variance. This problem becomes more acute as the number of predictors increases. Another simple model we consider is the historical average (HA), where the forecast is estimated as $\widehat{r}_{t+1} = (1/t)\sum_{s=1}^{t} r_s$.

In this study, we focus on the evaluation of machine learning methods for estimating f in the context of mean–variance portfolio optimization. To this end, we consider alternative models that belong to shrinkage, dimensionality reduction, nonlinear machine learning methods, and forecast combinations that combine the estimates of simple bivariate prediction models or those of machine learning models.

2.1 | Shrinkage methods

In general, shrinkage methods regularize the coefficient estimates and involve fitting the model in all p predictors. These procedures shrink the coefficients towards zero relative to the OLS estimates and aim at significantly reducing the respective coefficient variances. Shrinkage methods can also perform variable selection depending on the type of regularization. A shrinkage method is similar to the simple linear model, in that it considers only the baseline, untransformed predictors; however, it modifies the least squares problem by adding one additional term in the loss function. In the most general form, a shrinkage method includes a penalty term in the loss function:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}}[\mathcal{L}(\boldsymbol{\theta}) + \mathcal{P}(\boldsymbol{\beta}; \cdot)], \text{where } \mathcal{P}(\boldsymbol{\beta}; \cdot) = \sum_{i=1}^{p} \mathcal{P}(\beta_i; \cdot). \tag{4}$$

There are several choices for the penalty function $\mathcal{P}(\cdot)$.[1] We consider the shrinkage methods with the following penalties: ridge, lasso, elastic net, adaptive lasso, bridge, smoothly clipped absolute deviation, minimax concave penalty and smooth integration of counting and absolute deviation.

The first set of shrinkage methods is all based on convex penalties that can be derived by the following penalty function:

$$\mathcal{P}(\beta_i; \lambda; \gamma; a) = \lambda \widehat{w}_i \left[\gamma|\beta_i| + (1-\gamma)\beta_i^2\right], \text{ with } \widehat{w}_i = \frac{1}{\left|\widehat{\beta}_i\right|^a}, \tag{5}$$

where $\lambda > 0$ is a tuning parameter, which is determined separately and controls the amount of shrinkage and γ is a hyperparameter that controls the trade-off between l_1 and l_2 regularization. The parameter $\widehat{w}_i$ is the weight

[1]Note that the intercept, α, is not included in the penalty term. The penalty is applied to the coefficient vector $\boldsymbol{\beta}$ that measures the association of each predictor with the asset returns and not the intercept, which is a measure of the mean value of the asset returns when, $\mathbf{x}_1 = \mathbf{x}_2 = ... = \mathbf{x}_p = 0$. Penalization on the intercept is not typically considered, since it would make the optimization procedure dependent on the origin chosen for the asset returns; that is, adding a constant to each observation of the asset returns would not simply result in a shift of the predictions by the same amount.

corresponding to coefficient $|\beta_i|$, $\widehat{\beta}_i$ is the OLS estimate, and $a \geq 0$ is a hyperparameter that controls the strength of the weight.

When $a = 0$ and $\gamma = 0$, then the above function becomes *ridge regression* (Hoerl & Kennard, 1970) that shrinks the coefficients towards zero and when $a = 0$ and $\gamma = 1$ it yields the *least absolute shrinkage and selection operator* (lasso), introduced by Tibshirani (1996), which allows for both shrinkage and variable selection, by setting some of the coefficients equal to zero. The *elastic net* (EN), proposed by Zou and Hastie (2005), combines both l_1 and l_2 terms in the penalty, thus simultaneously performing continuous shrinkage and automatic variable selection and can also select groups of correlated variables. The penalty produces the EN solution when $a = 0$ and $\gamma \in (0,1)$. The *adaptive lasso* (ALasso), developed by Zou (2006), solves the drawback of the original lasso, which is that it does not necessarily satisfy the oracle properties (Fan & Li, 2001). This is achieved by modifying the lasso to include adaptive weights that penalize individual coefficients less severely. The adaptive lasso solution is derived by setting $a > 0$ and $\gamma = 1$.

The convex penalties described above are combinations of l_1 and l_2 terms that often select models that are overly dense (Fan & Li, 2001). It is typical in such situations to turn to greedier methods such as those that employ non-convex penalties that include an l_0 term, which is typically associated to best-subsets selection and penalizes the number of non-zero coefficients in the model. These penalties achieve sparser solutions for the same or improved prediction accuracy and enjoy superior variable-selection properties.

Bridge regression developed by Frank and Friedman (1993) and Friedman (2012) has a penalty term based on the l_γ norm and is given by

$$\mathcal{P}(\beta_i;\lambda;\gamma) = \lambda|\beta_i|^\gamma, \tag{6}$$

where $\lambda > 0$ and $\gamma \geq 0$ are the two tuning parameters. The bridge penalty term for $0 \leq \gamma \leq 2$ represents all the penalties between ridge regression and best-subsets selection. When using the squared error loss, it includes ridge ($\gamma = 2$), the lasso ($\gamma = 1$), and best-subsets ($\gamma = 0$), which produces the sparsest solutions by forcing many coefficients to be equal to zero and applies no shrinkage to the non-zero coefficients. For $\gamma > 1$, all coefficients are strictly non-zero, and all penalties in the power family are convex, while for $\gamma < 1$, the penalties are non-convex.

The *smoothly clipped absolute deviation* (SCAD), proposed by Fan and Li (2001), is a non-convex penalty function given by

$$\mathcal{P}(\beta_i;\lambda;\gamma) = \begin{cases} \lambda|\beta_i|, & \text{if } |\beta_i| \leq \lambda \\ \dfrac{2\gamma\lambda|\beta_i| - |\beta_i|^2 - \lambda^2}{2(\gamma-1)}, & \text{if } \lambda < |\beta_i| \leq \gamma\lambda \\ \dfrac{\lambda^2(\gamma+1)}{2}, & \text{if } |\beta_i| > \gamma\lambda \end{cases} \tag{7}$$

for $\gamma > 2$. SCAD coincides with the lasso until $|\beta_i| = \lambda$, then smoothly transitions to a quadratic function until $|\beta_i| = \gamma\lambda$ and then it remains constant for all $|\beta_i| > \gamma\lambda$. For small coefficients, the SCAD penalty has similar penalization rate as the lasso but leaves large coefficients not excessively penalized.

The *minimax concave penalty* (MCP), developed by Zhang (2010), is defined by

$$\mathcal{P}(\beta_i;\lambda;\gamma) = \begin{cases} \lambda|\beta_i| - \dfrac{|\beta_i|^2}{2\gamma}, & \text{if } |\beta_i| \leq \lambda\gamma \\ \dfrac{\gamma\lambda^2}{2}, & \text{if } |\beta_i| > \lambda\gamma \end{cases} \tag{8}$$

for each value of $\lambda > 0$ and $\gamma > 1$, there is a continuum of penalties and threshold operators varying from hard thresholding ($\gamma \to 1+$) to soft thresholding ($\gamma \to \infty$). MCP starts with the same rate of penalization as the lasso but smoothly relaxes the penalization rate to zero as the absolute value of the coefficient increases. Furthermore, MCP relaxes the penalization rate immediately, compared with SCAD, where the rate remains flat for a while before decreasing.

Finally, the *smooth integration of counting and absolute deviation* (SICA) penalty (Lv & Fan, 2009) takes the form

$$\mathcal{P}(\beta_i;\lambda;\gamma)=\lambda\frac{(\gamma+1)|\beta_i|}{\gamma+|\beta_i|}, \tag{9}$$

with $\lambda>0$ and a small shape parameter $\gamma>0$. SICA is another non-convex regularization method, which is a combination between the l_0 and l_1 penalties and therefore gives sparse solutions. For smaller values of γ, SICA yields results closer to the best-subsets selection, while for larger values of γ, it is closer to the lasso.[2]

2.2 | Dimensionality reduction methods

The next set of models incorporates the information of a large set of economic variables in a predictive regression framework using latent factors, which are estimated either in a supervised way (using information in both $\mathbf{r}$ and $\mathbf{X}$) or an unsupervised way (using information only in $\mathbf{X}$). The function f takes a similar form to Equation 2:

$$f(\mathbf{X})=\alpha+(\mathbf{XA})\boldsymbol{\beta}=\alpha+\mathbf{Z}\boldsymbol{\beta}, \tag{10}$$

where $\mathbf{A}=(\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,...,\boldsymbol{\alpha}_K)$ is a $p\times K$ matrix of weights, $\boldsymbol{\beta}=(\beta_1,\beta_2,...,\beta_K)$ is the coefficient vector, $\mathbf{Z}=(\mathbf{z}_0,\mathbf{z}_1,...,\mathbf{z}_{T-1})$ is a $T\times K$ matrix of latent factors, with $\mathbf{z}_{t-1}=(z_{1,t-1},z_{2,t-1},...,z_{K,t-1})'$ and $K\ll p$. The latent factor models we use differ based on how the matrix $\mathbf{A}$ is derived.

Partial least squares (PLS), introduced by Wold (1966) and more recently by Kelly and Pruitt (2015), identifies the features in a supervised way. Specifically, PLS decomposes the matrix of predictors $\mathbf{X}$ and the zero-mean vector of asset returns $\mathbf{r}$ into the form: $\mathbf{X}=\mathbf{ZP}'+\mathbf{E}$ and $\mathbf{r}=\mathbf{Zq}'+\boldsymbol{e}$, where the matrix $\mathbf{P}$ and the vector $\mathbf{q}$ are the loadings, while $\mathbf{E}$ and $\boldsymbol{e}$ are the residuals. In order to find the PLS component matrix $\mathbf{Z}$, the columns of the weight matrix $\mathbf{A}$ need to be obtained through successive optimization problems. The criterion to find the k^{th} estimated weight vector $\boldsymbol{a}_k$ is

$$\underset{\boldsymbol{a}}{\operatorname{argmax}}[\boldsymbol{\alpha}'(\mathbf{X}'\mathbf{r}\mathbf{r}'\mathbf{X})\boldsymbol{a}]\ \text{ s.t. }\ \boldsymbol{\alpha}'\boldsymbol{\alpha}=1,\ \ \boldsymbol{\alpha}'\boldsymbol{\Sigma}_{\mathrm{XX}}\boldsymbol{\alpha}=0, \tag{11}$$

where $\boldsymbol{\Sigma}_{\mathrm{XX}}$ is the sample covariance of $\mathbf{X}$. The version of PLS we employ is SIMPLS proposed by de Jong (1993). If $K=p$, then PLS would give a solution equivalent to the OLS estimates.

Sparse partial least squares (SPLS) is an extension of PLS that imposes the l_1 penalty to promote sparsity onto a surrogate weight vector $\boldsymbol{c}$ instead of the original weight vector $\boldsymbol{a}$, while keeping $\boldsymbol{a}$ and $\boldsymbol{c}$ close to each other (Chun & Keleş, 2010). The first SPLS weight vector solves

$$\underset{\boldsymbol{a},\boldsymbol{c}}{\operatorname{argmin}}\left[(\boldsymbol{c}-\boldsymbol{a})'(\mathbf{X}'\mathbf{r}\mathbf{r}'\mathbf{X})(\boldsymbol{c}-\boldsymbol{a})+\lambda_1\|\boldsymbol{c}\|_1+\lambda_2\|\boldsymbol{c}\|^2\right]\ \text{ s.t. }\ \boldsymbol{\alpha}'\boldsymbol{\alpha}=1, \tag{12}$$

where λ_1 and λ_2 are non-negative tuning parameters. To solve SPLS, a large λ_2 value is usually required, and setting $\lambda_2=\infty$ yields a solution that has the form of the soft threshold estimator by Zou and Hastie (2005). This reduces the number of tuning parameters to two, the tuning parameter λ_1 and the number of latent factors K.

In the dimensionality reduction methods described above the directions that best represent the predictors $\mathbf{X}$ are derived in a supervised way since the vector of asset returns, $\mathbf{r}$, is used to determine the component directions. The next set of models derives the latent factors in an unsupervised way.

Principal component analysis (PCA) is the most widely used method to obtain estimates of the latent factors. PCA can be viewed as a regression-type problem (see, e.g., Friedman et al., 2009) where the goal is to find the first K principal component weight vectors by minimizing

$$\underset{\mathbf{A}}{\operatorname{argmin}}\|\mathbf{X}-\mathbf{XAA}'\|^2,\quad \text{s.t.}\quad \mathbf{A}'\mathbf{A}=\mathbf{I}_K. \tag{13}$$

[2]For all penalized methods, we choose $\lambda\in[10^{-3},10^3]$. For the elastic net, $\gamma\in[0.1,0.9]$ so that the solutions differ from those of ridge or the lasso, while for the adaptive lasso $a\in[0.1,2]$. For the bridge penalty $\gamma\in[0.1,1.9]$, excluding $\gamma=1$, so that the ridge and lasso solutions are not included. For SCAD, $\gamma\in[3.7,10]$, 3.7 being the recommended value in Fan and Li (2001). For MCP, $\gamma\in[1.5,100]$, while for SICA we used the values, $\gamma=\{10^{-2},10^{-1}\}$.

The solution to this problem is most often obtained via singular value decomposition: $\mathbf{X}=\mathbf{UDV}'$, by setting $\mathbf{A}=\mathbf{V}$. The columns of $\mathbf{V}=(\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_K)$ are the principal components weights. Each $\mathbf{v}_k$ is used to derive the k th principal component, $\mathbf{z}_k=\mathbf{X}\mathbf{v}_k$; thus, $\mathbf{ZV}$ is the dimension reduced version of the original predictors. The derived variable $\mathbf{z}_1$ is the first principal component of $\mathbf{X}$ and has the largest sample variance amongst all normalized linear combinations of the columns of $\mathbf{X}$.

Sparse principal component analysis (SPCA), developed by Zou et al. (2006), is based on the regression/reconstruction property of PCA and produces modified principal components with sparse weights, such that each principal component is a linear combination of only a few of the original predictors. They show how PCA can be viewed in terms of a ridge regression problem and by adding the l_1 penalty; they convert it to an elastic net regression, which allows for the estimation of sparse principal components. The following regression criterion is proposed to derive the sparse principal component weights:

$$\underset{\mathbf{A},\mathbf{C}}{\operatorname{argmin}}\left[\|\mathbf{X}-\mathbf{XAC}'\|^2+\lambda_1\|\boldsymbol{c}\|_1+\lambda_2\|\boldsymbol{c}\|^2\right]\ \text{s.t.}\ \mathbf{A}'\mathbf{A}=\mathbf{I}_K, \tag{14}$$

where $\mathbf{C}$ is $p\times K$. If $\lambda_1=\lambda_2=0$, $T>p$ and restrict $\mathbf{C}=\mathbf{A}$, then the minimizer of the objective function is exactly the first K weight vectors of ordinary PCA. When $p\gg T$, in order to obtain a unique solution, $\lambda_2>0$ is required. The l_1 penalty on $\boldsymbol{c}_k$ induces sparseness of the weights, with larger values of λ_1 leading to sparser solutions. The algorithm by Zou and Hastie (2005) is used to compute the sparse approximations of each principal component.

Independent component analysis (ICA), developed by Comon (1994), aims at finding a linear representation of non-Gaussian data so that the components are statistically independent. The ICA objective is

$$\underset{\mathbf{A}}{\operatorname{argmin}}\|\mathbf{XA}\|_1,\quad \text{s.t.}\quad \mathbf{A}'\mathbf{A}=\mathbf{I}_K. \tag{15}$$

Solving the ICA problem amounts to finding an orthogonal $\mathbf{A}$ such that the components of the vector random variable $\mathbf{Z}=\mathbf{XA}$ are independent and non-Gaussian. More in detail, the independent components are estimated by iterative estimation of the matrix $\mathbf{A}$, systematically increasing the degree of independence of the components. However, since there is no direct measure of independence, non-Gaussianity is used instead.[3]

Ordinary ICA has two drawbacks; it requires constrained optimization which can become difficult in high dimensional settings and is sensitive to whitening, a pre-processing step that decorrelates the input data, which cannot always be computed exactly when $p\gg T$. Le et al. (2011) propose *reconstruction independent component analysis* (RICA), which overcomes the drawbacks of ICA, by replacing ICA's orthonormality constraint with a reconstruction penalty. This produces the unconstrained problem:

$$\underset{\mathbf{A}}{\operatorname{argmin}}\left[\|\mathbf{XA}\|_1+\lambda\|\mathbf{X}-\mathbf{XAA}'\|^2\right], \tag{16}$$

where $\lambda>0$ is a regularization parameter. RICA is equivalent to ICA when $K<p$; data are whitened, and λ approaches infinity.[4]

2.3 | Nonlinear machine learning methods

So far, the models we described assume a linear relationship between return and predictors. In this section, we consider three types of nonlinear models from the machine learning literature: ensembles of regression trees, support vector machines, and artificial neural networks.

[3]Popular approaches for measuring independence or non-Gaussianity in ICA are based on entropy. We use the FastICA algorithm developed by Hyvärinen and Oja (2000), which uses negentropy as a measure of Non-Gaussianity.

[4]For all dimensionality reduction approaches, the number of latent factors is $K\in[1,6]$. The parameter that controls for the l_1 penalty in the sparse methods is $\lambda_1\in[10^{-2},10^3]$. Then for reconstruction ICA $\lambda\in[10^{-2},10]$.

