5-2020

Forecast combinations in machine learning

Yue QIU

Tian XIE

Jun YU
Singapore Management University, yujun@smu.edu.sg

SMU ECONOMICS &
STATISTICS

Forecast Combinations in Machine Learning

Yue Qiu, Tian Xie, Jun Yu

May 2020

Paper No. 13-2020

ANY OPINION EXPRESSED ARE THOSE OF THE AUTHOR(S) AND NOT NECESSARILY THOSE OF
THE SCHOOL OF ECONOMICS, SMU

Forecast combinations in machine learning[*]

Yue Qiu[†] Tian Xie[‡] Jun Yu[§]

May 11, 2020

Abstract

This paper introduces novel methods to combine forecasts made by machine learning techniques. Machine learning methods have found many successful applications in predicting the response variable. However, they ignore model uncertainty when the relationship between the response variable and the predictors is nonlinear. To further improve the forecasting performance, we propose a general framework to combine multiple forecasts from machine learning techniques. Simulation studies show that the proposed machine-learning-based forecast combinations work well. In empirical applications to forecast key macroeconomic and financial variables, we find that the proposed methods can produce more accurate forecasts than individual machine learning techniques and the simple average method, later of which is known as hard to beat in the literature.

JEL classification: C52, C53
Keywords: Model uncertainty, Machine learning, Nonlinearity, Forecast combinations

1 Introduction

Making reliable forecasts based on data is important in policy-making, business decisions and many other activities. Machine learning is an automated way of identifying patterns

[*] We would like to thank Allan Timmermann, Dacheng Xiu, Xinyu Zhang, and Steven F. Lehrer for comments and suggestions that improved this paper. Xie's research is supported by the Natural Science Foundation of China (71701175), the Chinese Ministry of Education Project of Humanities and Social Sciences (17YJC790174), and the Fundamental Research Funds for the Central Universities. Yu acknowledges financial support from the Lee Foundation. The usual caveat applies.

[†] Finance School, Shanghai University of International Business and Economics, Shanghai, China.

[‡] College of Business, Shanghai University of Finance and Economics, Shanghai, China.

[§] School of Economics and Lee Kong Chian School of Business, Singapore Management University, Singapore.

in data and using them to make predictions. The success of machine learning in making predictions relies on its ability to detect a complex structure that may not be realizable analytically and also on its ability to reduce dimensionality when sparsity exists. It is essential to note that machine learning does not try to locate the true data generating process (DGP). Instead, it tries to find functions that can work well out-of-sample. Recent application of machine learning techniques in economics and finance can be found in Gu, Kelly, and Xiu (2020), Feng and He (2019) and Coulombe, Leroux, Stevanovic, and Surprenant (2020). Recent surveys on forecasting using machine learning techniques can be found in Mullainathan and Spiess (2017) and Xie, Yu, and Zeng (2020).

On the other hand, forecasting based on statistical modeling has a long history; see Elliott and Timmermann (2016) and Diebold (2017) for the textbook treatments of the subject. Typically, a statistical model is assumed to be the underlying DGP. Competing model specifications are used to approximate the DGP and to generate forecasts. By giving up the assumption that one can successfully specify the DGP and hence acknowledge model uncertainty, many researchers, pioneered by Barnard (1963), Reid (1968), and Bates and Granger (1969), have found evidence of great value in combining multiple forecasts from a set of competing models. That is, averaging forecasts from multiple models leads to higher accuracy than employing the forecasts from individual models. To a certain degree, the preeminence in forecast combinations suggests “all models are wrong, but some are useful”, as Box (1976) famously claimed.

Since then, considerable efforts have been made in the literature to investigate various issues concerning forecast combinations, including the choice of competing models, the choice of weights, how to estimate weights, whether the simple equal weights should be used. Methods and applications have appeared both in the Bayesian paradigm and in the frequentist paradigm. Excellent reviews of forecast combination techniques can be found in Clemen (1989), Hoeting, Madigan, Raftery, and Volinsky (1999), Timmermann (2006), Elliott and Timmermann (2016). Many successful applications of forecast combinations to economics and finance have been found in the literature; see for instance, Rapach, Strauss, and Zhou (2009), Elliott, Gargano, and Timmermann (2013), and Genre, Kenny, Meyler, and Timmermann (2013). In the literature, forecast combinations are almost always based on a set of conventional statistical models. For example, in Rapach et al. (2009) all possible univariate linear regression models were used. Elliott et al. (2013) employ complete subset regressions that contain all possible linear regression models with a fixed number of predictors.

In this paper, we synthesize the forecast combination literature with the machine learning literature. In particular, we use machine learning techniques to form a set of competing strategies so as to generate individual forecasts. We then build a weighted average of the machine learning forecasts to predict the response variable. Our methods can be regarded as a new ensemble learning approach. It is general as it can be implemented on many machine learning techniques. When the predicted response variable can be expressed as a weighted average of the historical response variable, a condition met by many popular machine learning techniques, we propose to choose weights that minimize the Mallows-type criteria. We demonstrate the superiority of our proposed machine-learning-based combinations in Monte Carlo simulations.

In the empirical exercise, we first contrast a list of 31 forecasting strategies including our proposed methods in forecasting the inflation rates, the GDP growth rates, and the unemployment rates in the euro area. The recent literature on macroeconomic forecasts pays increasing attention to the significance of model uncertainty as well as to machine learning, separately but not jointly. Wieland, Cwik, Müller, Schmidt, and Wolters (2012) discovered considerable model uncertainty in macroeconomic modeling after the global financial crisis. They demonstrated the merits of adopting multiple modeling approaches as opposed to relying on one single model. Coulombe et al. (2020) studied the benefits of adopting machine learning methods over standard macroeconometric methods in macroeconomic forecasting. Our simulation and empirical results uncover that many machine learning methods can beat linear econometric methods in out-of-sample analyses but individual machine learning methods may not outperform the combined forecasts based on linear econometric methods. More importantly, we find that combining forecasts based on machine learning methods significantly improves on many rival strategies, including linear econometric methods, combining forecast based on linear econometric methods, and individual machine learning methods. We also apply our methods to forecast the 3-month Treasury bill rate as an additional financial application. The results keep demonstrating the advantage of the proposed forecast combination machine learning methods. In general, our empirical results provide an answer to the question posed in the title of Genre et al. (2013) – can anything beat the simple average? We answer that our proposed methods beat the simple average and do so by a wide margin.