A regression tree is a non-parametric model that is constructed using a recursive binary splitting approach (Breiman et al., 1984). At first, starting from the top of the tree, we divide the predictor space into two distinct and non-overlapping rectangular regions or leaves and then model the asset returns as the simple average of $\mathbf{r}$ within that region. Then one or both of those regions are split into two more regions, and this process continues until certain stopping criteria are met. The predictor variable upon which a branch is based, and the value where the branch is split, is chosen to minimize the forecast error. Specifically, the prediction of a tree, $\mathcal{T}$, with M leaves $R_1, R_2, ..., R_M$ and maximum depth D, is defined as $\mathcal{T}(\mathbf{X}, c, M, D) = \sum_{m=1}^{M} c_m \mathrm{I}_{\{\mathbf{X} \in R_m(D)\}}$, where $R_m(D)$ represents one of the partitions of the predictor space. The score c associated with partition m, c_m, is the simple average of returns within region R_m, written as $c_m = 1/T_m \sum_{\mathbf{X} \epsilon R_m} \mathbf{r}$, where T_m denotes the number of observations in region m. At each branch, we choose the sorting variable among the set of predictors and the split value that minimize the following loss function:

$$Q_m(c, T_m) = \frac{1}{T_m} \sum_{\mathbf{X} \in R_m} (\mathbf{r} - c_m)^2. \tag{17}$$

Branching halts when the depth, D, of the tree reaches a pre-specified threshold that is tuned adaptively using a validation sample. Regression trees are among the methods that tend to overfit, resulting in inferior out-of-sample performance. To improve the prediction accuracy of regression trees, we consider several ensemble approaches that combine a large number of trees, $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_K$, to yield a single consensus prediction.

The first set of methods is based on boosting, first proposed by Schapire (1990) and Freund (1995) for classification problems, which recursively combines a large number of shallow trees, known as "weak learners," to form an ensemble of trees with greater stability than a single more complex tree. The first ensemble method we consider is the *gradient boosting machine* (GBM), proposed by Friedman (2001), who extends boosting to regression frameworks. Gradient boosting is initialized by fitting a shallow tree, then a second tree with the same depth is used to fit the residuals of the previous model and the forecasts of these two trees are added together to form a single ensemble prediction. This procedure is repeated sequentially until the total number of iterations K is reached. The goal is to minimize the objective function, based on the square loss

$$\underset{f}{\operatorname{argmin}} \mathcal{L}(\mathbf{r}, f_K(\mathbf{X})) = \underset{f}{\operatorname{argmin}} \|\mathbf{r} - f_K(\mathbf{X})\|^2. \tag{18}$$

Gradient boosting is an additive model that can be expressed as

$$f_K(\mathbf{X}) = \sum_{k=1}^{K} f_k(\mathbf{X}, c_k, M_k, D, v), \tag{19}$$

where K is the total number of trees and f_k is given by $f_k(\mathbf{X}, v) = f_{k-1}(\mathbf{X}) + v\mathcal{T}_k(\mathbf{X}, c_k, M_k, D)$.

The parameter v controls the learning rate of the boosting procedure that scales the contribution of each tree to the ensemble by a factor of $0 < v < 1$ and prevents the model from overfitting the residuals.

We also consider a *regularized gradient boosting machine* (RGBM), which is an extension to gradient boosting that includes a regularization term in the loss function to control for the complexity of the model and avoid overfitting (Chen & Guestrin, 2016). The updated objective function is

$$\underset{f}{\operatorname{argmin}} \mathcal{L}(\mathbf{r}, f_K(\mathbf{X})) + \Omega(f_K(\mathbf{X})). \tag{20}$$

Refining the definition of a tree as $\mathcal{T}(\mathbf{X}) = w_{q(\mathbf{X})}$, where w is a vector of scores for each region and q is a function assigning each observation to the corresponding leaf, the regularization term Ω is defined as

$$\Omega(f_K(\mathbf{X})) = \gamma M_f + \frac{1}{2}\lambda \|w\|^2 + a\|w\|_1, \tag{21}$$

where the regularization parameter γ is the complexity cost by introducing an additional leaf to a tree and the parameters λ and a control the l_2 and l_1 regularization of the weights. The set of hyperparameters for gradient boosting machine is $\{K,v,D\}$ and $\{K,v,D,\gamma,\lambda,a\}$ for extreme gradient boosting.

The second set of ensemble methods is based on bootstrap aggregating or bagging (Breiman, 1996) that combines many noisy but approximately unbiased models to reduce the variance of the estimates. The baseline bagging procedure estimates $\mathcal{T}_1,\mathcal{T}_2,...,\mathcal{T}_B$ based on B different bootstrap samples of the data and then averages their forecasts to obtain a single low-variance model, given by

$$f_B(\mathbf{X}) = \frac{1}{B}\sum_{b=1}^{B}\mathcal{T}_b(\mathbf{X}, c_b, M_b, D). \tag{22}$$

In bagging, most commonly the regression trees are simply i.d. (identically distributed, but not independent), with the variance of the average estimates, as the number of bootstrapped trees increases, depending on the product of the variance of each tree and the correlation among trees (see Hastie et al., 2009). We opt to use stationary bootstrap (see Politis & Romano, 1992), instead of standard bootstrap, due to the underlying time series structure of the data.[5]

The first variation of bagging we consider is *random forests* (RF), proposed by Breiman (2001), which aims to reduce the variance of the average estimate by minimizing the correlation among the regression trees in the ensemble. The random forest procedure builds a set of de-correlated trees by considering only a randomly drawn subset of predictors for splitting at each potential branch. This lowers the average correlation among predictions and further reduces the variance relative to bagging. We also consider *extremely randomized trees* (ERT), proposed by Geurts et al. (2006), which is an extension to the random forest algorithm. This model has two distinguishing features compared with random forests: The first is that each regression tree is trained using the full sample instead of a bootstrap sample and the second is that the top-down splitting of a tree is randomized. This is achieved by selecting the value where the branch is split randomly, instead of choosing the optimal split value locally for each input variable under consideration. Specifically, the split values are selected from a uniform distribution within the empirical range of the input variable in the training set, and the randomized split that gives the highest performance is chosen to split the region. The hyperparameters to be determined using the validation set approach within each iteration of the expanding window are the number of trees in the ensemble, the depth of the trees, and the size of the randomly selected subset of predictors.[6]

Support vector machine[7] (SVM) regression is a non-parametric technique that learns a nonlinear function by mapping linear functions into high dimensional kernel induced feature space. We first consider the *epsilon-insensitive SVM* (e-SVM), as seen in Vapnik (1995), where the objective is to find a function $f(\mathbf{X})$ that deviates from $\mathbf{r}$ by a value no greater than ϵ for each training observation of $\mathbf{X}$ and at the same time is as flat as possible. The e-SVM can be formulated as a Lagrange dual problem by introducing $\boldsymbol{a}$ and $\boldsymbol{a}^*$ vectors of non-negative multipliers for each observation of $\mathbf{X}$. The dual optimization problem for ϵ-SVM becomes

$$\begin{aligned} &\underset{\boldsymbol{a}}{\operatorname{argmin}}\frac{1}{2}(\boldsymbol{a}-\boldsymbol{a}^*)'\mathbf{Q}(\boldsymbol{a}-\boldsymbol{a}^*)+\mathbf{r}'(\boldsymbol{a}-\boldsymbol{a}^*)+\epsilon\mathbf{i}'(\boldsymbol{a}+\boldsymbol{a}^*)\\ &\text{s.t. } \mathbf{i}'(\boldsymbol{a}-\boldsymbol{a}^*)=0\\ &0\le a_i,a_i^*\le C, i=1,2,...,t, \end{aligned} \tag{23}$$

where $\mathbf{i}$ is the unity vector, $\mathbf{Q}$ is a $t\times t$ positive semidefinite matrix, $Q_{i,j}\equiv K(x_i,x_j)=\phi(x_i)'\phi(x_j)$ is a nonlinear kernel[8] function, and $\phi(x)$ is a transformation that maps x to a high-dimensional space. Based on the Karush-Kuhn-Tucker conditions, only a certain number of Lagrange multipliers $\boldsymbol{a}$ and $\boldsymbol{a}^*$ will assume non-zero values. The observations

[5]We would like to thank an anonymous referee for this suggestion.

[6]For the boosting-type methods, the number of trees is set to $K=1\sim1000$, the depth to $D=\{1,2\}$ and the learning rate to $v=\{0.01,0.1\}$. The regularization parameters for RGBM are set to $\gamma\in[10^{-5},10^{-3}]$ and $\lambda,a\in[0.1,10]$. For bagging-type approaches, the number of trees is set to $K=1\sim500$, the depth to $D=\{1,2\}$ and the size of the selected subset of predictors to $[2,44]$.

[7]The equivalence of support vector machines with the lasso has been examined by Jaggi (2014) and with the elastic net by Zhou et al. (2015).

[8]Kozak (2019) applies the kernel trick to the cross section of returns.

associated with them have approximation errors equal to or larger than $\epsilon > 0$ and are referred to as support vectors. The parameter ϵ is the trade-off between the sparseness of the representation and closeness to the data, with large values of ϵ resulting to fewer support vectors and thus sparser representation of the solution.[9] The cost parameter, $C > 0$, controls the penalty imposed on observations that lie outside the epsilon margin, ϵ, and helps to prevent overfitting as it represents the balance between the flatness of $f(\mathbf{X})$ and the extent to which violations to ϵ are tolerated.

One of the problems of e-SVM is choosing the appropriate value for the parameter ϵ. The *nu-support vector machine* (nu-SVM), proposed by Schölkopf et al. (1999) and Schölkopf et al. (2000), is a method that automatically adjusts ϵ, by considering it as part of the optimization problem. The dual optimization problem for nu-SVM becomes

$$\begin{aligned} \underset{\boldsymbol{a}}{\operatorname{argmin}} \frac{1}{2}(\boldsymbol{a}-\boldsymbol{a}^*)'\mathbf{Q}(\boldsymbol{a}-\boldsymbol{a}^*)+\mathbf{r}'(\boldsymbol{a}-\boldsymbol{a}^*) \\ \text{s.t. } \mathbf{i}'(\boldsymbol{a}-\boldsymbol{a}^*)=0 \\ 0 \le a_i, a_i^* \le C, i=1,2,...,t \\ \mathbf{i}'(\boldsymbol{a}+\boldsymbol{a}^*)=C\nu, \end{aligned} \tag{24}$$

where the parameter $0 \le \nu \le 1$ determines the proportion of the number of support vectors kept in the solution with respect to the total number of observations in the dataset. The regression function is then given by

$$f(\mathbf{X}) = \sum_{i=1}^{t} (a_i^* - a_i) K(x_i, x). \tag{25}$$

We use the radial basis function (RBF) kernel, with parameter $\gamma > 0$, given by $K(x_i, x) = \exp\left(-\gamma |x_i - x|^2\right)$. The set of hyperparameters to be selected are $\{C, \gamma, \epsilon\}$ for e-SVM and $\{C, \gamma, \nu\}$ for nu-SVM.[10]

The final nonlinear method we consider is artificial neural networks, which have been shown to be universal approximators for any continuous function f (Cybenko, 1989; Hornik et al., 1989). We focus on the *multilayer perceptron* (MLP), a type of feed-forward neural network where information flows through the function being estimated from $\mathbf{X}$, through intermediate computations used to approximate f, to the output $\mathbf{r}$. These models are composed of a number of layers with multiple nodes in each layer. They consist of an input layer of the predictors, one or more hidden layers, with nodes that transform the predictors using nonlinear activation functions and an output layer that allows a final transformation of the outcome of the hidden layers to form a prediction. For the training and design of the neural networks, we follow a similar approach to Gu et al. (2020).

Feed-forward neural networks can be defined as a composition of $h^{(1)}, h^{(2)}, ..., h^{(L)}$ nonlinear activation functions for each of the L hidden layers of the network $\mathbf{Z}^{(L)} = h^{(L)} \circ ... \circ h^{(2)} \circ h^{(1)}(\mathbf{X})$, with $\mathbf{Z}^{(l)} = h\left(b_0^{(l)} + \mathbf{W}^{(l)\prime}\mathbf{Z}^{(l-1)}\right)$, for $l = 1, 2, ..., L$, where $\mathbf{Z}^{(l)}$ is the l th layer of the network with $m = 1, 2, ..., M$ nodes, $\mathbf{W}^{(l)}$ is the matrix of weights, and $b_0^{(l)}$ is the bias. For the first hidden layer, the input is the matrix of predictors, $\mathbf{Z}^{(0)} = \mathbf{X}$, such that $\mathbf{Z}^{(1)} = h\left(b_0^{(1)} + \mathbf{W}^{(1)\prime}\mathbf{X}\right)$. The results from each hidden layer are aggregated in the output layer; $f(\mathbf{X}) = b_0^{(L+1)} + \mathbf{w}^{(L+1)\prime}\mathbf{Z}^{(L)}$. We consider three different network architectures based on whether the network has one, two, or three hidden layers (MLP1, MLP2, and MLP3). The number of hidden nodes in each layer is selected according to the geometric pyramid rule by Masters (1993). The activation function applied to each node can take various forms. We follow the existing literature and use the rectified linear unit (ReLU) defined as $h(x) = \max(0, x)$, which encourages sparsity in the number of active nodes and avoids the vanishing gradient problem (Glorot et al., 2011; Nair & Hinton, 2010).

The weights and biases of the neural network are estimated by minimizing the following objective function

$$\underset{\mathbf{W}, \mathbf{b}_0}{\operatorname{argmin}} \mathcal{L}(\mathbf{r}, f(\mathbf{X})) = \underset{\mathbf{W}, \mathbf{b}_0}{\operatorname{argmin}} \|\mathbf{r} - f(\mathbf{X})\|^2. \tag{26}$$

[9] As support vectors are usually only a small subset of the training observations, this characteristic is referred to as the sparsity of the solution.

[10] The cost parameter is set to $C \in [10^{-5}, 10^{-2}]$ and the parameter of the kernel function is set to $\gamma \in [10^{-3}, 10^{3}]$. The parameter specific to e-SVM, is set to $\epsilon \in [10^{-5}, 10^{-2}]$, while the parameter of nu-SVM is set to $\nu \in [0.1, 1]$.

The estimates of the parameters of a neural network are solutions of a non-convex optimization problem. The neural network is trained using stochastic gradient descent (SGD). Unlike standard gradient descent that uses the entire training sample to evaluate the gradient at each iteration of the optimization, SGD evaluates the gradient from a random subset of the data and iteratively minimizes the objective function through back propagation.[11] Training a neural network is challenging due to the large number of parameters to be estimated and the nonconvexity of the objective function. To alleviate those concerns, we modify the loss function by adding a parameter norm penalty. We consider an elastic net type of penalty (Goodfellow et al., 2016), based on the l_1 and l_2 norm of the weights. The regularized objective then becomes

$$\underset{\mathbf{W},\mathbf{b}_0}{\operatorname{argmin}}\left[\mathcal{L}(\mathbf{r},f(\mathbf{X}))+\lambda_1\|\mathbf{W}\|_1+\lambda_2\frac{1}{2}\|\mathbf{W}\|^2\right], \tag{27}$$

where higher values of the hyperparameters $\lambda_1>0$ and $\lambda_2>0$ correspond to more regularization. Similarly to the elastic net regression, the penalty term induces sparsity to the weights. We choose to penalize only the weights of the affine transformation at each layer and leave the biases unregularized. Following Gu et al. (2020), we consider several other forms of regularization. To improve the performance of the model, we implement early stopping, batch normalization, and forecast averaging. In each iteration of the optimization algorithm, the parameter estimates are updated so as to reduce prediction errors in the training sample and then the predictive performance of the model for that iteration is evaluated using data from the validation sample. Early stopping is implemented by stopping the training process prematurely when the validation error no longer decreases. Specifically, the optimization process halts when the maximum number of iterations (100) is reached or if the validation error has not improved for five consecutive iterations, preventing overfitting and significantly speeds up the training process. In both cases, the parameter estimates of the best performing model are retrieved. Batch normalization (Ioffe & Szegedy, 2015) reduces the variability of the predictors by scaling the input of activations. It was proposed to solve the problem of internal covariate shift in which the distribution of the inputs of the hidden layer change during training, as the parameters of the previous layers change. For each node in each training step, the algorithm cross-sectionally standardizes the output of a previous activation to restore the representation power of the node. This allows to increase the stability of the neural network and increase the speed of training. Finally, to reduce the prediction variance of the neural network, due to the stochastic nature of the optimization, we adopt an ensemble approach (Dietterich, 2000; Hansen & Salamon, 1990) by forming the final prediction from the average of the forecasts from 10 models initialized from different random seeds.[12]

2.4 | Forecast combination methods

The forecast combination approach was originally proposed by Bates and Granger (1969) and can be used as an alternative approach to individual forecasting methods (see Timmermann, 2006, for a comprehensive review). Forecast combinations may be preferred over using forecasts based on individual models, since the latter could suffer from model uncertainty and instability, while combining different models can increase accuracy by including valuable information from each model.