The rest of the paper is organized as follows. Section 2 reviews existing methods, including conventional forecasting methods based on statistical models, forecast combinations applied to conventional models, and some well-known machine learning methods. Section 3 introduces our new combination machine learning methods, including the simple averaging machine learning method and the Mallows-type averaging machine learning method. We check the performance of the proposed methods using simulated data in Section 4. Section 5 presents two empirical applications to forecast three major macroeconomic variables in the euro zone and a key financial variable in the US. Section 6 concludes. The appendix provides additional details on derivation of a projection matrix for the least squares (LS) support vector regression (LSSVR), the data polishing procedure for the empirical study and the construction of the candidate model set. It also reports the empirical results under a tainted candidate model set and under alternative values of tuning parameters. An online appendix contains a review of penalized regression methods, more details on tree-type machine learning methods, a detailed description of the estimation procedure of LSSVR. It also presents a complete set of outcomes for verifying nonlinearity between the response variable and the 30 predictors. Supplementary results in the empirical study are also reported in the online appendix.

2 A review of existing methods

Let x_{it} be a set of p predictors (or explanatory variables) for $i = 1, ..., p$ and $t = 1, ..., T$. Let y_t be a univariate response variable. Consider a data sample of $\{y_t, \boldsymbol{X}_t\}_{t=1}^{T}$, where $\boldsymbol{X}_t = [1, x_{1t}, ..., x_{pt}]^\top$. In the literature on predictive regressions, the response variable can be the return rate r_{t+1} at time $t+1$. In this case, the predictors $\{x_{it}\}_{t=1}^{T}$, such as the dividend-price ratio and the earning-price ratio, may be used to obtain one-step-ahead forecast of y_{T+1}. In the survey-based forecast, $\{x_{it}\}$ is the forecast of the i^{th} forecaster surveyed. Before we introduce the new methods, we briefly review some existing forecasting methods.

2.1 Least squares and penalized regressions

When the relationship between y_t and $\boldsymbol{X}_t$ is linear, that is

$$y_t = \boldsymbol{X}_t^\top \boldsymbol{\beta} + \varepsilon_t, \tag{1}$$

assuming $\boldsymbol{X}_{T+h}$ is known at time t. The h-period-ahead forecast of y_{T+h}, denoted by $\hat{y}_{T+h}$, can be written as

$$\hat{y}_{T+h} = \hat{\beta}_0 + \sum_{i=1}^{p} \hat{\beta}_i x_{i,T+h} = \boldsymbol{X}_{T+h}^\top \hat{\boldsymbol{\beta}}, \tag{2}$$

where $\hat{\boldsymbol{\beta}} = [\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p]^\top$ is an estimate of $\boldsymbol{\beta}$. If $\boldsymbol{\beta}$ is estimated by LS from the data, then $\hat{\boldsymbol{\beta}}^{\mathrm{LS}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$, where $\boldsymbol{X} = [\boldsymbol{X}_1, ..., \boldsymbol{X}_T]^\top$ and $\boldsymbol{y} = [y_1, ..., y_T]^\top$.

When the number of predictors p is large and a significant subset of predictors is not that useful in predicting the response variable, the LS method does not perform well out-of-sample. In this case, a penalized LS regression may be employed to select predictors so as to improve predictive performance. The penalized LS regression belongs to a family of methods, including the ridge regression, the least absolute shrinkage selective operator (LASSO), and the elastic net. A detailed description of the above three methods can be found in Section 1.1 of the online appendix.

2.2 Forecast combinations

Forecast combinations can be cast as a model averaging problem where the quantity of interest is the out-of-sample value of a response variable. Model averaging techniques are designed to combat model uncertainty by obtaining a weighted average of estimates of interest from a set of candidate models. Many studies have investigated issues such as the choice of candidate models and weights.

Regarding the choice of weights, many schemes have been proposed. For example, Bates and Granger (1969) suggested to select the weights to be inversely related to estimated forecast error variances. Buckland, Burnham, and Augustin (1997) advocated choosing the weights using the Akaike Information Criteria (AIC) of all competing models. Somewhat surprisingly, an empirically highly successful strategy is the simple averaging method, which assigns each candidate model an equal weight; see Rapach et al. (2009) and Elliott et al. (2013).[1]

In recent years, several model averaging methods, all based on LS estimates of competing linear models, have been proposed. For example, Hansen (2007) proposed the Mallows model averaging (MMA) method and Hansen (2008) showed that the MMA weights are asymptotically mean-squared-forecast-error optimal in the i.i.d. framework. Xie (2015) put forward the prediction model averaging (PMA) method and showed that the PMA weights are asymptotically mean-squared optimal in the i.i.d. framework. Zhao, Zhang, and Gao (2016) further extended the PMA method to allow for heteroskedastic error terms (HPMA).

The model averaging problem can be formulated as follows. Consider a sequence of candidate models for $m = 1, ..., M$ such that $\boldsymbol{y} = \boldsymbol{X}_{(m)}\boldsymbol{\beta}_{(m)} + \epsilon_{(m)} = \boldsymbol{\mu}_{(m)} + \epsilon_{(m)}$. The predictors in candidate model m form the $T \times p_{(m)}$ matrix $\boldsymbol{X}_{(m)}$, which is a subset of $\boldsymbol{X}$ with $p_{(m)} \leq (p+1)$. Let the vector of weights be in the following unit simplex:

$$\mathcal{H} \equiv \left\{ \boldsymbol{w} \in [0,1]^M : \sum_{m=1}^{M} w_{(m)} = 1 \right\}. \tag{3}$$