The forecast combinations, denoted by $\widehat{r}^{C}_{t+1}$, are the weighted averages of return forecasts from m individual models and can be expressed as

$$\widehat{r}^{C}_{t+1}=\widehat{\mathbf{r}}\boldsymbol{\omega}_t, \tag{28}$$

where $\widehat{\mathbf{r}}=(\widehat{r}_{1,t+1},\widehat{r}_{2,t+1},...,\widehat{r}_{m,t+1})'$ is the vector of m individual forecasts and $\boldsymbol{\omega}_t=(\omega_{1,t},\omega_{2,t},...,\omega_{m,t})$ are the combining weights of the individual forecasts at time t. We consider two forecast combination approaches based on the type of

[11]The version of SGD we implement is the adaptive moment estimation algorithm (Adam), introduced by Kingma and Ba (2015). Adam computes individual adaptive learning rates for the model parameters using estimates of first and second moments of the gradients.

[12]The tuning parameters for the neural networks are set to $\lambda_1,\lambda_2\in[10^{-5},10^{-3}]$ the learning rate is set to 0.01 and the parameters for Adam left to the defaults. The maximum epochs are set to 100, patience is set to 5 and the forecasts are an ensemble of 10 networks.

models that are averaged. In the first approach, we combine $m = p$ number of simple bivariate prediction models based on individual predictors. This simple linear model is based on a single predictor x_i, and the forecast for $t+1$ using the i th predictor is given by

$$\widehat{r}_{i,t+1} = \widehat{a}_i + x_{i,t}\widehat{\beta}_i, \text{ for } i = 1,2,...,p.$$

This model is estimated using OLS. In the second approach, we combine the forecasts generated by the $m = 23$ machine learning models.[13]

We further differentiate the forecast combinations based on the way $\boldsymbol{\omega}_t$ is computed. Specifically, the *mean combination* (MC) sets the weight $\omega_{i,t} = 1/m$ for $i = 1,2,...,m$ and the *median combination* (MDC) is the median of $\widehat{\mathbf{r}}_{t+1}$. The *trimmed mean combination* (TMC) sets $\omega_{i,t} = 0$ for 5% of the forecasts with the lowest and highest values and $\omega_{i,t} = 1/(0.9 \times m)$ for the remaining forecasts. These simple forecast averaging schemes do not require a holdout period, q_0.

For the second type of forecasting methods, the combining weights are computed based on the historical forecasting performance of the individual models over the holdout period, q_0. Aiolfi and Timmermann (2006) consider a method based on the rank of each model according to the mean squared forecast error (MSFE). This weighing scheme lets the weights be inversely proportional to the forecast models' rank, RANK_i, that is, $\omega_{i,t} = \text{RANK}_{i,t}^{-1} / \sum_{i=1}^{m} \text{RANK}_{i,t}^{-1}$, where the model with the lowest MSFE value gets a rank of 1, the model with the second lowest MSFE value gets a rank of 2 and so forth. Aiolfi and Timmermann (2006) also consider a clustering approach to combine forecasts. Specifically, the forecasts from the individual models are grouped into L equal-sized clusters based on their past MSFE performance, with the first cluster containing the models with the lowest MSFE. Each combination forecast is the average of the individual forecasts contained in the first cluster. This procedure starts with the initial holdout period and then goes through the end of the available OOS period using a rolling window. We consider forecast combinations with two (CL2) and three (CL3) clusters.

The third type of combining methods considered is also based on past performance of the individual models and uses time-varying combination weights. Stock and Watson (2004) proposed the *discounted mean square forecast error* combining method, which uses the following weights $\omega_{i,t} = \varphi_{i,t}^{-1} / \sum_{i=1}^{m} \varphi_{i,t}^{-1}$, where $\varphi_{i,t} = \sum_{s=R}^{t-1} \psi^{t-1-q_0} (r_{s+1} - \widehat{r}_{i,s+1})^2$, for $t = R + q_0, ..., T$, where R is the in-sample period and ψ is a discount factor, with $0 < \psi \leq 1$. In the case of $\psi < 1$, this method assigns greater weights to recent individual predictive regression forecasts. When $\psi = 1$, then there is no discounting and the equation above produces the optimal combination forecast derived by Bates and Granger (1969) for the case where the individual forecasts are uncorrelated. The values for ψ considered are 1 and 0.9 (DMSFE1 and DMSFE09).

To facilitate the discussion of our findings, Table 1 lists all the models used in this study grouped in six categories.

3 | DATA AND SAMPLE SPLITTING

3.1 | Data

On the asset side, our dataset consists of 50 individual stocks, 12 bond portfolios, and 52 commodities, for a total of $N = 114$ assets. On the predictor side, we consider a common set of $p = 44$ economic indicators. The sample period starts from January 1980 to December 2019, for a period of 40 years (or $T = 480$ monthly observations).

Specifically, we obtain the total returns for the 50 stocks with the highest market capitalization from the Center for Research in Security Prices (CRSP) that have no missing return observations over the full sample period.[14] For the bond portfolios, we use the returns of the Fama maturity portfolios, also retrieved from CRSP, where the first 10 portfolios are defined according to maturity in 6-month intervals for up to 60 months, while the remaining are portfolios for maturities between 60 and 120 months and for maturities greater than 120 months. The returns of the 52 commodities are calculated from the prices acquired from the Primary Commodity Price System dataset of the International

[13]We would like to thank an anonymous referee for suggesting the forecast combination of machine learning models.

[14]We restrict our selection to stocks listed to the NYSE, AMEX, and NASDAQ stock exchanges (exchange codes 1, 2, or 3) and to ordinary common shares (share codes 10 or 11).

TABLE 1 List of models

EW	Equally weighted portfolio
HA	Historical average
KS	Kitchen sink model (OLS)
Forecast combinations of bivariate prediction models	
MC	Equally-weighted average of forecasts (mean combination)
MDC	Median combination forecast
TMC	Trimmed mean combination forecast
Rank	Weighted average of forecasts based on MSE ranking
CL2	Average of forecasts contained in the first of two clusters
CL3	Average of forecasts contained in the first of three clusters
DMSFE1	Discounted mean square forecast error, $\psi=1$
DMSFE09	Discounted mean square forecast error, $\psi=0.9$
Shrinkage methods	
Ridge	Ridge regression
Lasso	Least absolute shrinkage and selection operator
EN	Elastic net
aLasso	Adaptive least absolute shrinkage and selection operator
Bridge	Bridge regression
SCAD	Smoothly clipped absolute deviation
MCP	Minimax concave penalty
SICA	Smooth integration of counting and absolute deviation
Dimensionality reduction methods	
PCA	Principal component analysis
SPCA	Sparse principal component analysis
PLS	Partial least squares
SPLS	Sparse partial least squares
ICA	Independent component analysis
RICA	Reconstruction independent component analysis
Nonlinear machine learning methods	
RF	Random forests
ERT	Extremely randomized trees
GBM	Gradient boosting machine
RGBM	Regularized gradient boosting machine
e-SVM	Epsilon support vector machine
nu-SVM	Nu support vector machine
MLP1	Multilayer perceptron, one hidden layer
MLP2	Multilayer perceptron, two hidden layers
MLP3	Multilayer perceptron, three hidden layers
Forecast combinations of machine learning models	
MC_{ML}	Equally-weighted average of ML forecasts (mean combination)
MDC_{ML}	Median combination of ML forecasts
TMC_{ML}	Trimmed mean combination of ML forecasts
$Rank_{ML}$	Weighted average of ML forecasts based on MSE ranking
$CL2_{ML}$	Average of ML forecasts contained in the first of two clusters

TABLE 1 (Continued)

$CL3_{ML}$	Average of ML forecasts contained in the first of three clusters
$DMSFE1_{ML}$	Discounted mean square forecast error, $\psi = 1$, of ML forecasts
$DMSFE09_{ML}$	Discounted mean square forecast error, $\psi = 0.9$, of ML forecasts

Monetary Fund (IMF) and are chosen based on having a full price history over the sample period. The majority of the commodities is classified as food or beverages, for a total of 27 assets, while the remaining belong to the following general categories: 11 are metals, 9 are agricultural raw materials, 3 are categorized as fertilizer, and 2 are energy commodities. Further details regarding the assets along with their identifiers can be found in Tables S1 to S3, for stocks, bond, and commodities respectively.

The choice of predictors draws upon the literature of return predictability, such as Rapach et al. (2005), Welch and Goyal (2008), Rapach et al. (2010), Neely et al. (2014), Lima and Meng (2017) for stocks, Ludvigson and Ng (2009), Lin et al. (2017), and Gargano et al. (2017) for bonds, and Gargano and Timmermann (2014) and Gao and Nardari (2018) for commodities. The first set of predictors are the factors used in the Fama-French 3- and 5-factor models (Fama & French, 1993, 2016) and the 4-factor model of Carhart (1997). We include the market (MKT), size (SMB), value (HML), profitability (RMW), investment (CMA), and momentum (MOM) factors. As a proxy to market liquidity, we use the Pástor and Stambaugh (2003) aggregate innovation in liquidity measure (LIQ). We also consider variables from Welch and Goyal (2008), which are the dividend-price ratio (DP), dividend yield (DY), earnings-price ratio (EP), dividend-payout ratio (DE), stock variance (SVAR), book-to-market ratio (BM), net equity expansion (NTIS), treasury bill rate (TBL), long-term yield (LTY), long-term return (LTR), term spread (TMS), default yield spread (DFY), and default return spread (DFR). Additional predictors related to the bond market are the ICE BofA US corporate total return index (BAML) and the Cochrane and Piazzesi (2005) factor (CP). In the set of predictors, we include economic indicators such as the industrial production (INDPRO), the money stock M1 (M1), the consumer price index (CPI), the producer price index (PPI), capacity utilization (CAP), the unemployment rate (UNRATE), the inventory-sales ratio (IR), housing starts (HOUST), and total consumer credit (Credit). As a proxy for the overall commodity market, we use the S&P Goldman Sachs commodity total return index (GSCI) and include four commodity currencies (Chen et al., 2010), the Australian dollar-US dollar (USAU), the Canadian dollar-US dollar (USCA), the Indian rupee-US dollar (USIN), and the New Zealand dollar-US dollar (USNZ). Finally, we consider eight additional variables, namely, the Chicago Board Options Exchange volatility index (VXO), the University of Michigan consumer sentiment index (UMSENT), the Chicago Fed national activity index (CFNAI), Kilian's (2009) real economic activity index (REA), the Philadelphia Fed business outlook survey current and future activity indices (GAC and GAF), and the macroeconomic and financial uncertainty indices (UMacro and UFin, respectively), proposed by Jurado et al. (2015) and Ludvigson et al. (2015). The candidate variables and their sources are given in Table S4, while descriptive statistics of the predictors are reported in Table S5.

3.2 | Sample splitting and hyperparameter tuning

We generate out-of-sample forecasts of asset returns by employing a recursive forecasting scheme. The total sample, T, is divided into the in-sample part, R and the out-of-sample part, $Q = T - R$. The first q_0 forecasts of the out-of-sample period, Q, serve as the hold-out period for the forecast combination methods that require it. The initial size of the recursive window, used to estimate the individual forecasting models, is set to $R = 180$ monthly observations (or 15 years, from January 1980 to December 1994) and the hold-out period, for the forecast combination methods that require it, is set to $q_0 = 60$ observations (or 5 years, from January 1995 to December 1999). The window expands by one observation at a time, leading to an out-of-sample size of 240 monthly observations from January 2000 to December 2019.

The machine learning models used to generate the return forecasts rely on hyperparameter tuning. The choice of hyperparameters controls the amount of model complexity and is critical for the performance of the model. Specifically, we adopt a validation sample approach similar to Gu et al. (2020), in which the optimal set of values for the tuning parameters is selected in the validation sample. One of the advantages of using this approach over k-fold cross validation is that we maintain the temporal ordering of the data. Specifically, in each iteration of the expanding window, the in-sample is split into two disjoint periods, the training subsample, consisting of 80% of the observations, with the remaining observations belonging to the validation subsample. In the training subsample, the model is estimated for

several sets of hyperparameters. The second subsample is used to select the optimal set of tuning parameters, by constructing forecasts, using the model estimates from the training sample for the respective hyperparameter set, for the observations in the validation sample. The optimal set of hyperparameters is chosen so as to minimize the mean squared error over the validation subsample. Once the optimal set of hyperparameters is chosen, the model is refitted using all data from the in-sample period and the estimates of the model parameters are kept to construct the forecasts. The predictors are standardized for all methods, first separately for observations within each of the training and validation subsamples, during the selection of the optimal set of hyperparameters, and then using all observations of the in-sample window, when estimating the parameters of the model. For a detailed description of cross-validation, see Friedman et al. (2009). Due to the computational cost of training these methods, we avoid recursively fitting the models in each iteration. Instead, we estimate the parameters once a year and retain those estimates to derive the remaining forecasts for that year.

4 | OPTIMAL ASSET ALLOCATION

Consider an investor who allocates her wealth among N individual assets with a $N \times 1$ portfolio weight vector: $\mathbf{w} = (w_1, w_2, ..., w_N)$. The initial wealth is normalized to 1. The benchmark strategy is the naive diversification rule of an equal-weighted portfolio, where $w_j = 1/N$, for $j = 1, 2, ..., N$. The objective of the main framework is to optimize the trade-off between risk and return. As a basic measure of portfolio risk, the standard deviation of the portfolio (Markowitz, 1952) is used. The mean–variance (MV) optimization problem is

$$\underset{\mathbf{w}}{\operatorname{argmin}}[\gamma \mathbf{w}' \boldsymbol{\Sigma} \mathbf{w} - \mathbf{w}' \widehat{\mathbf{r}}], \tag{29}$$

where $\boldsymbol{\Sigma}$ is the $N \times N$ covariance matrix of asset returns, $\widehat{\mathbf{r}} = (\widehat{r}_{1,t+1}, \widehat{r}_{2,t+1}, ..., \widehat{r}_{N,t+1})$ is the $N \times 1$ vector of return forecasts for each asset, and γ is the coefficient of relative risk aversion. As an alternative to the $1/N$ benchmark, portfolios using historical average forecasts are considered. The two benchmarks are compared with portfolios based on forecasts generated by multivariate prediction models.

All portfolio models include short-selling and leverage constraints to avoid implausible positions. The first constraint sets an upper bound to the sum of the portfolio weights, $\mathbf{w}' \mathbf{i}_N = h$, where $\mathbf{i}_N$ is an N-vector of ones and h denotes the maximum leverage, for example, $h = 1$ ensures that the portfolio weights sum up to one, while $h = 1.5$ indicates that the investor cannot borrow more than 50% of total wealth. The second constraint sets a lower bound to the weight of each asset as $w_j \geq 0$, with $j = 1, ..., N$, which leads to portfolios without short selling (Jagannathan & Ma, 2003). The portfolio return at $t+1$ can then be computed as

$$r_{P,t+1} = \widehat{\mathbf{w}}_t' \mathbf{r} + \left(1 - \widehat{\mathbf{w}}_t' \mathbf{i}_N\right) r_{f,t+1}, \tag{30}$$

where $\mathbf{r} = (r_{1,t+1}, r_{2,t+1}, ..., r_{N,t+1})$ is an N-vector of risky asset returns. In the case of $h = 1$, the portfolio return is equivalent to $r_{P,t+1} = \widehat{\mathbf{w}}_t' \mathbf{r}$. For the mean–variance optimization framework, two approaches are used to estimate the covariance matrix. The first is the graphical lasso, which is a static estimator that derives a sparse version of the covariance matrix. The second is a dynamic estimator based on the DCC GARCH. The estimates of the covariance matrix, $\widehat{\boldsymbol{\Sigma}}$, for assets $i = 1, 2, ..., N$, are based on a rolling window of 120 observations.

The graphical lasso algorithm, proposed by Friedman et al. (2008), estimates the sparse precision matrix (inverse of the covariance matrix), using the l_1 (lasso) penalty to enforce sparsity. The graphical lasso problem is to maximize the following penalized log likelihood:

$$\log(\det \boldsymbol{\Theta}_t) - \operatorname{tr}(\mathbf{S}_t \boldsymbol{\Theta}_t) - \rho \|\boldsymbol{\Theta}_t\|_1, \tag{31}$$

where $\mathbf{S}_t$ is the sample covariance matrix and $\rho \geq 0$ is a tuning parameter controlling the amount of regularization. Here, $\boldsymbol{\Theta}_t = \boldsymbol{\Sigma}_t^{-1}$, with entries $\theta_{i,j}$, is the $N \times N$ inverse of the covariance matrix and $\|\boldsymbol{\Theta}_t\|_1$ is the l_1 norm of $\boldsymbol{\Theta}_t$—the sum of the absolute value of the elements $\theta_{i,j}$. The penalty parameter ρ is chosen by the validation sample approach, to make

the value of $\log\left(\det\boldsymbol{\Sigma}_{1,t}^{-1}\right)-\mathrm{tr}\left(\boldsymbol{\Sigma}_{2,t}\boldsymbol{\Sigma}_{1,t}^{-1}\right)$ large, where $\boldsymbol{\Sigma}_{1,t}$ is the covariance matrix estimated using the training set and $\boldsymbol{\Sigma}_{2,t}$ is the covariance estimated over the validation set.

For the dynamic conditional correlation GARCH model, proposed by Engle (2002), the one-period ahead covariance based on the DCC GARCH model evolves according to

$$\boldsymbol{\Sigma}_{t+1}=\mathbf{D}_{t+1}\mathbf{R}_{t+1}\mathbf{D}_{t+1}, \tag{32}$$

where $\mathbf{D}_{t+1}$ is an $N\times N$ diagonal matrix with conditional standard deviation $\widehat{\sigma}_{i,t+1}$ on the i th diagonal element and $\mathbf{R}_{t+1}$ is the $N\times N$ correlation matrix, with ones on the diagonal and conditional correlations in the off-diagonal. The estimation of the DCC GARCH has two steps. The first step involves estimating the diagonal elements of the conditional standard deviation matrix, $\mathbf{D}_{t+1}$, where we estimate the conditional standard deviation, $\widehat{\sigma}_{i,t+1}$, of the i th asset is using a GARCH(1,1) model. The second step involves the estimation of the conditional correlation matrix, $\mathbf{R}_{t+1}$. Removing the conditional mean from the N series of asset returns yields the residuals, $\boldsymbol{\varepsilon}_{t+1}$ and the standardized residuals, $\mathbf{u}_{t+1}$, can be obtained using the conditional standard deviation matrix, $\mathbf{D}_{t+1}$: $\mathbf{u}_{t+1}=\mathbf{D}_{t+1}^{-1}\boldsymbol{\varepsilon}_{t+1}$. The conditional correlation structure then is

$$\begin{aligned}\mathbf{Q}_{t+1}&=(1-a-b)\overline{\mathbf{Q}}+a\mathbf{u}_t\mathbf{u}_t'+b\mathbf{Q}_t\\ \mathbf{R}_{t+1}&=\mathbf{Q}_{t+1}^{*^{-1}}\mathbf{Q}_{t+1}\mathbf{Q}_{t+1}^{*^{-1}},\end{aligned} \tag{33}$$

where $\overline{\mathbf{Q}}$ is the unconditional covariance of the standardized residuals and $\mathbf{Q}_{t+1}^{*}$ is a diagonal matrix composed of the square root of the diagonal elements of $\mathbf{Q}_{t+1}$.

4.1 | Portfolio performance

In this section, we assess the economic value of using return forecasts in asset allocation. The portfolios are constructed recursively using the related return and covariance estimates in each iteration, starting in January 2000. The buy-and-hold portfolio returns are calculated for the period of 1 month, and the portfolio is rebalanced monthly until the end of the evaluation period (December 2019). Each portfolio is computed for different combination of weight constraints: unleveraged long-only portfolios ($0\le w_j\le 1$) and leverage restricted to 50% of wealth ($0\le w_j\le 1.5$). Two types of investors are considered based on different values of the coefficient of risk aversion, $\gamma=2$ for an aggressive investor and $\gamma=10$ for a conservative investor. Similar values of risk aversion are used in Callot et al. (2017) and DeMiguel et al. (2009).

The performance of the portfolios is evaluated over the out-of-sample period using the average return of the portfolio and the out-of-sample Sharpe ratio. The Sharpe ratio (SR) is calculated as the fraction of the out-of-sample excess return (average realized return less the risk-free rate) divided by the standard deviation of the out-of-sample portfolio returns: $\mathrm{SR}=\left(\overline{r}_P-r_f\right)/\widehat{\sigma}_P^2$, where $\overline{r}_P=1/(Q-q_0)\sum_{t=1}^{Q-q_0}r_{P,t}$ is the average realized return of the portfolio over the out-of-sample period, r_f is the risk free rate, and $\widehat{\sigma}_P$ is the standard deviation of the portfolio excess returns over the out-of-sample period. Additionally, we report portfolio turnover, defined as a measure of the amount of trading required to implement a particular strategy. Following DeMiguel et al. (2009), the portfolio turnover is defined as the average absolute change of the portfolio weights over the $Q-q_0$ rebalancing periods across the N assets and is given as follows:

$$\mathrm{PT_P}=\frac{1}{Q-q_0-1}\sum_{t=1}^{Q-q_0-1}\sum_{j=1}^{N}\left|w_{j,t+1}-w_{j,t}\right|, \tag{34}$$

where $w_{j,t+1}$ is the weight in asset j at time $t+1$ and $w_{j,t}$ is the weight in asset j at time before rebalancing at $t+1$. When a portfolio is rebalanced at $t+1$, $\left|w_{j,t+1}-w_{j,t}\right|$ denotes the magnitude of trading asset j.

Table 2 presents the results of the portfolio evaluation based on average return. The first row gives the average return of the $1/N$ portfolio, which is 5.73% across all panels since derivation of the weights for this strategy does not involve any optimization or estimation. The remaining rows of the table report the average return of the mean–variance portfolio based on the historical average, KS, and alternative forecasting models. These results vary based on the estimator of the covariance matrix, the type of investor, and combination of weight constraints.

TABLE 2 Portfolio performance based on average return

	A. Sparse covariance				B. Dynamic covariance			
	Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)		Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)	
Model	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**
EW	5.73	5.73	5.73	5.73	5.73	5.73	5.73	5.73
HA	4.00	7.12	6.06	9.16	3.69	6.77	6.52	10.28
KS	24.96	37.38	20.43	27.29	23.74	33.74	18.84	25.79
Forecast combinations of bivariate prediction models								
MC	5.29	8.30	6.65	10.22	5.83	8.79	6.88	10.58
MDC	5.53	9.05	6.62	9.81	5.15	8.48	7.06	10.65
TMC	5.39	8.28	6.51	9.91	5.58	8.53	6.90	10.54
Rank	8.11	12.56	7.17	10.81	8.32	12.06	7.68	11.86
CL2	4.41	7.05	6.40	9.81	5.32	7.78	6.27	10.24
CL3	4.78	7.33	6.48	10.12	5.68	7.92	5.99	10.16
DMSFE1	5.29	8.30	6.66	10.24	5.82	8.72	6.86	10.60
DMSFE09	5.34	8.30	6.64	10.28	5.82	8.56	6.85	10.71
Shrinkage methods								
Ridge	25.31	36.55	20.20	27.40	23.89	33.15	17.97	24.76
Lasso	23.49	32.02	18.13	24.35	22.81	31.95	15.37	20.62
EN	23.30	31.59	18.03	24.13	22.65	31.48	15.23	20.20
aLasso	22.43	31.67	18.11	24.79	21.10	29.56	15.15	20.65
Bridge	22.09	30.50	18.68	26.23	21.07	29.44	16.31	23.41
SCAD	21.17	29.63	18.92	25.92	20.50	27.84	15.94	23.94
MCP	25.19	35.76	20.21	27.48	24.60	34.26	18.52	26.04
SICA	21.31	30.36	17.27	23.86	19.78	28.09	15.25	20.98
Dimensionality reduction methods								
PCA	9.44	14.11	9.43	12.67	9.99	14.66	9.41	12.69
SPCA	9.64	16.14	9.52	12.44	9.31	16.14	10.25	13.11
PLS	24.06	34.96	16.94	21.27	21.86	31.56	14.52	17.79
SPLS	21.14	29.52	13.58	17.53	20.45	29.33	13.80	17.07
ICA	11.23	15.53	7.51	9.95	13.56	18.26	9.16	12.18
RICA	8.68	12.28	8.11	11.68	7.26	9.71	6.62	9.23
Nonlinear machine learning methods								
RF	12.19	17.12	9.64	13.48	11.93	16.54	8.77	11.79
ERT	5.88	9.76	7.60	11.45	6.78	10.09	7.59	11.56
GBM	10.88	16.54	11.50	17.16	11.22	16.72	10.53	15.50
RGBM	9.98	15.33	9.55	13.57	6.68	10.92	6.12	8.43
e-SVM	9.48	13.70	8.77	12.63	6.82	9.78	7.91	11.67
nu-SVM	7.05	11.78	6.86	10.59	7.63	13.03	8.66	12.55
MLP1	24.66	34.66	18.05	24.75	22.52	31.98	16.03	20.87
MLP2	13.12	18.73	10.35	13.51	11.96	17.06	7.60	10.71
MLP3	6.79	10.16	6.83	10.20	7.10	8.72	5.63	8.62
Forecast combinations of machine learning models								
MC_{ML}	25.17	34.12	16.77	21.11	24.05	32.11	13.98	18.33

TABLE 2 (Continued)

	A. Sparse covariance				B. Dynamic covariance			
	Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)		Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)	
Model	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**
MDC_{ML}	24.52	36.15	17.25	20.82	22.64	31.98	15.32	19.23
TMC_{ML}	25.25	34.99	16.87	21.63	24.24	32.68	13.94	18.75
$Rank_{ML}$	19.60	30.09	16.54	21.37	18.89	28.18	14.50	19.08
$CL2_{ML}$	23.45	35.10	19.07	24.84	24.28	36.41	17.46	21.47
$CL3_{ML}$	24.01	35.62	18.24	23.88	26.12	38.53	18.17	22.71
$DMSFE1_{ML}$	25.49	35.66	17.07	21.55	24.33	33.70	14.41	18.61
$DMSFE09_{ML}$	27.08	38.93	17.93	23.21	26.60	37.28	15.49	19.23

Note: This table reports the annualized average return of the mean–variance portfolios, with monthly rebalancing, for the out-of-sample period of January 2000 to December 2019. Panel A reports the performance based on a sparse covariance estimator estimated using the graphical lasso, while the results of panel B are based on a dynamic covariance estimated using the DCC GARCH. The performance is reported for different levels of risk aversion ($\gamma = 2, 10$), for unleveraged portfolios ($w_j \in [0,1]$) and portfolios with leverage ($w_j \in [0,1.5]$). The models with the highest average return are underlined.

Focusing on the results of the first panel, which are based on the graphical lasso estimates of the covariance matrix, the average return of the HA portfolio for an aggressive investor ($\gamma = 2$) is 4% for $w_j \in [0,1]$ and 7.12% for $w_j \in [0,1.5]$. On the other hand, the average return of a conservative investor ($\gamma = 10$), for weight constraints $0 \leq w_j \leq 1$, is 6.06% and when $0 \leq w_j \leq 1.5$, it is 9.16%. For the portfolios based on the KS forecasts, the average returns range from 24.96% to 37.38% for an aggressive investor and from 20.43% to 27.29% for a conservative investor. Turning to the results of the second panel, HA portfolios of a conservative investor benefit more when using the DCC estimator instead of the graphical lasso, while for the KS portfolios, the sparse estimator yields higher average returns across different combinations of weight constraints and levels of risk aversion.

Overall, forecast combinations of machine learning models tend to achieve the best performance in terms of average return, especially for an aggressive investor. Individual models that perform well are those based on shrinkage methods, partial least squares, sparse partial least squares, and a neural network with a single hidden layer. Relaxing the leverage constraint consistently improves the performance of the models. The average return of the mean–variance portfolios, based on the graphical lasso, across all alternative predictive regressions, for an aggressive investor, is between 4.41% (CL2) and 27.08% ($DMSFE09_{ML}$) for the case when no short sales or leverage is allowed. For a 50% leverage constraint, the respective returns are 7.05% and 38.93%. In the case of a conservative investor, for weight constraints $0 \leq w_j \leq 1$, the average return is between 6.40% (CL2) and 20.21% (MCP), and when leverage is allowed, returns range from 9.81% (MDC and CL2) to 27.48% (MCP). Comparing the results between the two specifications for the covariance matrix, the graphical lasso outperforms the DCC, except in the case of portfolios based on certain forecast combinations of bivariate prediction models and for portfolios based on nu-SVM. The majority of the machine learning models and their combinations outperform the equal-weighted allocation and the HA benchmarks, while portfolios based on the KS model prove to be harder benchmarks to overcome. The model that fails to outperform the EW portfolio is the neural network with three hidden layers, for portfolios based on the dynamic covariance estimator and $\gamma = 10$, while the HA portfolio yields higher average return than RGBM and a neural network with three hidden layers for portfolios of a conservative investor based on the DCC covariance.

Table 3 reports the results for the annualized Sharpe ratio. Our findings indicate that between 18 and 19 models for an aggressive investor and 20 to 25 models for a conservative investor outperform the EW benchmark, with a ratio of 0.78, depending on the weight constraints and estimates of the covariance matrix. The simple forecast combination methods, along with ensemble methods and support vector machines fail to outperform the EW in the majority of specifications. The Sharpe ratio for the HA portfolio with $\gamma = 2$ varies between 0.25 and 0.37 and is outperformed by 37 to 39 of the models, while for a conservative investor, the ratio is between 0.59 and 0.74, with 32 to 37 models generating better performance. More importantly, the models that consistently outperform both benchmarks are the forecast combinations of machine learning models. For example, for an aggressive investor with a 50% leverage constraint, the Sharpe ratio ranges from 0.95 ($Rank_{ML}$) to 1.28 ($DMSFE09_{ML}$).

TABLE 3 Portfolio performance based on Sharpe ratio

	A. Sparse covariance				B. Dynamic covariance			
	Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)		Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)	
Model	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**
EW	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78
HA	0.30	0.37	0.59	0.67	0.25	0.34	0.63	0.74
KS	1.05	1.16	1.25	1.25	0.96	1.00	1.14	1.16
Forecast combinations of bivariate prediction models								
MC	0.37	0.42	0.65	0.76	0.39	0.42	0.65	0.75
MDC	0.40	0.47	0.64	0.72	0.35	0.41	0.69	0.77
TMC	0.38	0.42	0.63	0.74	0.37	0.41	0.66	0.75
Rank	0.46	0.54	0.68	0.79	0.44	0.49	0.69	0.81
CL2	0.30	0.35	0.62	0.72	0.35	0.38	0.60	0.73
CL3	0.31	0.35	0.63	0.75	0.37	0.39	0.57	0.72
DMSFE1	0.37	0.42	0.65	0.76	0.39	0.42	0.65	0.75
DMSFE09	0.37	0.42	0.64	0.76	0.39	0.41	0.65	0.76
Shrinkage methods								
Ridge	1.07	1.14	1.25	1.27	0.96	0.99	1.11	1.16
Lasso	0.96	0.95	1.10	1.12	0.86	0.89	0.90	0.93
EN	0.94	0.93	1.09	1.12	0.85	0.87	0.89	0.91
aLasso	0.91	0.93	1.09	1.12	0.80	0.83	0.89	0.93
Bridge	0.90	0.91	1.17	1.21	0.80	0.84	0.99	1.07
SCAD	0.86	0.89	1.19	1.19	0.79	0.81	0.98	1.10
MCP	1.05	1.10	1.25	1.27	0.97	1.01	1.12	1.19
SICA	0.86	0.90	1.09	1.12	0.73	0.78	0.92	0.97
Dimensionality reduction methods								
PCA	0.48	0.54	0.78	0.79	0.50	0.56	0.75	0.76
SPCA	0.39	0.50	0.76	0.80	0.39	0.50	0.82	0.81
PLS	1.08	1.12	1.05	0.98	0.95	1.00	0.87	0.79
SPLS	0.92	0.92	0.84	0.81	0.85	0.86	0.83	0.78
ICA	0.48	0.48	0.56	0.56	0.58	0.56	0.63	0.64
RICA	0.41	0.44	0.62	0.67	0.35	0.35	0.49	0.51
Nonlinear machine learning methods								
RF	0.59	0.61	0.72	0.76	0.57	0.59	0.65	0.64
ERT	0.37	0.46	0.74	0.83	0.40	0.45	0.70	0.79
GBM	0.45	0.48	0.65	0.74	0.46	0.47	0.59	0.66
RGBM	0.43	0.47	0.53	0.57	0.29	0.33	0.37	0.38
e-SVM	0.55	0.57	0.79	0.90	0.39	0.40	0.68	0.76
nu-SVM	0.44	0.53	0.62	0.74	0.46	0.57	0.76	0.83
MLP1	1.01	1.02	1.09	1.14	0.91	0.90	0.90	0.87
MLP2	0.59	0.61	0.77	0.79	0.53	0.58	0.55	0.60
MLP3	0.36	0.40	0.59	0.68	0.36	0.33	0.44	0.52
Forecast combinations of machine learning models								
MC_{ML}	1.16	1.16	1.16	1.13	1.07	1.04	0.94	0.94

TABLE 3 (Continued)

	A. Sparse covariance				B. Dynamic covariance			
	Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)		Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)	
Model	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**
MDC_{ML}	1.04	1.11	1.09	1.00	0.95	0.96	0.96	0.92
TMC_{ML}	1.16	1.18	1.17	1.16	1.07	1.06	0.95	0.98
$Rank_{ML}$	0.86	0.95	1.15	1.17	0.84	0.88	1.02	1.06
$CL2_{ML}$	1.00	1.04	1.17	1.19	1.07	1.13	1.15	1.11
$CL3_{ML}$	0.99	1.04	1.08	1.08	1.13	1.18	1.15	1.14
$DMSFE1_{ML}$	1.17	1.19	1.18	1.15	1.06	1.07	0.98	0.97
$DMSFE09_{ML}$	1.22	1.28	1.23	1.23	1.18	1.21	1.08	1.03

Note: This table reports the annualized Sharpe ratio of the mean–variance portfolios, with monthly rebalancing, for the out-of-sample period of January 2000 to December 2019. Panel A reports the performance based on a sparse covariance estimator estimated using the graphical lasso, while the results of panel B are based on a dynamic covariance estimated using the DCC GARCH. The performance is reported for different levels of risk aversion ($\gamma = 2, 10$), for unleveraged portfolios ($w_j \in [0,1]$) and portfolios with leverage ($w_j \in [0,1.5]$). The models with the highest Sharpe ratio are underlined.