The PMA method estimates $\boldsymbol{w}$ by

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w} \in \mathcal{H}}{\arg\min} \left\| \boldsymbol{y} - \boldsymbol{P}(\boldsymbol{w})\boldsymbol{y} \right\|^2 + 2\hat{\sigma}^2(\boldsymbol{w})p(\boldsymbol{w}),$$

where $\boldsymbol{P}(\boldsymbol{w}) \equiv \sum_{m=1}^{M} w_{(m)}\boldsymbol{P}_{(m)}$ with $\boldsymbol{P}_{(m)}$ being the projection matrix of $\boldsymbol{X}_{(m)}$, $p(\boldsymbol{w}) \equiv \sum_{m=1}^{M} w_{(m)}p_{(m)}$ is the effective number of parameters, and $\hat{\sigma}^2(\boldsymbol{w}) = \left\| \boldsymbol{y} - \boldsymbol{P}(\boldsymbol{w})\boldsymbol{y} \right\|^2 / (n - p(\boldsymbol{w}))$ is the averaged variance. The prediction of y_{T+h} by PMA is $\boldsymbol{X}_{T+h}^{\top}\hat{\boldsymbol{\beta}}(\hat{\boldsymbol{w}})$, where

$$\hat{\boldsymbol{\beta}}(\hat{\boldsymbol{w}}) = \sum_{m=1}^{M} \hat{w}_{(m)}\boldsymbol{\Gamma}_{(m)}\hat{\boldsymbol{\beta}}_{(m)},$$

with $\boldsymbol{\Gamma}_{(m)} = (\boldsymbol{X}^{\top}\boldsymbol{X})^{-1}\boldsymbol{X}^{\top}\boldsymbol{X}_{(m)}$ being a $(p+1) \times p_{(m)}$ binary matrix. $\boldsymbol{\Gamma}_{(m)}$ stretches the $p_{(m)} \times 1$ dimensional LS estimates $\hat{\boldsymbol{\beta}}_{(m)}$ to a dimension of $k \times 1$ by adding zeros.

The PMA method assumes homoskedastic errors. When the error terms exhibit het-

[1] In the Bayesian paradigm, the weights are the posterior distributions of the competing models, a by-product of the Bayesian model averaging.

eroskedasticity, we can adopt the heteroskedastic PMA (HPMA) by Zhao et al. (2016). In this case, the weight vector can be estimated by

$$\hat{\boldsymbol{w}} = \arg\min_{w \in \mathcal{H}} \|\boldsymbol{y} - \boldsymbol{P}(\boldsymbol{w})\boldsymbol{y}\|^2 + 2\sum_{t=1}^{T} \hat{\epsilon}_t(\boldsymbol{w})^2 P_{tt}(\boldsymbol{w}),$$

where $\hat{\epsilon}_t(\boldsymbol{w})$ is the t^{th} element of $\hat{\boldsymbol{\epsilon}}(\boldsymbol{w}) = \boldsymbol{y} - \boldsymbol{P}(\boldsymbol{w})\boldsymbol{y}$ and $P_{tt}(\boldsymbol{w})$ is the t^{th} diagonal element of $\boldsymbol{P}(\boldsymbol{w})$.

Regarding the choice of candidate models, Elliott et al. (2013) proposed the complete subset regression (CSR) that combines forecasts from all possible linear regression models with a fixed number of predictors (say q). The weights assigned to all subset regression models are identical. If the total number of predictors is p, the total number of candidate models is C_q^p and the weight for each model is $1/C_q^p$. If C_q^p is too large to handle, it is recommended by Genre et al. (2013) to randomly pick an acceptable number of candidate models instead.

When surveys of professional forecasters are available, forecast combination becomes a standard procedure to aggregate private information available to individual forecasters. In this case, a common practice is to take a simple average to combine survey-based forecasts and there is no need to estimate coefficients. Such a forecast is hard to beat as shown by Genre et al. (2013). As our empirical example is also based on survey-based forecasts, it is natural to use the simple averaging method as the **benchmark**.

2.3 Nonlinear machine learning methods

Methods discussed above rely on the linear formulation as in Equation (1). If the linear restriction is relaxed, we have

$$y_t = f(\boldsymbol{X}_t) + \epsilon_t, \tag{4}$$

where the function $f(\cdot)$ can be nonlinear or even nonparametric. Machine learning techniques do not try to find a consistent estimator of $f(\cdot)$ from data. Instead they try to search a function, which may or may not be analytically available, to approximate $f(\cdot)$ so that it generates sound predictions for y_t.

In this section, we first review the tree-type machine learning methods. The building block is regression tree (RT) proposed by Breiman, Friedman, and Stone (1984). Starting from the original data (the root node), all possible binary splits of the values for each predictor are considered and a "best split" is determined by certain criterion, for example, the reduction in the sum of squared residuals (SSR). Such a partitioning process can be implemented iteratively until it reaches a pre-determined boundary. Many modeling parameters need to be decided or calculated *ex ante*.[2] Data in the terminal nodes (also called

[2] These so-called tuning parameters, or hyperparameter, include but are not limited to a splitting criterion function, stopping rules, etc.

tree leaves) are considered to be homogeneous, hence a simple average of all the data $\boldsymbol{y}_l$ within the tree leaf l is used as the fitted value. To make predictions based on $\boldsymbol{X}_{T+h}$, we simply drop $\boldsymbol{X}_{T+h}$ down the tree and end up in a specific tree leaf l. The corresponding prediction, $\hat{y}_{T+h}$, is measured by the sample average of $\boldsymbol{y}_l$ for $\boldsymbol{y}_l \in \boldsymbol{y}$.

We can apply the bootstrap aggregation (bagging) technique developed in Breiman (1996) to RT. Using the original sample $\{y_t, \boldsymbol{X}_t\}_{t=1}^T$, the bagging RT (BAG) method first generates B bootstrap samples $\{y_t^{(b)}, \boldsymbol{X}_t^{(b)}\}_{t=1}^T$ for $b = 1, ..., B$, where the value of B must be predetermined. Next we apply RT to each bootstrap sample and obtain the prediction $\hat{y}_{T+h}^{(b)}$ based on $\boldsymbol{X}_{T+h}$. The final forecast by the BAG method is the simple average of all the B forecasts $\hat{y}_{T+h} = \frac{1}{B}\sum_{b=1}^{B} \hat{y}_{T+h}^{(b)}$. The variance of BAG forecasts can be large owing to the high correlation among trees. Such an issue can be circumvented by random forest (RF) of Breiman (2001). RF also constructs B new trees by bootstrapping, in which a random sample (without replacement) of q ($q < p$) predictors is taken for each splitting procedure within each tree. In this way, the trees for RF are less correlated and the final RF forecast is still the simple average of forecasts from all the constructed trees. There are many other tree-type methods worth mentioning. Section 1.2 of the online appendix reviews some other popular tree-type methods.