The KS portfolio generates ratios from 0.96 to 1.16 for an aggressive investor and between 1.14 and 1.25 for a conservative investor. The Sharpe ratio for portfolios with relative risk aversion of 2 is between 0.29 and 1.22 for unleveraged and long only portfolios and 0.33 to 1.28 for a 50% leverage constraint. Among the best-performing portfolios are those with return forecasts generated by machine learning model combinations, PLS, Ridge, and MLP1. Based on different weight constraints, the Sharpe ratios for a conservative investor are higher compared with those of a more aggressive investor, with values from 0.37 to 1.25 when $w_j \in [0,1]$ and 0.38 to 1.27 for $w_j \in [0,1.5]$. Shrinkage methods, PLS, MLP1, and ML forecast combinations are among the models that yield the highest ratios across all weight constraints. Finally, portfolios based on graphical lasso tend to generate slightly higher Sharpe ratios than the DCC GARCH.

The first part of Figure 1 presents the cumulative return series of the EW and HA benchmarks and the eight portfolios based on forecast combinations of machine learning models of an aggressive investor with 50% leverage. The eight strategies significantly outperform the EW and HA benchmarks throughout the out-of-sample period, especially after 2002–2003. The cumulative return series of the forecast combinations exhibit overall an upward trend and experience a slight downward trend during the global financial crisis. The discounted mean square forecast error scheme with $\psi = 0.9$ displays superior gains compared to the other models after about 2013 and achieves the highest end-of-period value. The worst performing forecast combination is based on the rank scheme, while the cumulative returns of the remaining models evolve in a similar way over time.

The second part of Figure 1 depicts the cumulative return of the EW and HA benchmarks and the eight portfolios based on forecast combinations of machine earning models of a conservative investor without any leverage. Overall, the performance of the eight portfolios of the conservative investor is more stable over time compared with those of an aggressive investor; however, the end of period value is approximately half that of the portfolios with $\gamma = 2$. The eight portfolios considerably outperform the HA and EW benchmarks, especially after the global financial crisis, with the portfolio based on a cluster combination scheme of two clusters yielding the highest end-of-period-value.

We report the annualized average turnover of the portfolios as a percentage in Table 4. The portfolio with the lowest turnover (except for the $1/N$) is the HA, with values between 6.55% and 10.58%, in the case of graphical lasso and from 16.12% to 39.31% for the DCC. The majority of the portfolios of an aggressive investor has a higher turnover than those of a conservative investor. Additionally, most of the portfolios based on the DCC have higher turnover, the exceptions being shrinkage methods, PLS, and certain machine learning combinations in the case of $\gamma = 2$ and neural networks for most of the combinations of portfolio parameters.

Among the multivariate prediction models, the simple forecast combinations generate the lowest values, with turnover between 12.67% (MDC) and 164.42% (Rank) for portfolios based on sparse covariance estimates and from 27.19% (MDC) to 191.68% (Rank) for the dynamic covariance estimator. Machine learning models have a turnover in the range of 57.35% and 276.97%, while combinations of ML models generate turnover from 147.25% to 264.40%. Portfolios based

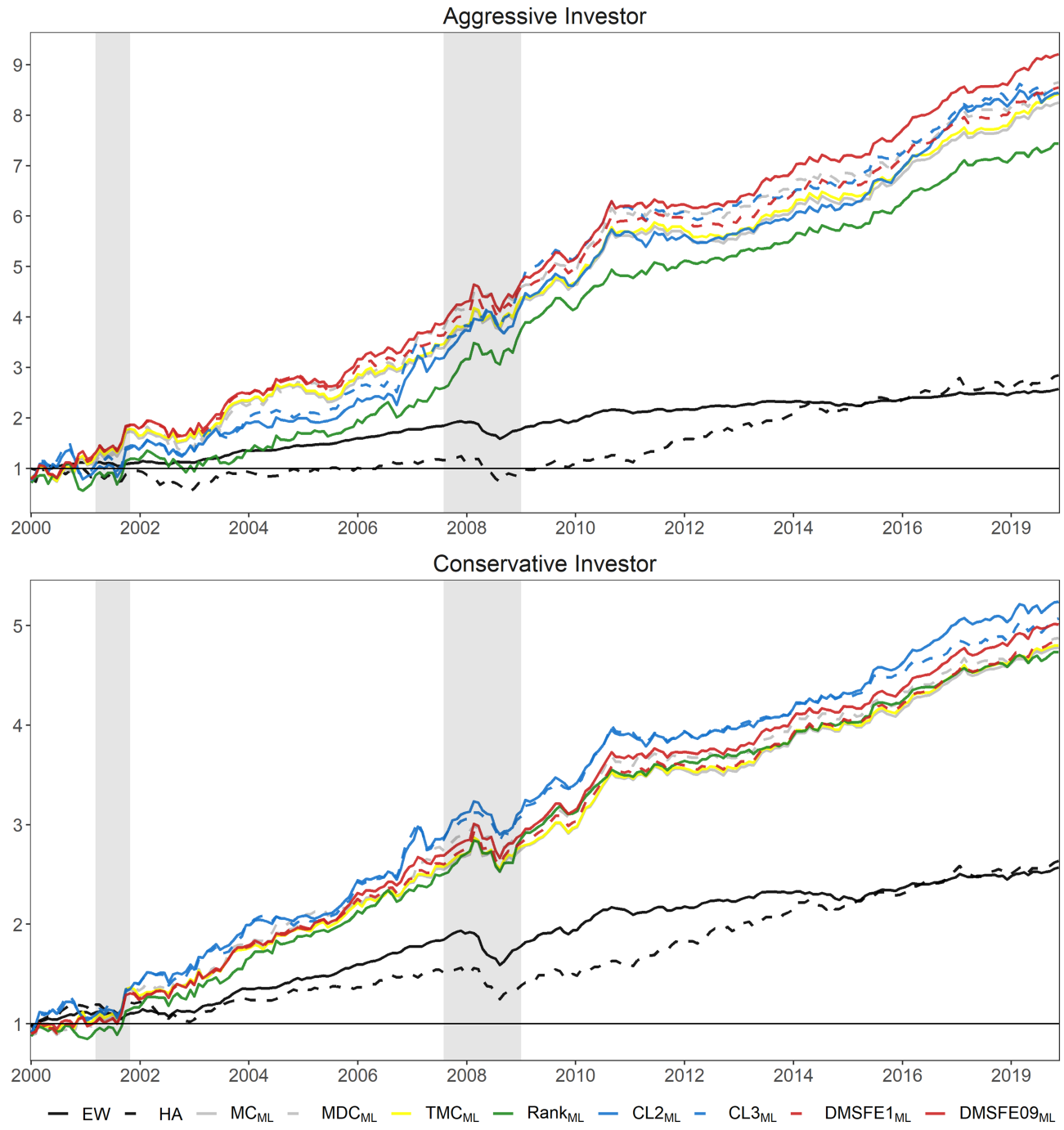

FIGURE 1 Cumulative returns for portfolios without transaction costs. The mean–variance portfolios are based on the graphical lasso estimator, with combination of weight constraints $w_j \in [0, 1.5]$, for the aggressive investor ($\gamma = 2$) and $w_j \in [0, 1]$, for the conservative investor ($\gamma = 10$). The shaded regions depict the NBER-dated recessions and expansions

on extremely randomized trees have the lowest turnover, while a neural network with three hidden layers consistently produces the highest average turnover across all models.

4.2 | Asset selection and portfolio weights

The percentage of wealth invested in each asset class throughout the out-of-sample period is reported in Table 5 for portfolios based on the graphical lasso. In the case of the EW portfolio 43.86% is invested in stocks, 10.53% in bonds

TABLE 4 Portfolio performance based on average turnover

	A. Sparse covariance				B. Dynamic covariance			
	Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)		Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)	
Model	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**
EW	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
HA	8.04	9.79	6.55	10.58	16.12	24.82	23.35	39.31
KS	182.42	273.82	177.69	260.47	180.35	270.92	178.03	263.34
Forecast combinations of bivariate prediction models								
MC	40.67	48.02	24.61	35.98	51.88	67.30	36.82	59.07
MDC	21.04	24.48	12.67	19.02	29.62	40.78	27.19	45.93
TMC	37.17	44.13	22.71	33.30	47.37	61.99	34.49	56.34
Rank	122.36	164.42	82.44	113.65	136.22	191.68	102.25	145.23
CL2	59.60	74.43	38.22	55.29	69.89	92.74	48.42	79.69
CL3	78.89	103.17	52.67	74.59	90.25	122.22	63.01	98.70
DMSFE1	40.83	48.31	24.77	36.18	52.05	67.56	36.92	59.20
DMSFE09	40.73	48.51	24.73	36.19	52.09	67.83	36.71	58.71
Shrinkage methods								
Ridge	182.75	274.05	177.30	258.44	180.47	270.85	178.39	263.68
Lasso	177.26	267.33	170.80	247.97	174.01	262.86	175.47	259.47
EN	178.04	268.31	171.49	248.62	175.21	264.24	175.77	259.54
aLasso	177.83	267.44	170.35	246.90	175.67	263.87	175.10	258.56
Bridge	182.63	273.42	175.50	257.16	180.47	270.65	177.79	264.61
SCAD	182.63	274.20	174.96	255.89	180.92	270.54	177.20	263.06
MCP	183.03	274.55	177.54	260.03	181.34	271.11	178.87	265.27
SICA	181.75	271.61	174.18	254.07	180.34	269.76	176.32	261.25
Dimensionality reduction methods								
PCA	151.17	215.06	118.22	167.30	157.19	230.38	130.80	187.16
SPCA	155.49	224.36	128.32	179.96	163.11	239.16	142.87	205.50
PLS	177.51	266.15	169.86	246.51	174.65	263.06	172.22	253.15
SPLS	177.85	265.59	168.19	243.13	178.80	266.75	174.92	258.38
ICA	152.13	224.07	128.19	179.37	155.72	230.42	138.87	198.97
RICA	153.47	223.24	131.73	188.33	154.96	228.85	140.82	204.02
Nonlinear machine learning methods								
RF	151.46	219.92	123.11	170.48	154.25	229.41	138.38	196.37
ERT	82.87	110.56	57.37	80.06	101.30	138.07	72.43	106.63
GBM	178.06	266.65	172.91	255.86	178.92	268.34	174.85	259.51
RGBM	181.99	271.01	173.65	254.98	182.68	272.69	177.34	260.04
e-SVM	121.57	176.16	105.33	151.77	126.40	189.28	119.60	175.38
nu-SVM	117.27	172.07	96.80	135.37	124.46	184.33	114.53	168.32
MLP1	163.82	247.16	166.66	248.59	160.98	242.38	163.10	244.87
MLP2	177.05	266.60	175.71	259.82	174.24	262.03	172.86	258.34
MLP3	185.95	276.97	181.64	269.58	185.44	276.57	180.35	269.61
Forecast combinations of machine learning models								
MC_{ML}	176.62	263.17	166.56	241.71	175.56	261.14	169.32	250.21
MDC_{ML}	176.91	263.59	167.65	243.65	177.58	264.40	169.98	250.88

(Continues)

TABLE 4 (Continued)

	A. Sparse covariance				B. Dynamic covariance			
	Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)		Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)	
Model	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**
TMC_{ML}	177.22	263.41	165.86	239.91	176.16	262.47	169.94	250.85
$Rank_{ML}$	168.14	250.13	150.69	211.93	169.14	253.28	158.75	228.07
$CL2_{ML}$	166.36	246.79	151.69	216.84	166.34	248.31	158.09	230.43
$CL3_{ML}$	165.37	244.75	147.25	208.76	166.25	248.27	156.09	225.24
$DMSFE1_{ML}$	175.92	261.89	165.69	239.49	174.48	260.09	168.90	250.31
$DMSFE09_{ML}$	175.81	261.93	165.96	239.93	175.35	263.88	171.18	253.19

Note: This table reports the annualized average turnover of the mean–variance portfolios, with monthly rebalancing, for the out-of-sample period of January 2000 to December 2019. Panel A reports the performance based on a sparse covariance estimator estimated using the graphical lasso, while the results of panel B are based on a dynamic covariance estimated using the DCC GARCH. The performance is reported for different levels of risk aversion ($\gamma = 2, 10$), for unleveraged portfolios ($w_j \in [0,1]$) and portfolios with leverage ($w_j \in [0,1.5]$). The models with the highest turnover are underlined.

and 45.61% in commodities. Inspecting the table, we observe that the remaining models allocate little to no wealth to bonds. Focusing on the results for unleveraged portfolios, the HA and simple forecast combination methods tend to invest primarily in stocks with only a small percentage (approximately 5% to 10% depending on the degree of risk aversion) invested in commodities. The KS model and shrinkage methods show similar results, placing approximately equal weights between stocks and commodities. Portfolios based on dimensionality reduction and nonlinear methods invest on average 70% in stocks and 30% on commodities, while in the case of forecast combinations of machine learning models the split is 60% to stocks and 40% to commodities.

Figure 2 presents the average contribution of each asset to the portfolio across the OOS period. The 30 assets with the highest contribution to the portfolio of an aggressive and a conservative investor are reported. The assets that contribute most to the portfolio of an aggressive investor are 20 stocks and 10 commodities, while for the conservative investor there are 18 stocks and 12 commodities. The commodities with the highest contributions according to both investors are bananas, lead, nickel, oats, oil, pork, silver, sugar (free market), uranium, and urea, while the portfolio of a conservative investor places significant weights also on iron and orange juice. Portfolios based on the HA, simple forecast combinations, unsupervised dimensionality reduction methods, bagging-type ensembles, and support vector machines tend to focus heavily on certain stocks such as Walmart (WMT) and Altria Group (MO), while the remaining methods allocate wealth across assets more evenly.

4.3 | Portfolio performance during business cycles

In this section, we examine the performance of the portfolios during NBER dated recessions and expansions. Table 6 reports the results for the average return. For the EW, HA, and the majority of the simple forecast combinations, the average return is positive during expansions but negative during recessions, the exception being the rank combination scheme for an aggressive investor that generates higher and positive returns during recessions. For the forecast combinations of machine learning models, the portfolios of an aggressive investor and those of a conservative investor without any leverage also yield better performance during expansions; however, compared with the simple forecast combinations, the returns during both subperiods are positive. This pattern persists for most of the dimensionality reduction or nonlinear methods and is reversed for shrinkage methods, where the models yield higher return during recessionary subperiods.

We report the results in terms of Sharpe ratio in Table 7. Our findings suggest that the portfolios of a conservative investor are consistently higher during expansions than in recessions, with most models generating positive Sharpe ratios during both subperiods, except for EW, HA, and forecast combinations of bivariate regression models that have negative ratios during recessions. For an aggressive investor, the performance based on the Sharpe ratio is qualitatively similar to that based on average return, with the exception of forecast combinations of machine learning models that tend to have higher ratios during recessions than expansions.

TABLE 5 Average weight in each asset class for portfolios based on the sparse covariance estimator

	Aggressive ($\gamma = 2$)						Conservative ($\gamma = 10$)					
	[0, 1]			[0, 1.5]			[0, 1]			[0, 1.5]		
Model	**S**	**B**	**C**	**S**	**B**	**C**	**S**	**B**	**C**	**S**	**B**	**C**
EW	43.86	10.53	45.61	43.86	10.53	45.61	43.86	10.53	45.61	43.86	10.53	45.61
HA	97.46	0.00	2.54	142.69	0.00	7.31	90.71	1.29	8.00	120.23	17.09	12.68
KS	50.25	0.00	49.75	75.33	0.00	74.67	48.49	0.00	51.51	70.38	0.00	79.62
Forecast combinations of bivariate prediction models												
MC	96.21	0.00	3.79	141.59	0.00	8.41	90.31	1.15	8.54	120.68	15.45	13.87
MDC	97.47	0.00	2.53	142.91	0.00	7.09	90.98	1.12	7.90	121.16	16.22	12.62
TMC	96.44	0.00	3.56	141.95	0.00	8.05	90.47	1.15	8.38	120.87	15.51	13.62
Rank	91.72	0.00	8.28	135.55	0.09	14.36	84.40	2.61	12.99	114.70	13.76	21.54
CL2	95.23	0.00	4.77	140.44	0.00	9.56	88.98	1.53	9.49	119.48	15.16	15.36
CL3	94.30	0.00	5.70	139.25	0.06	10.69	87.49	2.07	10.44	118.09	14.77	17.14
DMSFE1	96.19	0.00	3.81	141.57	0.00	8.43	90.28	1.14	8.58	120.68	15.40	13.92
DMSFE09	96.27	0.00	3.73	141.68	0.00	8.32	90.25	1.16	8.58	120.60	15.44	13.96
Shrinkage methods												
Ridge	51.01	0.00	48.99	76.76	0.00	73.24	49.32	0.00	50.68	71.47	0.00	78.53
Lasso	51.06	0.00	48.94	77.44	0.00	72.56	49.76	0.01	50.22	71.61	0.22	78.17
EN	51.03	0.00	48.97	77.35	0.00	72.65	50.04	0.01	49.95	72.08	0.20	77.72
aLasso	50.82	0.00	49.18	76.91	0.00	73.09	48.93	0.00	51.07	70.38	0.34	79.28
Bridge	51.68	0.00	48.32	77.79	0.00	72.21	49.86	0.00	50.14	72.65	0.07	77.28
SCAD	51.23	0.00	48.77	78.33	0.00	71.67	50.36	0.00	49.64	72.85	0.02	77.13
MCP	51.38	0.00	48.62	76.91	0.00	73.09	49.21	0.00	50.79	71.69	0.00	78.31
SICA	51.31	0.00	48.69	77.43	0.00	72.57	50.31	0.00	49.69	72.22	0.04	77.75
Dimensionality reduction methods												
PCA	74.36	2.60	23.04	111.13	4.53	34.34	68.38	5.83	25.79	95.56	13.54	40.91
SPCA	74.02	1.58	24.40	109.68	3.06	37.25	68.01	4.86	27.13	95.23	11.72	43.05
PLS	57.22	0.00	42.78	86.80	0.00	63.20	56.12	0.18	43.70	81.04	0.84	68.12
SPLS	63.61	0.00	36.39	95.33	0.00	54.67	59.49	0.61	39.89	85.28	2.32	62.40
ICA	79.66	0.13	20.20	118.22	0.51	31.27	71.71	2.22	26.07	99.58	7.06	43.37
RICA	73.15	0.67	26.18	108.35	1.34	40.31	66.17	2.33	31.51	92.72	6.52	50.76
Nonlinear machine learning methods												
RF	64.60	0.10	35.30	98.25	0.28	51.48	68.30	0.81	30.89	100.72	4.24	45.05
ERT	92.97	0.01	7.02	138.11	0.29	11.60	86.41	1.28	12.31	118.98	10.85	20.17
GBM	50.09	0.00	49.91	75.63	0.00	74.37	50.34	0.05	49.61	73.41	0.14	76.45
RGBM	59.12	0.00	40.88	87.64	0.00	62.36	55.25	0.08	44.66	80.05	0.35	69.59
e-SVM	93.38	0.17	6.45	139.02	0.68	10.29	84.27	4.65	11.07	111.32	18.61	20.07
nu-SVM	84.91	0.00	15.09	126.30	0.13	23.57	77.05	2.05	20.89	104.28	9.41	36.31
MLP1	53.18	0.00	46.82	81.59	0.00	68.41	55.71	0.10	44.19	80.95	0.33	68.72
MLP2	55.15	0.30	44.55	82.56	0.61	66.82	49.51	2.75	47.74	68.41	6.31	75.29
MLP3	54.16	1.92	43.92	80.54	3.48	65.98	45.63	7.18	47.19	61.65	15.96	72.39
Forecast combinations of machine learning models												
MC_{ML}	59.53	0.00	40.47	90.03	0.00	59.97	57.63	0.27	42.10	81.44	1.22	67.34
MDC_{ML}	58.40	0.00	41.60	87.90	0.06	62.03	56.32	0.55	43.13	80.78	2.08	67.14

(Continues)

TABLE 5 (Continued)

	Aggressive ($\gamma = 2$)						Conservative ($\gamma = 10$)					
	[0, 1]			[0, 1.5]			[0, 1]			[0, 1.5]		
Model	**S**	**B**	**C**	**S**	**B**	**C**	**S**	**B**	**C**	**S**	**B**	**C**
TMC_{ML}	59.51	0.00	40.49	90.46	0.00	59.54	58.15	0.33	41.52	82.29	1.38	66.32
$Rank_{ML}$	59.41	0.00	40.59	89.33	0.00	60.67	61.39	0.86	37.75	89.78	3.69	56.54
$CL2_{ML}$	53.18	0.00	46.82	79.41	0.00	70.59	54.46	1.00	44.54	81.02	3.39	65.59
$CL3_{ML}$	52.55	0.00	47.45	79.03	0.13	70.84	53.57	1.51	44.92	80.12	4.71	65.17
$DMSFE1_{ML}$	60.17	0.00	39.83	91.50	0.00	58.50	59.12	0.39	40.49	83.70	1.69	64.61
$DMSFE09_{ML}$	61.71	0.00	38.29	93.48	0.00	56.52	59.34	0.33	40.33	83.44	1.68	64.87

Note: This table reports the average weights in each of the three asset classes (Stock, Bond, and Commodities), for the out-of-sample period of January 2000 to December 2019. The mean–variance portfolios are based on the graphical lasso estimator. The average weights are reported for different levels of risk aversion ($\gamma = 2, 10$), for unleveraged portfolios ($w_j \in [0,1]$) and portfolios with leverage ($w_j \in [0,1.5]$).

FIGURE 2 Asset selection. Columns correspond to individual models and rows to the 30 assets that contribute most to the portfolio, according to the average weight across the out-of-sample period. The mean–variance portfolios are based on the graphical lasso estimator, with combination of weight constraints $w_j \in [0,1.5]$, for the aggressive investor ($\gamma = 2$) and $w_j \in [0,1]$, for the conservative investor ($\gamma = 10$). A darker colour indicates a greater influence of an asset to the portfolio utilising the return forecasts generated from the respective alternative model

TABLE 6 Portfolio performance based on average return during business cycles

	A. Expansions				B. Recessions			
	Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)		Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)	
Model	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**
EW	7.42	7.42	7.42	7.42	−8.15	−8.15	−8.15	−8.15
HA	5.87	9.96	8.13	12.11	−11.34	−16.27	−10.98	−15.05
KS	24.24	35.83	19.91	26.58	30.89	50.13	24.68	33.18
Forecast combinations of bivariate prediction models								
MC	6.18	10.16	8.53	12.77	−2.04	−6.96	−8.88	−10.77
MDC	6.82	11.24	8.50	12.52	−5.04	−9.00	−8.85	−12.52
TMC	6.34	10.31	8.44	12.56	−2.38	−8.41	−9.38	−11.88
Rank	6.67	11.39	8.36	12.69	19.96	22.11	−2.60	−4.67
CL2	5.86	9.64	8.21	12.38	−7.55	−14.20	−8.50	−11.32
CL3	5.75	9.31	8.14	12.53	−3.29	−8.96	−7.21	−9.70
DMSFE1	6.18	10.13	8.54	12.78	−2.06	−6.74	−8.84	−10.64
DMSFE09	6.29	10.15	8.52	12.78	−2.47	−6.89	−8.82	−10.24
Shrinkage methods								
Ridge	23.12	33.30	19.20	26.64	43.39	63.31	28.40	33.68
Lasso	21.91	29.73	17.70	24.65	36.47	50.86	21.66	21.91
EN	21.69	29.16	17.56	24.41	36.58	51.54	21.89	21.81
aLasso	20.66	29.07	17.47	24.47	37.05	53.06	23.37	27.45
Bridge	20.39	28.09	18.26	26.19	36.09	50.34	22.16	26.52
SCAD	19.87	27.44	18.88	26.45	31.87	47.67	19.23	21.58
MCP	23.13	32.49	19.26	26.76	42.11	62.66	28.07	33.45
SICA	18.85	26.99	17.10	24.36	41.53	58.09	18.62	19.69
Dimensionality reduction methods								
PCA	9.08	13.65	9.77	13.83	12.36	17.90	6.65	3.14
SPCA	10.26	17.33	10.83	14.68	4.56	6.30	−1.24	−5.94
PLS	24.16	35.50	17.93	23.00	23.19	30.55	8.74	7.00
SPLS	20.12	29.50	14.66	19.29	29.58	29.68	4.74	2.99
ICA	12.52	18.69	10.74	15.04	0.61	−10.41	−19.10	−31.96
RICA	10.26	14.57	9.57	14.16	−4.30	−6.58	−3.89	−8.69
Nonlinear machine learning methods								
RF	13.32	18.77	11.61	16.82	2.92	3.55	−6.52	−14.03
ERT	6.16	11.41	9.83	14.96	3.51	−3.84	−10.73	−17.44
GBM	15.41	23.26	14.43	20.56	−26.45	−38.73	−12.58	−10.82
RGBM	14.37	22.14	13.12	17.86	−26.15	−40.67	−19.78	−21.74
e-SVM	9.97	14.99	9.55	13.82	5.41	3.04	2.40	2.84
nu-SVM	8.36	13.89	8.33	12.58	−3.78	−5.57	−5.17	−5.76
MLP1	23.24	32.72	17.69	24.59	36.32	50.67	21.01	26.07
MLP2	12.29	18.37	11.50	15.43	19.94	21.73	0.82	−2.36
MLP3	8.75	12.86	7.40	10.97	−9.28	−12.03	2.17	3.86
Forecast combinations of machine learning models								
MC_{ML}	22.32	30.91	16.59	21.77	48.65	60.57	18.25	15.70
MDC_{ML}	22.94	34.37	17.22	21.36	37.57	50.75	17.51	16.40

(Continues)

TABLE 6 (Continued)

	A. Expansions				B. Recessions			
	Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)		Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)	
Model	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**
TMC_{ML}	22.49	31.70	16.58	22.28	48.03	62.04	19.19	16.35
$Rank_{ML}$	16.83	26.11	15.92	21.77	42.39	62.80	21.67	18.03
$CL2_{ML}$	21.50	31.98	18.99	25.63	39.42	60.82	19.75	18.33
$CL3_{ML}$	21.40	32.23	18.14	24.79	45.49	63.53	19.04	16.45
$DMSFE1_{ML}$	22.74	32.23	16.82	22.22	48.10	63.89	19.14	16.07
$DMSFE09_{ML}$	24.91	36.78	17.91	24.03	44.97	56.63	18.04	16.48

Note: This table reports the annualized average return of the mean–variance portfolios, with monthly rebalancing, for the out-of-sample period of January 2000 to December 2019, during business cycles. The mean–variance portfolios are based on the graphical lasso estimator. Panel A reports the performance during NBER expansions, while panel B reports the performance during NBER recessions. The performance is reported for different levels of risk aversion ($\gamma = 2, 10$), for unleveraged portfolios ($w_j \in [0,1]$) and portfolios with leverage ($w_j \in [0,1.5]$). The models with the highest average return are underlined.

4.4 | Variable importance

In this section, we present the contribution of each feature to the learning process. Gu et al. (2020) highlight the importance of quantifying the influence of each predictor as a way to interpret machine learning models. The measure of variable importance is calculated as the change in the out-of-sample R^2, as defined in Campbell and Thompson (2007), from setting the values of a predictor to zero within each iteration of the out-of-sample period and then averaging these values to obtain a single variable importance measure for each predictor of a particular model. To gain further insight as to which features contribute most for each asset class, we further summarize the results by averaging the variable importance measure, of a specific feature and model, for all assets depending on whether they are a stock, bond, or a commodity. Figure 3 reports the variable importance, for the three asset classes and the 20 most influential predictors of each model. Variables are ranked so that those with the highest variable importance are on top and the lowest are at the bottom, with the ranking starting from the first model (KS).

For stocks and the KS model, the most important features are the industrial production, market return, dividend-price ratio, capacity utilization, earnings-price ratio, and term spread. The important variables for shrinkage methods are similar to those of the KS model; however, the MKT variable becomes less important, while financial uncertainty is more influential. For dimensionality reduction methods, influential variables for PCA- and PLS-type of methods include financial uncertainty, the payout ratio, stock variance, and real economic activity index, while for ICA and RICA, the dividend-price ratio and the market are the most important features. Nonlinear machine learning methods tend to use a broader set of predictors, the exception being the neural network with a single hidden layer that focuses on dividend-price ratio and the market proxy. The variable importance results for the bond market indicate that most models use the information from the full set of predictors, except for RICA and MLP1 that are skewed towards the dividend-price ratio. For the commodities, influential predictors for most shrinkage methods include the market, dividend-price ratio, industrial production, capacity utilization, CFNAI and GSCI. Turning to the dimensionality reduction methods, ICA and RICA focus especially on the market, DP and TMS. On the other hand, the remaining methods draw information from a wider set of variables. Nonlinear methods again appear to not focus on a particular predictor, except for the market and dividend-price ratio variables in the case of MLP1.

5 | FURTHER PERFORMANCE EVALUATION AND ROBUSTNESS CHECKS

In this section, we examine portfolio performance when transactions costs are introduced, for different rebalancing frequency and for time-varying risk aversion parameter. We also perform several robustness checks, such as employing an alternative shrinkage estimator of the covariance matrix, including power series and interactions in the predictor set and using the Huber loss function as an objective function.

TABLE 7 Portfolio performance based on Sharpe ratio during business cycles

	A. Expansions				B. Recessions			
	Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)		Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)	
Model	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**
EW	1.25	1.25	1.25	1.25	−0.59	−0.59	−0.59	−0.59
HA	0.45	0.55	0.85	0.97	−0.70	−0.68	−0.74	−0.76
KS	1.09	1.20	1.35	1.34	0.90	1.04	0.94	0.95
Forecast combinations of bivariate prediction models								
MC	0.46	0.55	0.90	1.03	−0.11	−0.25	−0.59	−0.56
MDC	0.51	0.61	0.88	1.00	−0.28	−0.35	−0.59	−0.64
TMC	0.47	0.56	0.89	1.01	−0.12	−0.30	−0.61	−0.61
Rank	0.41	0.53	0.87	1.02	0.72	0.62	−0.16	−0.22
CL2	0.43	0.52	0.88	0.99	−0.34	−0.46	−0.55	−0.56
CL3	0.42	0.49	0.88	1.01	−0.13	−0.27	−0.45	−0.48
DMSFE1	0.46	0.55	0.90	1.03	−0.11	−0.24	−0.58	−0.55
DMSFE09	0.46	0.55	0.90	1.03	−0.13	−0.25	−0.58	−0.53
Shrinkage methods								
Ridge	1.04	1.12	1.32	1.37	1.30	1.34	1.08	0.96
Lasso	0.96	0.95	1.21	1.28	1.02	1.01	0.78	0.59
EN	0.94	0.93	1.20	1.27	1.03	1.03	0.79	0.59
aLasso	0.91	0.93	1.19	1.25	1.02	1.03	0.83	0.73
Bridge	0.87	0.90	1.27	1.33	1.08	1.07	0.87	0.78
SCAD	0.85	0.87	1.30	1.34	0.98	1.03	0.77	0.61
MCP	1.03	1.07	1.33	1.38	1.26	1.32	1.06	0.94
SICA	0.82	0.88	1.20	1.27	1.13	1.12	0.72	0.56
Dimensionality reduction methods								
PCA	0.50	0.58	0.93	1.02	0.41	0.42	0.31	0.11
SPCA	0.45	0.59	1.02	1.14	0.13	0.12	−0.05	−0.20
PLS	1.17	1.23	1.28	1.25	0.71	0.65	0.31	0.17
SPLS	0.91	0.97	0.99	1.00	0.97	0.68	0.19	0.08
ICA	0.55	0.61	0.92	0.99	0.02	−0.25	−0.88	−1.11
RICA	0.59	0.65	0.95	1.07	−0.11	−0.12	−0.14	−0.23
Nonlinear machine learning methods								
RF	0.70	0.73	0.96	1.06	0.09	0.08	−0.32	−0.52
ERT	0.43	0.59	1.05	1.20	0.13	−0.12	−0.70	−0.88
GBM	0.65	0.69	0.86	0.95	−0.98	−0.99	−0.53	−0.33
RGBM	0.64	0.70	0.76	0.79	−1.00	−1.07	−0.88	−0.70
e-SVM	0.59	0.65	0.88	1.03	0.25	0.10	0.17	0.16
nu-SVM	0.54	0.66	0.81	0.96	−0.19	−0.19	−0.31	−0.26
MLP1	1.06	1.08	1.23	1.31	0.93	0.89	0.73	0.67
MLP2	0.59	0.64	0.95	1.03	0.65	0.51	0.04	−0.08
MLP3	0.47	0.52	0.68	0.79	−0.46	−0.40	0.13	0.17
Forecast combinations of machine learning models								
MC_{ML}	1.13	1.15	1.32	1.35	1.47	1.32	0.71	0.47
MDC_{ML}	1.03	1.12	1.18	1.16	1.16	1.10	0.71	0.47

(Continues)

TABLE 7 (Continued)

	A. Expansions				B. Recessions			
	Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)		Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)	
Model	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**
TMC_{ML}	1.12	1.17	1.32	1.38	<u>1.47</u>	1.35	0.77	0.49
$Rank_{ML}$	0.82	0.93	1.26	1.37	1.19	1.22	0.87	0.57
$CL2_{ML}$	0.98	1.00	1.24	1.33	1.18	1.29	0.83	0.58
$CL3_{ML}$	0.93	1.00	1.12	1.19	1.33	1.34	0.85	0.52
$DMSFE1_{ML}$	1.14	1.18	1.33	1.38	1.46	<u>1.36</u>	0.76	0.48
$DMSFE09_{ML}$	1.23	1.32	<u>1.40</u>	<u>1.45</u>	1.33	1.20	0.72	0.51

Note: This table reports the annualized Sharpe ratio of the mean–variance portfolios, with monthly rebalancing, for the out-of-sample period of January 2000 to December 2019, during business cycles. The mean–variance portfolios are based on the graphical lasso estimator. Panel A reports the performance during NBER expansions, while panel B reports the performance during NBER recessions. The performance is reported for different levels of risk aversion ($\gamma = 2, 10$), for unleveraged portfolios ($w_j \in [0,1]$) and portfolios with leverage ($w_j \in [0,1.5]$). The models with the highest Sharpe ratio are underlined.