The tree-type methods search for heterogeneity within data set and categorize their features by the tree leaves. Another popular machine learning method that responds to local features of data is the support vector regression (SVR) proposed by Drucker, Burges, Kaufman, Smola, and Vapnik (1996). The SVR framework approximates $f(\boldsymbol{X}_t)$ in terms of a set of basis functions $\{h_s(\cdot)\}_{s=1}^S$:

$$y_t = f(\boldsymbol{X}_t) + \epsilon_t = \sum_{s=1}^{S} \beta_s h_s(\boldsymbol{X}_t) + \epsilon_t, \tag{5}$$

where $h_s(\cdot)$ is implicit and can be infinite-dimensional. Following Hastie, Tibshirani, and Friedman (2009, Chapter 12), the intercept is ignored for simplicity. The coefficients $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_S]^\top$ are estimated through the minimization of

$$H(\boldsymbol{\beta}) = \sum_{t=1}^{T} V_e\left(y_t - f(\boldsymbol{X}_t)\right) + \lambda \sum_{s=1}^{S} \beta_s^2, \tag{6}$$

where the loss function

$$V_e(r) = \begin{cases} 0 & \text{if } |r| < e \\ |r| - e & \text{otherwise} \end{cases}$$

is called an e-insensitive error measure that ignores errors of size less than e. As part of the loss function V_e, the parameter e is usually decided beforehand. On the other hand, λ is a more traditional regularization parameter that can be estimated by cross-validation.

Suykens and Vandewalle (1999) made a modification to SVR which leads to solving a set of linear equations under a squared loss function. The above method, known as the LSSVR, considers minimizing

$$H(\boldsymbol{\beta}) = \sum_{t=1}^{T} (y_t - f(\boldsymbol{X}_t))^2 + \lambda \sum_{s=1}^{S} \beta_s^2, \tag{7}$$

where a squared loss function replaces $V_e(\cdot)$ for the LSSVR.

We form up Lagrangian equations for (6) and (7) and solve for optimal solutions. The estimation functions for SVR and LSSVR take the following forms

$$\text{SVR} \quad : \quad \hat{f}(\boldsymbol{x}) = \sum_{t=1}^{T} (\hat{\alpha}_t^* - \hat{\alpha}_t') K(\boldsymbol{x}, \boldsymbol{X}_t), \tag{8}$$

$$\text{LSSVR} \quad : \quad \hat{f}(\boldsymbol{x}) = \sum_{t=1}^{T} \hat{\alpha}_t K(\boldsymbol{x}, \boldsymbol{X}_t), \tag{9}$$

for any given input variable $\boldsymbol{x}$. $\{\hat{\alpha}_t^*\}_{t=1}^T$ and $\{\hat{\alpha}_t'\}_{t=1}^T$ are the estimated Lagrangian multipliers for SVR,[3] $\{\hat{\alpha}_t\}_{t=1}^T$ are the estimated Lagrangian multipliers for LSSVR, and $K(\cdot, \cdot)$ is the predetermined kernel function. See Section 1.3 of the online appendix for a more comprehensive description of the estimation procedure.

As Equations (8) and (9) indicate, no explicit forms of the basis functions are demanded in the estimation procedure. It is the kernel function that plays a crucial role in the estimation process. In this paper, we consider the following kernel functions

$$\begin{aligned} \text{Linear} \quad &: \quad K(\boldsymbol{x}, \boldsymbol{X}_t) = \boldsymbol{x}^\top \boldsymbol{X}_t, \\ \text{Gaussian} \quad &: \quad K(\boldsymbol{x}, \boldsymbol{X}_t) = \exp\left(- \frac{\|\boldsymbol{x} - \boldsymbol{X}_t\|^2}{2\sigma_x^2} \right), \\ \text{Polynomial} \quad &: \quad K(\boldsymbol{x}, \boldsymbol{X}_t) = (\gamma + \boldsymbol{x}^\top \boldsymbol{X}_t)^d, \end{aligned}$$

where σ_x^2, γ, and d are hyperparameters. We denote the SVR, LSSVR with linear, Gaussian, polynomial kernels as SVR_L, SVR_G, SVR_P, LSSVR_L, LSSVR_G, and LSSVR_P, respectively. Note that SVR_L follows the linear formulation as in (1), and the corresponding basis function is explicit. This also occurs to LSSVR_L. In fact, the LSSVR with a linear kernel is identical to the ridge regression discussed in the online appendix.

[3] Note that additional Lagrangian multipliers are required for SVR estimation, since the absolute values in $V_e(\cdot)$ can be reformulated into two linear expressions.

3 Combing forecasts by machine learning methods

Most of the machine learning methods covered in Section 2 do not account for model specification uncertainty, which can be dangerously arrogant in practice. In an unstable forecasting environment, it is hard to believe that a single machine learning strategy[4] always generates the best forecast. In this paper, we apply the concept of forecast combination to outcomes from machine learning strategies. The proposed combinations of machine learning can be regarded as an ensemble learning algorithm.

3.1 Simple averaging machine learning

The simple averaging machine learning method assigns an equal weight to selected machine learning strategies. We denote this approach as simple averaging machine learning (SAML), which is general and applicable to any machine learning strategy. Suppose we have a set of M forecasts, each of which is generated by a candidate strategy. Denote $\hat{y}_{T+h}(m)$ the h-step-ahead forecast of y_{T+h} based on the m^{th} strategy. Then the simple averaging combination forecast of y_{T+h} is

$$\hat{y}^{\text{SA}}_{T+h} = \frac{1}{M}\sum_{m=1}^{M} \hat{y}_{T+h}(m), \tag{10}$$

where the superscript "SA" is short for simple averaging. For example, if we focus on LSSVR with the Gaussian kernel, a representative candidate strategy with all the included predictors can be implemented to produce one of the forecasts. Then the simple average of all the forecasts from a full combination of predictors can be used as the combined forecast of the response variable in this case. Clearly, this method requires all candidate strategies can generate reasonable forecasts. However, if a subset of candidate strategies generates poor forecasts, simple averaging may fail to deliver satisfactory out-of-sample performance. This idea is demonstrated in the simulation and in the empirical application. See Section 4 and Appendix D for further details.

3.2 Mallows-type averaging machine learning

In practice, more dynamic model weights sometimes generate more robust forecasting results than simple averaging. If we focus on machine learning techniques that meet the following condition, the weights can be obtained by minimizing the Mallows-type criteria. This type of ensemble learning is denoted as Mallows-type averaging machine learning (MAML).

[4] In this paper, machine learning strategies can represent various model specifications under one or many machine learning methods.

Condition 1 Given the formulation as in (4), the prediction based on predictors $\boldsymbol{x}$ must obey the following formulation

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{P}(\boldsymbol{x}, \boldsymbol{X})\boldsymbol{y}, \tag{11}$$

where $\boldsymbol{y}$ and $\boldsymbol{X}$ are the matrices of response variables and predictors, respectively, and the form of $\boldsymbol{P}(\boldsymbol{x}, \boldsymbol{X})$ is explicit.

Condition 1 requires that predictions based on $\boldsymbol{x}$ are a weighted average of $\boldsymbol{y}$ with the weights depending on $\boldsymbol{x}$ and $\boldsymbol{X}$ in a possibly nonlinear manner. It can be shown that many machine learning methods satisfy this condition. A more thorough discussion is provided in Section 3.3.

Let us assume that the m^{th} candidate strategy implies the following relationship between y_t and $\boldsymbol{X}_t^{(m)}$

$$y_t = f(\boldsymbol{X}_t^{(m)}) + \epsilon_t^{(m)},$$

where the $p^{(m)} \times 1$ vector $\boldsymbol{X}_t^{(m)}$ is a subset of $\boldsymbol{X}_t$ that includes the corresponding variables, and the superscript (m) indicates variables associated with the m^{th} strategy. Let $\boldsymbol{X}_{(m)}$ be the matrix of $\boldsymbol{X}_t^{(m)}$ for all t. We define $\hat{f}(\boldsymbol{X}_{(m)}) = \hat{\boldsymbol{f}}_{(m)} = \boldsymbol{P}_{(m)}\boldsymbol{y}$ as the prediction of $\boldsymbol{y}$ by the m^{th} candidate strategy, where $\boldsymbol{P}_{(m)} = \boldsymbol{P}(\boldsymbol{X}_{(m)}, \boldsymbol{X}_{(m)})$ is explicit for all $m = 1, ..., M$. Let the weight vector $\boldsymbol{w} \in \mathcal{H}$, where the set $\mathcal{H}$ is defined in (3). The weighted average prediction is given by

$$\hat{\boldsymbol{f}}(\boldsymbol{w}) = \sum_{m=1}^{M} w_{(m)}\hat{\boldsymbol{f}}_{(m)} = \boldsymbol{P}(\boldsymbol{w})\boldsymbol{y},$$

where $\boldsymbol{P}(\boldsymbol{w}) = \sum_{m=1}^{M} w_{(m)}\boldsymbol{P}_{(m)}$.

Inspired by the works of Hansen (2007), Hansen (2008), Xie (2015), Zhao et al. (2016) and Ullah and Wang (2013), we propose to estimate the weight vector $\boldsymbol{w}$ by minimizing either of the following Mallows-type criteria, with the restriction of $\boldsymbol{w} \in \mathcal{H}$ and various assumptions on the error term:

$$\begin{aligned} C_1(\boldsymbol{w}) &= \|\boldsymbol{y} - \boldsymbol{P}(\boldsymbol{w})\boldsymbol{y}\|^2 + 2\hat{\sigma}^2(\boldsymbol{w}) \sum_{t=1}^{T} P_{tt}(\boldsymbol{w}), & (12) \\ C_2(\boldsymbol{w}) &= \|\boldsymbol{y} - \boldsymbol{P}(\boldsymbol{w})\boldsymbol{y}\|^2 + 2\sum_{t=1}^{T} \hat{\epsilon}_t(\boldsymbol{w})^2 P_{tt}(\boldsymbol{w}), & (13) \end{aligned}$$

where $P_{tt}(\boldsymbol{w})$ is the t^{th} diagonal term in $\boldsymbol{P}(\boldsymbol{w})$. Note that C_1 assumes homoskedasticity and C_2 considers heteroskedasticity. Define the averaged residual by

$$\hat{\boldsymbol{\epsilon}}(\boldsymbol{w}) = \sum_{m=1}^{M} w_{(m)}\hat{\boldsymbol{\epsilon}}^{(m)} = (\boldsymbol{I} - \boldsymbol{P}(\boldsymbol{w}))\boldsymbol{y}.$$

Criterion C_1 incorporates the variance of the averaged error term

$$\hat{\sigma}^2(\boldsymbol{w}) = \|\boldsymbol{y} - \boldsymbol{P}(\boldsymbol{w})\boldsymbol{y}\|^2/T,$$

whereas C_2 acknowledges heteroskedasticity by considering the square of each element in the averaged residual vector, similar to HPMA. Estimating $\boldsymbol{w}$ by C_1 or C_2 is a convex optimization process. Once $\hat{\boldsymbol{w}}$ is obtained, the combination forecast of y_{T+h} is

$$\hat{y}^{\text{MA}}_{T+h} = \sum_{m=1}^{M} \hat{w}_{(m)} \hat{y}_{T+h}(m), \tag{14}$$

where the superscript "MA" is the abbreviation for Mallows-type averaging.

3.3 Condition 1 for machine learning methods

In this section, we discuss how several machine learning methods satisfy Condition 1 and demonstrate their representations of $\boldsymbol{P}(\boldsymbol{x}, \boldsymbol{X})$. The discussion starts with RT. To make predictions based on $\boldsymbol{x}$, we simply drop $\boldsymbol{x}$ down the constructed tree and end up with a specific tree leaf $\boldsymbol{y}_l$ with T_l observations. The related prediction is then measured by the simple average of all observations within $\boldsymbol{y}_l$. Since $\boldsymbol{y}_l$ is a subset of $\boldsymbol{y}$, it is obvious that the prediction based on $\boldsymbol{x}$ obeys

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{P}^{\text{RT}}(\boldsymbol{x}, \boldsymbol{X})\boldsymbol{y},$$

where $\boldsymbol{P}^{\text{RT}}(\boldsymbol{x}, \boldsymbol{X})$ is a $1 \times T$ sparse vector with elements of $1/T_l$ and zero otherwise, corresponding to their counterparts in $\boldsymbol{y}_l$. To conduct Mallows-type averaging on RT, we first construct the $T \times T$ matrix $\boldsymbol{P}^{\text{RT}}_{(m)}$ for each candidate strategy m, where the t^{th} row of $\boldsymbol{P}^{\text{RT}}_{(m)}$ is $\boldsymbol{P}^{\text{RT}}(\boldsymbol{X}^{(m)}_t, \boldsymbol{X})$. We then apply the collection of $\{\boldsymbol{P}^{\text{RT}}_{(m)}\}_{m=1}^{M}$ to C_1 or C_2 respectively, and compute for $\boldsymbol{w}$ through the convex optimization. The combined forecast with the estimated $\boldsymbol{w}$ is denoted as RT^{MA}. Similarly, RT^{SA} indicates the simple averaging RT.