FIGURE 3 Variable importance. Columns correspond to individual models and rows to the 20 most influential predictors for each asset class. The color of each model-predictor combination indicates the importance of the respective variable to the model, from most influential variable (blue) to least influential variables (white)

5.1 | Effects of transaction costs

The estimation of transaction costs is based on portfolio turnover. Given a transaction cost of c, the trading cost of the entire portfolio is $c\sum_{j=1}^{N}\left|w_{j,t+1}-w_{j,t}\right|$. The return of the portfolio after transaction costs is as follows:

$$r_{P,t+1}^{TC}=(1+r_{P,t+1})\left(1-c\sum_{j=1}^{N}\left|w_{j,t+1}-w_{j,t}\right|\right)-1. \tag{35}$$

We follow Olivares-Nadal and DeMiguel (2018) who show that incorporating a l_p transaction cost term in the mean–variance portfolio problem may help to reduce the impact of estimation error. Here we consider the case of proportional transaction costs and modify the mean–variance optimization problem by adding a l_1 transaction cost term, which is equivalent to assuming that transaction costs are proportional to the amount traded. The new constrained optimization problem becomes

$$\begin{aligned}&\underset{\mathbf{w}}{\operatorname{argmin}}\left[\gamma\mathbf{w}'\boldsymbol{\Sigma}\mathbf{w}-\mathbf{w}'\widehat{\mathbf{r}}+c\|\mathbf{w}-\mathbf{w}_0\|_1\right]\\ &\text{s.t.}\mathbf{w}'\mathbf{i}_N=h \ \text{ and } \ w_j\geq 0, \text{ with } \ j=1,...,N,\end{aligned} \tag{36}$$

where c is the transaction cost parameter and $\mathbf{w}_0$ are the weights of the portfolio from the previous period before rebalancing. For the first period of the expanding window the weights $\mathbf{w}_0$ are initialized based on the original mean–variance allocation. The transaction costs are set to $c=50$ bps for each asset. When $c=0$ the above optimization problem becomes equivalent to the one in 29.

Panel A of Table 8 reports the portfolio performance in terms of Sharpe ratio, for transaction costs of 50 bps using the penalized portfolio objective function with a covariance matrix estimated by the graphical lasso. When transaction costs are introduced, the portfolios of a conservative investor are affected more than those of an aggressive investor, with several models failing to outperform the EW portfolio and the number of models with Sharpe ratios lower than the EW and HA portfolios increasing.

More in detail, the Sharpe ratio for the HA portfolio of an aggressive (conservative) investor is 0.36 (0.57) and 0.42 (0.66) for unleveraged and leveraged positions, while the respective values for the KS portfolios range from 0.64 to 0.68. The Sharpe ratio for portfolios of an aggressive investor based on alternative predictive models is between −0.15 (MLP3) and 0.83 ($\text{DMSFE09}_{\text{ML}}$) for unleveraged portfolios and −0.24 (MLP3) to 0.82 ($\text{DMSFE09}_{\text{ML}}$) for a 50% leverage constraint. Turning to the results for the conservative investor, for $w_j\in[0,1]$, the Sharpe ratio is between −0.23 (MLP3) and 0.80 (Rank_{ML}) or from −0.25 (MLP3) to 0.83 (Rank_{ML}) when leverage is allowed. Overall, models based on machine learning forecast combinations yield the highest ratios, while those based on RICA and a neural network with three hidden layers generate the lowest Sharpe ratios. Turning to the results for average turnover (panel B), the added penalty to the mean–variance objective has the greatest effect on the HA and simple forecast combination portfolios, which produce lower turnover compared with the remaining strategies. The turnover for portfolios with $\gamma=2$ ranges from 15.30% (ERT) to 260.69% (Ridge), and for $\gamma=10$, the respective values are 11.01% (ERT) and 239.16% (GBM).

The cumulative returns of the EW, HA, and forecast combinations of machine learning models for an aggressive investor are plotted in the first part of Figure 4. The cumulative returns of all series fluctuate much more compared with the case without transaction costs (Figure 1); this volatility becomes more pronounced around the two crises periods. After the global financial crisis, the portfolios based on machine learning forecast combinations show significant gains over the two benchmarks; however, all models exhibit similar behavior over time with the discounted mean square forecast error scheme with $\psi=0.9$ having the highest end-of-period value. The cumulative return graphs of the eight forecast combinations along with the two benchmarks for portfolios with $\gamma=10$ and transaction costs are presented in the second part of Figure 4. Overall, the difference in behavior of the cumulative return series across models is more pronounced, compared with when $\gamma=2$. The portfolios based on cluster combination forecasts outperform all other models (especially after the dot-com bubble) and achieve the highest end-of-period values. The performance of the remaining forecast combinations is superior to the EW portfolio after the global financial crisis.

TABLE 8 Portfolio performance after transaction costs of 50 bps

	A. Sharpe ratio				B. Average turnover			
	Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)		Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)	
Model	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**
EW	0.78	0.78	0.78	0.78	0.00	0.00	0.00	0.00
HA	0.36	0.42	0.57	0.66	0.21	0.27	0.53	0.75
KS	0.64	0.68	0.67	0.65	172.65	259.98	165.22	236.91
Forecast combinations of bivariate prediction models								
MC	0.32	0.41	0.60	0.65	0.40	0.62	0.71	1.22
MDC	0.34	0.41	0.57	0.63	0.26	0.32	0.60	0.96
TMC	0.33	0.42	0.60	0.64	0.45	0.60	0.71	1.17
Rank	0.50	0.50	0.65	0.72	13.16	18.72	10.32	13.28
CL2	0.25	0.30	0.57	0.64	1.09	1.87	1.48	2.24
CL3	0.36	0.41	0.59	0.65	2.23	3.85	2.64	3.87
DMSFE1	0.32	0.41	0.60	0.65	0.41	0.59	0.74	1.25
DMSFE09	0.31	0.41	0.60	0.66	0.34	0.49	0.72	1.24
Shrinkage methods								
Ridge	0.64	0.64	0.66	0.65	174.06	260.69	162.27	234.07
Lasso	0.60	0.56	0.56	0.50	163.57	244.23	151.78	216.24
EN	0.58	0.55	0.56	0.50	163.93	246.26	152.48	217.76
aLasso	0.57	0.54	0.55	0.52	165.68	247.76	152.19	215.10
Bridge	0.50	0.48	0.58	0.57	170.09	253.71	160.23	232.95
SCAD	0.51	0.45	0.62	0.59	171.32	256.09	160.13	230.69
MCP	0.60	0.62	0.65	0.61	173.00	260.04	163.64	236.80
SICA	0.48	0.47	0.52	0.50	167.75	251.42	157.10	225.52
Dimensionality reduction methods								
PCA	0.16	0.18	0.45	0.51	64.89	91.64	47.84	64.66
SPCA	0.21	0.30	0.57	0.58	80.08	110.94	54.58	70.95
PLS	0.62	0.65	0.50	0.40	159.27	236.95	145.62	203.96
SPLS	0.54	0.53	0.32	0.29	157.38	233.93	141.72	197.18
ICA	0.30	0.24	0.24	0.29	79.86	111.50	57.27	75.14
RICA	−0.02	0.07	0.29	0.37	92.57	131.23	69.56	94.96
Nonlinear machine learning methods								
RF	0.27	0.28	0.38	0.43	91.11	129.81	63.41	81.11
ERT	0.42	0.45	0.64	0.71	15.30	21.21	11.01	13.94
GBM	0.07	0.06	0.10	0.11	169.70	253.09	162.74	239.16
RGBM	0.02	0.02	0.05	0.04	167.95	250.45	157.67	226.08
e-SVM	0.27	0.27	0.57	0.70	28.85	44.20	28.59	37.44
nu-SVM	0.11	0.19	0.38	0.47	27.61	38.54	21.89	29.15
MLP1	0.62	0.63	0.53	0.51	151.05	229.23	150.86	220.42
MLP2	0.14	0.11	0.05	0.00	160.23	238.90	149.26	215.41
MLP3	−0.15	−0.24	−0.23	−0.25	167.83	246.51	156.72	228.84
Forecast combinations of machine learning models								
MC_{ML}	0.74	0.73	0.59	0.52	146.90	213.65	124.10	168.97

TABLE 8 (Continued)

	A. Sharpe ratio				B. Average turnover			
	Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)		Aggressive ($\gamma = 2$)		Conservative ($\gamma = 10$)	
Model	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**	**[0, 1]**	**[0, 1.5]**
MDC_{ML}	0.71	0.68	0.60	0.50	150.71	220.99	129.56	174.37
TMC_{ML}	0.75	0.76	0.62	0.56	146.09	214.09	122.09	166.32
$Rank_{ML}$	0.58	0.68	0.80	0.83	119.23	170.80	85.02	107.90
$CL2_{ML}$	0.59	0.62	0.78	0.83	124.73	181.71	94.61	124.17
$CL3_{ML}$	0.62	0.66	0.73	0.74	129.24	186.32	95.54	125.94
$DMSFE1_{ML}$	0.74	0.77	0.64	0.58	141.85	208.46	118.85	159.95
$DMSFE09_{ML}$	0.83	0.82	0.72	0.68	142.91	209.06	117.23	159.25

Note: This table reports the annualized Sharpe ratio and average turnover of the mean–variance portfolios, with monthly rebalancing, for the out-of-sample period of January 2000 to December 2019. The portfolios are based on the graphical lasso estimator and on the penalized mean–variance objective function with transaction costs of 50 bps. Panel A reports the performance based on the Sharpe ratio, panel B reports the results based on average turnover. The performance is reported for different levels of risk aversion ($\gamma = 2, 10$), for unleveraged portfolios ($w_j \in [0, 1]$) and portfolios with leverage ($w_j \in [0, 1.5]$). The models with the highest Sharpe ratio and turnover are underlined.

5.2 | Alternative rebalancing frequencies

In the analysis detailed above, we demonstrate that for monthly rebalancing, portfolios based on machine learning forecast combinations generate the highest performance. In this section, we examine the effects to portfolio performance when the rebalancing frequency of the portfolios is reduced from monthly to quarterly or annual. We report the results for the two alternative rebalancing frequencies, for the average return and the Sharpe ratio in Tables S6 and S7, respectively. Overall, the results indicate that machine learning models favor higher rebalancing frequencies. Reducing the frequency with which a portfolio is rebalanced leads to considerably lower performance for all models, in terms of both measures. Specifically, the results based on average return show that the best performing portfolio in the case of quarterly rebalancing is based on a shallow neural network. However, when rebalancing frequency is reduced to annual, the best performing portfolios become sparse PCA and a neural network with two hidden layers for the aggressive and conservative investors, respectively. On the contrary, the results based on the Sharpe ratio show that all alternative models fail to outperform the equally weighted allocation.

5.3 | Time-varying parameter of risk-aversion

In the previous section, we used values for the parameter of risk aversion that remain fixed across the portfolio formation period. Here, we employ a new measure of time-varying risk aversion, named RAbex, proposed by Bekaert et al. (2021), which is derived from observable financial variables, such as earnings yield, corporate return spread (Baa-Aaa), term spread (10 yr-3mth), equity return realized variance, corporate bond return realized variance, and equity risk-neutral variance.[15] The results for the average return, Sharpe ratio, and turnover are reported in Table S8. In terms of average return and Sharpe ratio, the best performing models are those based on forecast combinations of machine learning models, followed by the KS model, a shallow neural network (MLP1), and shrinkage methods. The best performing forecast combination is $DMSFE09_{ML}$ with AR (SR) 26.56% (1.30) for unleveraged positions and 37.04% (1.33) when leverage is allowed. The KS model, the neural network with three hidden layers, and shrinkage methods exhibit the highest average turnover. Overall, the performance of portfolios with time-varying risk-aversion is between that of the portfolios with $\gamma = 2$ and $\gamma = 10$ from the previous section.

[15]We would like to thank an anonymous referee for their suggestion.

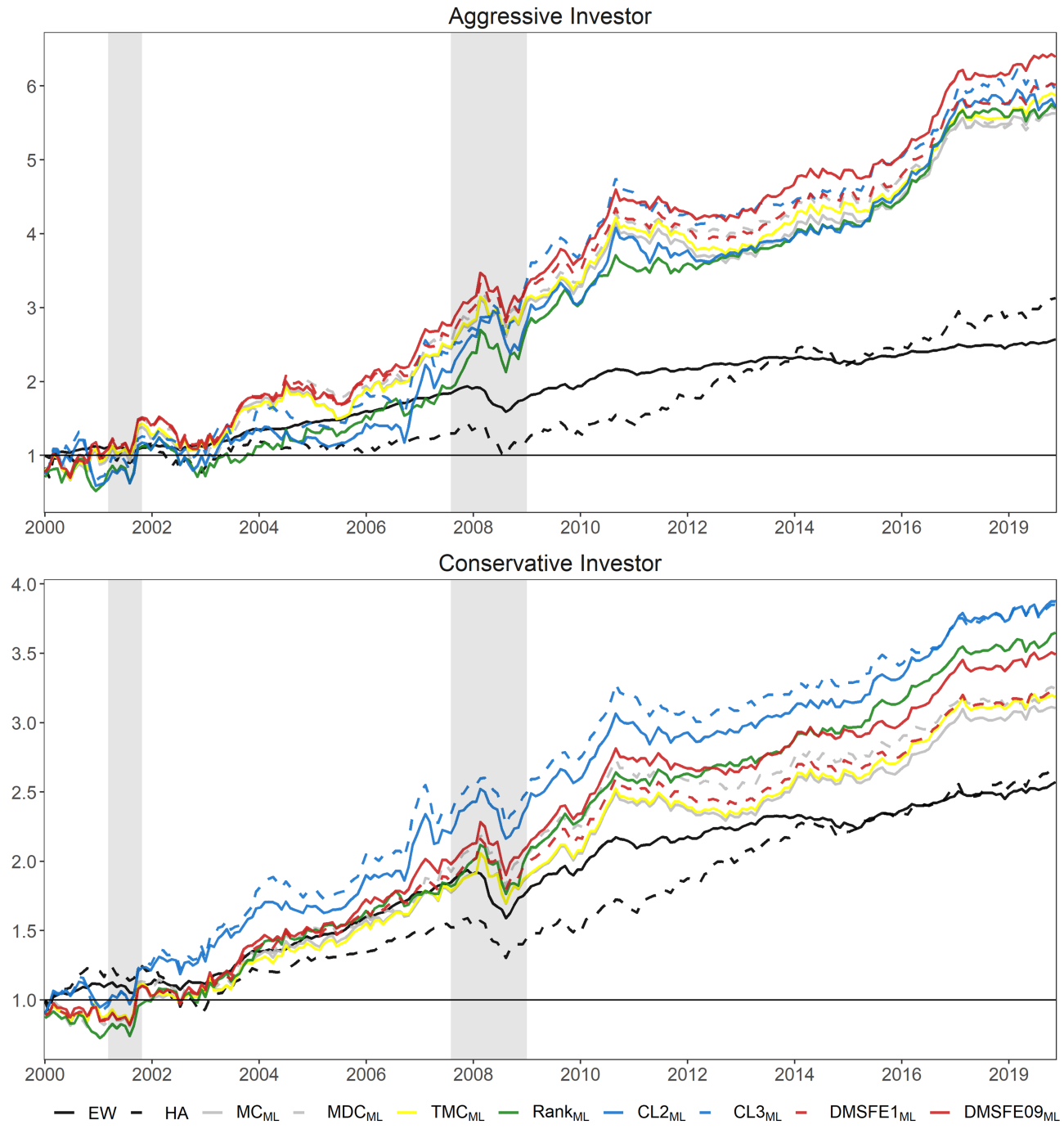

FIGURE 4 Cumulative returns for portfolios with transaction costs of 50 bps. The mean–variance portfolios are based on the graphical lasso estimator and on the penalized mean–variance objective function with transaction costs of 50 bps, with combination of weight constraints $w_j \in [0, 1.5]$, for the aggressive investor ($\gamma = 2$) and $w_j \in [0, 1]$, for the conservative investor ($\gamma = 10$). The shaded regions depict the NBER-dated recessions and expansions

5.4 | Shrinkage covariance estimator

As an alternative to the graphical lasso used in the previous section, we consider portfolios based on the linear shrinkage estimator of the covariance matrix proposed by Ledoit and Wolf (2004). Specifically, the sample covariance matrix is shrunk towards a one-parameter matrix, where all the variances are the same and all covariances are zero. The results, presented in Table S9, remain qualitatively similar to the graphical lasso estimator. The

quantitative difference in performance between the two estimators is small and any small benefits from using the shrinkage over the sparse estimator are mixed and depend on the model used to generate the return forecasts.

5.5 | Power-series and interactions

Our findings so far have shown that shrinkage methods tend to outperform the majority of the nonlinear methods, with the exception of a shallow neural network. In this respect, we also examine how the performance of the portfolios is affected when the predictor set of the shrinkage methods is replaced with one that includes higher polynomials and two-way interactions of the original variables. The results are reported in Table S10. Our findings suggest that the performance of the portfolios is significantly reduced across combinations of weight constraints and values for the parameter of risk aversion. Moreover, shrinkage methods with non-convex penalties are more affected than those based on convex penalties.