Since RT satisfies Condition 1, it is straightforward to demonstrate that all the other tree-type methods in the online appendix satisfy Condition 1, which include BAG and RF algorithms. Note that $\boldsymbol{P}(\boldsymbol{x}, \boldsymbol{X})$ may not be sparse for ensemble tree methods since they incorporate trees based on multiple generated samples. In this paper, we also consider the combined forecasts based on Mallows-type averaged BAG and RF as representatives of tree-type ensemble methods, which are termed BAG^{MA} and RF^{MA}, respectively.

It is conventional for forecasts from regression trees to follow a local constant model that assumes homogeneity in outcomes within each terminal leave. Lehrer and Xie (2018) proposed strategies undertaking model averaging within tree leaves to generate forecasts. By permitting model uncertainty within each leaf subgroup, richer forms of heterogeneous relationships between input and output variables are allowed. Their methods focus on model uncertainty at the level of local leaves, while our Mallows-type averaging tree methods consider specification uncertainty globally across all the candidate strategies. To avoid notational confusion, we denote their model averaging BAG and RF methods as MAB and MARF, respectively.

The last considered method that satisfies Condition 1 is LSSVR. Suppose $\boldsymbol{H}$ be the $T \times r$ implicit basis matrix where $r > T$.[5] The coefficient $\boldsymbol{\beta}$ can be estimated by minimizing the following penalized LS criterion

$$H(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2.$$

The solution, $\hat{\boldsymbol{\beta}}$, should satisfy $-\boldsymbol{H}^\top(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{\beta}}) + \lambda\hat{\boldsymbol{\beta}} = 0$ and the in-sample prediction is given by

$$\hat{f}(\boldsymbol{X}) = \boldsymbol{H}\hat{\boldsymbol{\beta}} = \left(\boldsymbol{H}\boldsymbol{H}^\top + \lambda\boldsymbol{I}_T\right)^{-1}\boldsymbol{H}\boldsymbol{H}^\top\boldsymbol{y} \equiv \boldsymbol{P}^{\text{LSSVR}}(\boldsymbol{X})\boldsymbol{y}, \tag{15}$$

where $\boldsymbol{I}_T$ is a $T \times T$ identity matrix and

$$\boldsymbol{P}^{\text{LSSVR}}(\boldsymbol{X}) \equiv \left(\boldsymbol{H}\boldsymbol{H}^\top + \lambda\boldsymbol{I}_T\right)^{-1}\boldsymbol{H}\boldsymbol{H}^\top \tag{16}$$

is a $T \times T$ matrix. Note that the $T \times T$ matrix $\boldsymbol{H}\boldsymbol{H}^\top$ is the kernel matrix with elements being $K(\boldsymbol{X}_t, \boldsymbol{X}_{t'}) \equiv \sum_{s=1}^{S} h_s(\boldsymbol{X}_t)h_s(\boldsymbol{X}_{t'})$ for different t and t'. Equation (15) implies that although the basis matrix is implicit, we can still make predictions since the kernel matrix is explicit. Note that the above derivation is based on the no-intercept assumption following Hastie et al. (2009, Chapter 12). If an intercept must be included in the model, Equation (15) still holds but with a more complicated form of $\boldsymbol{P}^{\text{LSSVR}}(\boldsymbol{X})$. See Appendix A for a detailed discussion.

The Mallows-type averaging LSSVR is conducted in a similar fashion as RT^{MA}. We first construct the $T \times T$ matrix $\boldsymbol{P}_{(m)}^{\text{LSSVR}}$ for each candidate strategy m, and then apply the collection of $\{\boldsymbol{P}_{(m)}^{\text{LSSVR}}\}_{m=1}^{M}$ to C_1 or C_2 and compute for $\boldsymbol{w}$ through the convex optimization separately. The combined forecast based on Mallows-type averaging LSSVR with Gaussian and polynomial kernels are denoted as $\text{LSSVR}_{\text{G}}^{\text{MA}}$ and $\text{LSSVR}_{\text{P}}^{\text{MA}}$, respectively.

4 Monte Carlo simulations

To evaluate how the proposed machine learning-based combination methods work, we first conduct a Monte Carlo simulation experiment. We design the experiment to study

[5] In this paper, we consider Gaussian and polynomial kernels for LSSVR. With a linear kernel, the LSSVR method also follows the linear formulation, which is equivalent to the ridge regression discussed in the online appendix 1.1. In the linear case, the basis matrix is explicit and identical to $\boldsymbol{X}$, where we cannot impose $r > T$.

the forecasting performance of the proposed methods and compare them with conventional methods with and without combination, as well as with machine learning methods without considering combination.

Inspired by Lehrer and Xie (2018), the response variable is generated from the true DGP:

$$y_t = \sin(x_{1t}) + \cos(x_{2t}) + \epsilon_t \quad \text{for } t = 1, ..., T+1.$$

We assume that we have access to a set of p predictors $\boldsymbol{X}_t = [x_{1t}, x_{2t}, ..., x_{pt}]^\top$ and hence $p-2$ of them are redundant. The exact identification of $p-2$ redundant variables is unknown to us. Suppose all the $\{x_{it}\}_{i=1}^{p}$ follow $x_{it} \sim i.i.d.\mathrm{N}(0,4)$ for $i = 1, ..., p$ and the error term ϵ_t follows

$$\epsilon_t \sim \begin{cases} \mathrm{N}(0,1) & \text{under homoskedasticity,} \\ \mathrm{N}(0, 0.05x_{1t}^2 + 0.01) & \text{under heteroskedasticity.} \end{cases}$$

We generate the data for $t = 1, ..., T+1$ and use T periods of the sample as the training set. Finally, the forecasts of y_{T+1} are made based on the test set of $\boldsymbol{X}_{T+1}$.

We consider the following methods for forecasting y_{T+1}: (1) simple averaging forecast by using $\{x_{i,T+1}\}_{i=1}^{p}$ to predict y_{T+1}; (2) LS; (3) LASSO; (4) CSR; (5) RT; (6) BAG; (7) RF; (8) SVR_L; (9) SVR_G; (10) $\mathrm{LSSVR}_\mathrm{G}$; (11) RT^SA; (12) BAG^SA; (13) RF^SA; (14) $\mathrm{LSSVR}_\mathrm{G}^\mathrm{SA}$; (15) RT^MA; (16) BAG^MA; (17) RF^MA; (18) $\mathrm{LSSVR}_\mathrm{G}^\mathrm{MA}$.

In this experiment, we set $p = 4$. Other values of p have been tried and the results remain the same qualitatively. The situations of homoskedastic and heteroskedastic error terms are also verified, and the hyperparameters are set to their default values.[6] These include and are not limited to:

1. We include all the subset regressions with two non-constant predictors for CSR;
2. The penalty coefficient is set to one for LASSO, SVR_L, SVR_G, $\mathrm{LSSVR}_\mathrm{G}$, and $\mathrm{LSSVR}_\mathrm{G}^\mathrm{MA}$;
3. All the tree-type methods follow the settings (i) the minimum leaf size is one and (ii) the maximum number for splits is $T-1$;
4. The learning cycles are assumed to be 100 for all ensemble methods;
5. The number of selected predictors is set at $\lfloor p/3 \rfloor$ for all RF-type methods;
6. $\sigma_x = 1$ for the Gaussian kernel;
7. Candidate model sets are constructed by a full combination of all the included predictors.

[6] When setting the hyperparameters to some extreme values, the machine learning techniques can sometimes be surpassed by LS. However, the SAML and MAML methods always outperform their machine learning counterpart under the same values of hyperparameters.

For each method, the number of replications is set to $B = 1000$ and a list of forecasts $\hat{y}_{T+1}^{(b)}$ are compared with the actual $y_{T+1}^{(b)}$ for $b = 1, ..., B$. The forecasting performance is assessed by the following two criteria:

$$\begin{aligned}\text{MSFE} &= \frac{1}{B}\sum_{b=1}^{B} e_{(b)}^2, \\ \text{MAFE} &= \frac{1}{B}\sum_{b=1}^{B} |e_{(b)}|,\end{aligned}$$

where $e_{(b)} = y_{T+1}^{(b)} - \hat{y}_{T+1}^{(b)}$ is the forecast error in the b^{th} simulation.

Table 1 reports simulation results for $T = 50$ with the best result under each criterion in boldface. The first column reports alternative forecasting strategies, whereas Columns 2-3 and 4-5 correspond to the results under homoskedasticity and heteroskedasticity, respectively. We distinguish each MAML method with $C_1(\boldsymbol{w})$ from that with $C_2(\boldsymbol{w})$ using subscripts 1 and 2.

Table 1: Simulation results for $T = 50$

Method	Homoskedasticity		Heteroskedasticity	
	MSFE	MAFE	MSFE	MAFE
Benchmark	2.6843	1.3093	2.1737	1.1590
LS	2.1197	1.1677	1.5706	0.9793
LASSO	2.0018	1.1412	1.4905	0.9749
CSR	1.9919	1.1372	1.4730	0.9616
RT	2.2895	1.2023	1.4109	0.8924
BAG	1.5912	1.0063	1.0061	0.7568
RF	1.5954	1.0088	1.0259	0.7702
SVR_L	2.2128	1.1908	1.6641	0.9966
SVR_G	1.9019	1.1099	1.3680	0.9241
LSSVR_G	1.6232	1.0228	1.0652	0.7887
RT^SA	1.6223	1.0169	1.0272	0.7772
BAG^SA	1.5420	0.9946	0.9871	0.7631
RF^SA	1.6076	1.0176	1.0777	0.8069
$\text{LSSVR}_\text{G}^\text{SA}$	1.5673	1.0065	1.0411	0.7911
RT_1^MA	1.7403	1.0517	1.0875	0.7942
RT_2^MA	1.7649	1.0588	1.1014	0.7988
BAG_1^MA	1.5201	0.9828	0.9119	0.7147
BAG_2^MA	1.5249	**0.9816**	0.9110	0.7165
RF_1^MA	1.5200	0.9834	0.9148	0.7225
RF_2^MA	1.5215	0.9834	0.9164	0.7218
$\text{LSSVR}_\text{G1}^\text{MA}$	**1.5161**	0.9832	**0.8499**	**0.6803**
$\text{LSSVR}_\text{G2}^\text{MA}$	1.5203	0.9844	0.8705	0.6909

Several remarkable findings are worth stressing. The benchmark simple averaging method yields the lowest forecasting accuracy regardless of the formulation of error terms.

Overall, LASSO and SVRs also have poor performance. CSR performs better than LS in all cases, although the improvement is quite marginal. The performance of RT is disappointing under homoskedasticity. In contrast, its performance is improved dramatically under heteroskedasticity. Other machine learning methods have in general higher forecast accuracy than conventional strategies. Most importantly, we find out that the Mallows-type averaging methods (RT$^{\mathrm{MA}}$, BAG$^{\mathrm{MA}}$, RF$^{\mathrm{MA}}$, and LSSVR$_{\mathrm{G}}^{\mathrm{MA}}$) always improve on their base methods (RT, BAG, RF, and LSSVR$_{\mathrm{G}}$) in terms of yielding lower MSFE and MAFE. Under both homoskedasticity and heteroskedasticity, LSSVR$_{\mathrm{G1}}^{\mathrm{MA}}$ gains the best forecast accuracy according to MSFE, although its heteroskedasticity-robust version, LSSVR$_{\mathrm{G2}}^{\mathrm{MA}}$, manifests a fairly close performance.

On the other hand, all the SAML methods have less impressive performance. This finding is not surprising since the set of candidate strategies is constructed from the full combination of all the included predictors without any screening. Obviously, strategies that incorporate only the irrelevant predictors, tend to generate unsatisfactory forecasts. As a result, the impact of poor forecasts does not diminish due to the use of equal weights in SAML.