5.6 | Alternative loss function

The presence of extreme observations in financial and economic variables can undermine the stability of the models due to the emphasis placed by least squares on large errors. In some cases, it is possible to improve model performance by using an alternative to the least squares objective function in Equation 3. Similarly to Gu et al. (2020), we consider the Huber loss function which is a hybrid of the least squares loss for relatively small errors and the absolute loss for relatively large errors. Specifically, we replace the least square loss for the shrinkage methods with convex penalties (Ridge, Lasso, and Elastic Net) and boosting ensembles (GBM and RGBM). The optimal parameter of the Huber loss function, which interpolates between the mean and the median regression, is chosen using the validation sample approach described in Section 3.2. The results are presented in Table S11. Our findings suggest that shrinkage methods tend to generate higher average returns and Sharpe ratios when using the least square loss. However, the Huber loss function provides small economic benefits to portfolios based on forecasts from boosting methods for the case of a conservative investor ($\gamma = 10$).

6 | CONCLUSION

This study sets out to explore whether return forecasts generated by machine learning add value to portfolios consisting of stocks, bonds and commodities. The portfolios are constructed based on the proposed models and their performance is compared to that of the equal-weighted portfolio and a mean–variance portfolio based on the historical average. The majority of the portfolios utilizing return forecasts outperform the $1/N$ and historical average benchmarks in terms of average returns and Sharpe ratio, with forecast combinations of machine learning models yielding the highest performance. Portfolios with leverage generate higher average return than the unleveraged allocations. There are no major changes when comparing either performance measure across different specifications of the covariance matrix. When transaction costs are introduced and a mean–variance objective function with a transaction cost penalty is utilized, the results for the monthly-rebalanced portfolios continue to favor forecast combinations of machine learning methods even though the performance of the portfolios is considerably reduced.

OPEN RESEARCH BADGES

This article has been awarded Open Data Badge for making publicly available the digitally-shareable data necessary to reproduce the reported results. Data is available at http://qed.econ.queensu.ca/jae/datasets/kynigakis001/.

REFERENCES

Aiolfi, M., & Timmermann, A. (2006). Persistence in forecasting performance and conditional combination strategies. *Journal of Econometrics*, *135*, 31–53. https://doi.org/10.1016/j.jeconom.2005.07.015

Babii, A., Ghysels, E., & Striaukas, J. (2019). Estimation and HAC-based inference for machine learning time series regressions. Available at SSRN: https://ssrn.com/abstract=3503191

Bai, J., & Ng, S. (2008). Forecasting economic time series using targeted predictors. *Journal of Econometrics*, *146*, 304–317. https://doi.org/10.1016/j.jeconom.2008.08.010

Bates, J. M., & Granger, C. W. (1969). The combination of forecasts. *Journal of the Operational Research Society*, *20*, 451–468. https://doi.org/10.1057/jors.1969.103

Bekaert, G., Engstrom, E. C., & Xu, N. R. (2021). The time variation in risk appetite and uncertainty. *Management Science*. https://doi.org/10.1287/mnsc.2021.4068

Bianchi, D., Büchner, M., & Tamoni, A. (2020). Bond risk premia with machine learning. *The Review of Financial Studies*, *34*, 1046–1089. https://doi.org/10.1093/rfs/hhaa062

Bianchi, D., & Guidolin, M. (2014). Can long-run dynamic optimal strategies outperform fixed-mix portfolios? Evidence from multiple data sets. *European Journal of Operational Research*, *236*, 160–176. https://doi.org/10.1016/j.ejor.2014.01.030

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140. https://doi.org/10.1007/BF00058655

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Wadsworth Inc.

Callot, L., Caner, M., Önder, A. O., & Ulaşan, E. (2019). A nodewise regression approach to estimating large portfolios. *Journal of Business & Economic Statistics*, *39*, 1–12. https://doi.org/10.1080/07350015.2019.1683018

Callot, L. A., Kock, A. B., & Medeiros, M. C. (2017). Modeling and forecasting large, realized covariance matrices and portfolio choice. *Journal of Applied Econometrics*, *32*, 140–158. https://doi.org/10.1002/jae.2512

Campbell, J. Y., & Thompson, S. B. (2007). Predicting excess stock returns out of sample: Can anything beat the historical average? *The Review of Financial Studies*, *21*, 1509–1531. https://doi.org/10.1093/rfs/hhm055

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

Carrasco, M., & Rossi, B. (2016). In-sample inference and forecasting in misspecified factor models. *Journal of Business & Economic Statistics*, *34*, 313–338. https://doi.org/10.1080/07350015.2016.1186029

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.

Chen, Y., Rogoff, K. S., & Rossi, B. (2010). Can exchange rates forecast commodity prices? *The Quarterly Journal of Economics*, *125*, 1145–1194. https://doi.org/10.1162/qjec.2010.125.3.1145

Chun, H., & Keleş, S. (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, *72*, 3–25. https://doi.org/10.1111/j.1467-9868.2009.00723.x

Cochrane, J. H., & Piazzesi, M. (2005). Bond risk premia. *American Economic Review*, *95*, 138–160. https://doi.org/10.1257/0002828053828581

Comon, P. (1994). Independent component analysis, a new concept? *Signal Processing*, *36*, 287–314. https://doi.org/10.1016/0165-1684(94)90029-9

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, *2*, 303–314. https://doi.org/10.1007/BF02551274

De Jong, S. (1993). SIMPLS: An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, *18*, 251–263. https://doi.org/10.1016/0169-7439(93)85002-X

De Mol, C., Giannone, D., & Reichlin, L. (2008). Forecasting using a large number of predictors: Is Bayesian shrinkage a valid alternative to principal components? *Journal of Econometrics*, *146*, 318–328. https://doi.org/10.1016/j.jeconom.2008.08.011

DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *The Review of Financial Studies*, *22*, 1915–1953. https://doi.org/10.1093/rfs/hhm075

D'Hondt, C., De Winne, R., Ghysels, E., & Raymond, S. (2020). Artificial intelligence alter egos: Who might benefit from robo-investing? *Journal of Empirical Finance*, *59*, 278–299. https://doi.org/10.1016/j.jempfin.2020.10.002

Dietterich, T. G. (2000). Ensemble methods in machine learning. International Workshop on Multiple Classifier Systems. 1–15. https://doi.org/10.1007/3-540-45014-9_1

Duchin, R., & Levy, H. (2009). Markowitz versus the talmudic portfolio diversification strategies. *Journal of Portfolio Management*, *35*, 71–74. https://doi.org/10.3905/JPM.2009.35.2.071

Engle, R. (2002). Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *Journal of Business & Economic Statistics*, *20*, 339–350. https://doi.org/10.1198/073500102288618487

Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, *33*, 3–56. https://doi.org/10.1016/0304-405X(93)90023-5

Fama, E. F., & French, K. R. (2016). Dissecting anomalies with a five-factor model. *The Review of Financial Studies*, *29*, 69–103. https://doi.org/10.1093/rfs/hhv043

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, *96*, 1348–1360. https://doi.org/10.1198/016214501753382273

Frank, L. E., & Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, *35*, 109–135. https://doi.org/10.1080/00401706.1993.10485033

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and Computation*, *121*, 256–285. https://doi.org/10.1006/inco.1995.1136

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, *9*, 432–441. https://doi.org/10.1093/biostatistics/kxm045

Friedman, J., Trevor, H., & Robert, T. (2009). *The elements of statistical learning: Data mining, inference, and prediction, Springer Series in Statistics* (2nd ed.). New York, NY, USA. https://doi.org/10.1007/978-0-387-84858-7

Friedman, J. H. (2001). greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*, 1189–1232. https://doi.org/10.1214/aos/1013203451

Friedman, J. H. (2012). Fast sparse regression and classification. *International Journal of Forecasting*, *28*, 722–738. https://doi.org/10.1016/j.ijforecast.2012.05.001

Gao, X., & Nardari, F. (2018). Do commodities add economic value in asset allocation? New evidence from time-varying moments. *Journal of Financial and Quantitative Analysis*, *53*, 365–393. https://doi.org/10.1017/S002210901700103X

Gargano, A., Pettenuzzo, D., & Timmermann, A. (2017). Bond return predictability: Economic value and links to the macroeconomy. *Management Science*, *65*, 508–540. https://doi.org/10.1287/mnsc.2017.2829

Gargano, A., & Timmermann, A. (2014). Forecasting commodity price indexes using macroeconomic and financial predictors. *International Journal of Forecasting*, *30*, 825–843. https://doi.org/10.1016/j.ijforecast.2013.09.003

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*, 3–42. https://doi.org/10.1007/s10994-006-6226-1

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. 315–323.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Gu, S., Kelly, B. T., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, *33*, 2223–2273. https://doi.org/10.1093/rfs/hhaa009

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*, 993–1001. https://doi.org/10.1109/34.58871

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Prediction, inference and data mining*. Springer-Verlag. https://doi.org/10.1007/978-0-387-84858-7

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*, 55–67. https://doi.org/10.1080/00401706.1970.10488634

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*, 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks*, *13*, 411–430. https://doi.org/10.1016/S0893-6080(00)00026-5

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning, *37*, 448–456.

Jagannathan, R., & Ma, T. (2003). Risk reduction in large portfolios: Why imposing the wrong constraints helps. *The Journal of Finance*, *58*, 1651–1683. https://doi.org/10.1111/1540-6261.00580

Jaggi, M. (2014). *An equivalence between the lasso and support vector machines*. Chapman and Hall/CRC.

Jurado, K., Ludvigson, S. C., & Ng, S. (2015). Measuring uncertainty. *American Economic Review*, *105*, 1177–1216. https://doi.org/10.1257/aer.20131193

Kelly, B., & Pruitt, S. (2015). The three-pass regression filter: A new approach to forecasting using many predictors. *Journal of Econometrics*, *186*, 294–316. https://doi.org/10.1016/j.jeconom.2015.02.011

Kelly, B. T., Pruitt, S., & Su, Y. (2019). Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics*, *134*, 501–524. https://doi.org/10.1016/j.jfineco.2019.05.001

Kilian, L. (2009). Not all oil price shocks are alike: Disentangling demand and supply shocks in the crude oil market. *American Economic Review*, *99*, 1053–1069. https://doi.org/10.1257/aer.99.3.1053

Kim, H. H., & Swanson, N. R. (2014). Forecasting financial and macroeconomic variables using data reduction methods: New empirical evidence. *Journal of Econometrics*, *178*, 352–367. https://doi.org/10.1016/j.jeconom.2013.08.033

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. International Conference on Learning Representations.

Kirby, C., & Ostdiek, B. (2012). It's all in the timing: Simple active portfolio strategies that outperform naive diversification. *Journal of Financial and Quantitative Analysis*, *47*, 437–467. https://doi.org/10.1017/S0022109012000117

Kotchoni, R., Leroux, M., & Stevanovic, D. (2019). Macroeconomic forecast accuracy in a data-rich environment. *Journal of Applied Econometrics*, *34*, 1050–1072. https://doi.org/10.1002/jae.2725

Kozak, S. (2019). Kernel trick for the cross-section. Available at SSRN: https://ssrn.com/abstract=3307895

Kritzman, M., Page, S., & Turkington, D. (2010). In defense of optimization: The fallacy of 1/N. *Financial Analysts Journal*, *66*, 31–39. https://doi.org/10.2469/faj.v66.n2.6

Le, Q. V., Karpenko, A., Ngiam, J., & Ng, A. Y. (2011). ICA with reconstruction cost for efficient overcomplete feature learning. *Advances in Neural Information Processing Systems*, *24*, 1017–1025.

Ledoit, O., & Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, *88*, 365–411. https://doi.org/10.1016/S0047-259X(03)00096-4

Lima, L. R., & Meng, F. (2017). Out-of-sample return predictability: A quantile combination approach. *Journal of Applied Econometrics*, *32*, 877–895. https://doi.org/10.1002/jae.2549

Lin, H., Wu, C., & Zhou, G. (2017). Forecasting corporate bond returns with a large set of predictors: An iterated combination approach. *Management Science*, *64*, 4218–4238. https://doi.org/10.1287/mnsc.2017.2734

Ludvigson, S. C., Ma, S., & Ng, S. (2015). Uncertainty and business cycles: Exogenous impulse or endogenous response? NBER Working Paper No. w21803. Available at SSRN: https://ssrn.com/abstract=2703208

Ludvigson, S. C., & Ng, S. (2009). Macro factors in bond risk premia. *The Review of Financial Studies*, *22*, 5027–5067. https://doi.org/10.1093/rfs/hhp081

Lv, J., & Fan, Y. (2009). A unified approach to model selection and sparse recovery using regularized least squares. *The Annals of Statistics*, *37*, 3498–3528. https://doi.org/10.1214/09-AOS683

Markowitz, H. M. (1952). Portfolio selection. *The Journal of Finance*, *7*, 77–91. https://doi.org/10.1111/j.1540-6261.1952.tb01525.x

Masters, T. (1993). *Practical neural network recipes in C*. Morgan Kaufmann. https://doi.org/10.1016/B978-0-08-051433-8.50017-3

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. Proceedings of the 27th International Conference on Machine Learning. 807–814.

Neely, C. J., Rapach, D. E., Tu, J., & Zhou, G. (2014). Forecasting the equity risk premium: The role of technical indicators. *Management Science*, *60*, 1772–1791. https://doi.org/10.1287/mnsc.2013.1838

Olivares-Nadal, A. V., & DeMiguel, V. (2018). A robust perspective on transaction costs in portfolio optimization. *Operations Research*, *66*, 733–739. https://doi.org/10.1287/opre.2017.1699

Pástor, L., & Stambaugh, R. F. (2003). Liquidity risk and expected stock returns. *Journal of Political Economy*, *111*, 642–685. https://doi.org/10.1086/374184

Politis, D. N., & Romano, J. P. (1992). A general resampling scheme for triangular arrays of α-mixing random variables with application to the problem of spectral density estimation. *The Annals of Statistics*, *20*, 1985–2007. https://doi.org/10.1214/aos/1176348899

Rapach, D. E., Strauss, J. K., Tu, J., & Zhou, G. (2019). Industry return predictability: A machine learning approach. *The Journal of Financial Data Science*, *1*, 9–28. https://doi.org/10.3905/jfds.2019.1.3.009

Rapach, D. E., Strauss, J. K., & Zhou, G. (2010). Out-of-sample equity premium prediction: Combination forecasts and links to the real economy. *The Review of Financial Studies*, *23*, 821–862. https://doi.org/10.1093/rfs/hhp063

Rapach, D. E., Strauss, J. K., & Zhou, G. (2013). International stock return predictability: What is the role of the United States? *The Journal of Finance*, *68*, 1633–1662. https://doi.org/10.1111/jofi.12041

Rapach, D. E., Wohar, M. E., & Rangvid, J. (2005). Macro variables and international stock return predictability. *International Journal of Forecasting*, *21*, 137–166. https://doi.org/10.1016/j.ijforecast.2004.05.004

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*, 197–227. https://doi.org/10.1007/BF00116037

Schölkopf, B., Bartlett, P., Smola, A., & Williamson, R. (1999). Shrinking the tube: A new support vector regression algorithm. In *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II* (pp. 330–336). Cambridge, MA, USA: MIT Press.

Schölkopf, B., Smola, A. J., Williamson, R. C., & Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, *12*, 1207–1245. https://doi.org/10.1162/089976600300015565

Stock, J. H., & Watson, M. W. (2004). Combination forecasts of output growth in a seven-country data set. *Journal of Forecasting*, *23*, 405–430. https://doi.org/10.1002/for.928

Stock, J. H., & Watson, M. W. (2012). Generalized shrinkage methods for forecasting using many predictors. *Journal of Business & Economic Statistics*, *30*, 481–493. https://doi.org/10.1080/07350015.2012.715956

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B: Methodological*, *58*, 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Timmermann, A. (2006). Forecast combinations. *Handbook of Economic Forecasting*, *1*, 135–196. https://doi.org/10.1016/S1574-0706(05)01004-9

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Wiley. https://doi.org/10.1007/978-1-4757-2440-0

Welch, I., & Goyal, A. (2008). A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies*, *21*, 1455–1508. https://doi.org/10.1093/rfs/hhm014

Wold, H. (1966). Estimation of principal components and related models by iterative least squares. In P. R. Krishnaiah (Ed.), *Multivariate analysis* (pp. 391–420). New York: Academic Press.

Zhang, C. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, *38*, 894–942. https://doi.org/10.1214/09-AOS729

Zhou, Q., Chen, W., Song, S., Gardner, J., Weinberger, K., & Chen, Y. (2015). A reduction of the elastic net to support vector machines with an application to GPU computing. Proceedings of the AAAI conference on artificial intelligence, *29*.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, *101*, 1418–1429. https://doi.org/10.1198/016214506000000735

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, *67*, 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

Zou, H., Hastie, T., & Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, *15*, 265–286. https://doi.org/10.1198/106186006X113430

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Kynigakis, I., & Panopoulou, E. (2022). Does model complexity add value to asset allocation? Evidence from machine learning forecasting models. *Journal of Applied Econometrics*, *37*(3), 603–639. https://doi.org/10.1002/jae.2885