We extend the above exercise by considering a more dynamic setting with expanding training sample sizes of $T = 50, 100, ..., 500$. The outcomes are plotted in Figure 1, in which subplots (a) to (d) imply the MSFE under homoskedasticity, the MAFE under homoskedasticity, the MSFE under heteroskedasticity, and the MAFE under heteroskedasticity, respectively. To avoid the figure being cluttered, we only present the results by LS, LSSVR$_{\mathrm{G}}$, LSSVR$_{\mathrm{G1}}^{\mathrm{SA}}$, and LSSVR$_{\mathrm{G1}}^{\mathrm{MA}}$, which are captured by dotted, dash-dotted, dashed, and solid lines, respectively. For presentation convenience, we standardize all results by the risk of LS. The horizontal axis represents the sample size and the vertical axis stands for the estimated relative risk.

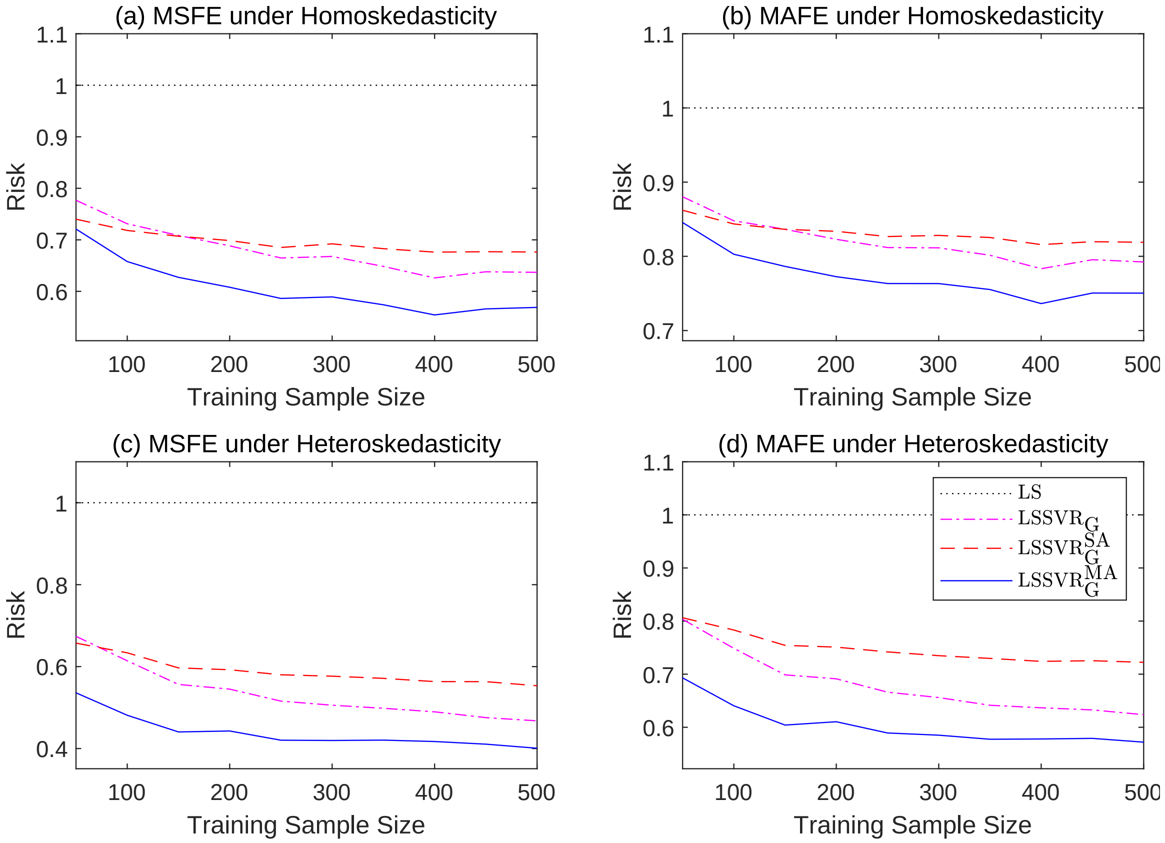
The pattern in Figure 1 is consistent. The results by LS are surely fixed at one for all T since it is the benchmark. The results by LSSVR$_{\mathrm{G}}$, LSSVR$_{\mathrm{G1}}^{\mathrm{SA}}$, and LSSVR$_{\mathrm{G1}}^{\mathrm{MA}}$ are downward sloping indicating that their gains relative to LS strengthen as T increases. LSSVR$_{\mathrm{G1}}^{\mathrm{SA}}$ has mixed performance relative to LSSVR$_{\mathrm{G}}$. In contrast, the line of LSSVR$_{\mathrm{G1}}^{\mathrm{MA}}$ is always below LSSVR$_{\mathrm{G}}$ in each subplot, which implies the advantage of LSSVR$_{\mathrm{G1}}^{\mathrm{MA}}$ as opposed to LSSVR$_{\mathrm{G}}$. We also notice that the relative risks by LSSVR$_{\mathrm{G1}}^{\mathrm{MA}}$ are lower under heteroskedasticity than those under homoskedasticity.

It is also interesting to further investigate the improvement of MAML over its base method. The comparison between LSSVR$_{\mathrm{G1}}^{\mathrm{MA}}$ and LSSVR$_{\mathrm{G}}$ is taken as an example, with the computed improvement ratio (IR) via

$$\mathrm{IR} = \frac{r_{\mathrm{LSSVR_G}} - r_{\mathrm{LSSVR_{G1}^{MA}}}}{r_{\mathrm{LSSVR_{G1}^{MA}}}} \times 100\%,$$

where $r_{\mathrm{LSSVR_G}}$ and $r_{\mathrm{LSSVR_{G1}^{MA}}}$ define the respective risks by LSSVR$_{\mathrm{G}}$ and LSSVR$_{\mathrm{G1}}^{\mathrm{MA}}$. Results for $T = 50, ..., 500$ are depicted in Figure 2. Subplots (a) to (d) correspond to the same cases

Figure 1: Simulation results by various sample sizes



(a) MSFE under Homoskedasticity

(b) MAFE under Homoskedasticity

(c) MSFE under Heteroskedasticity

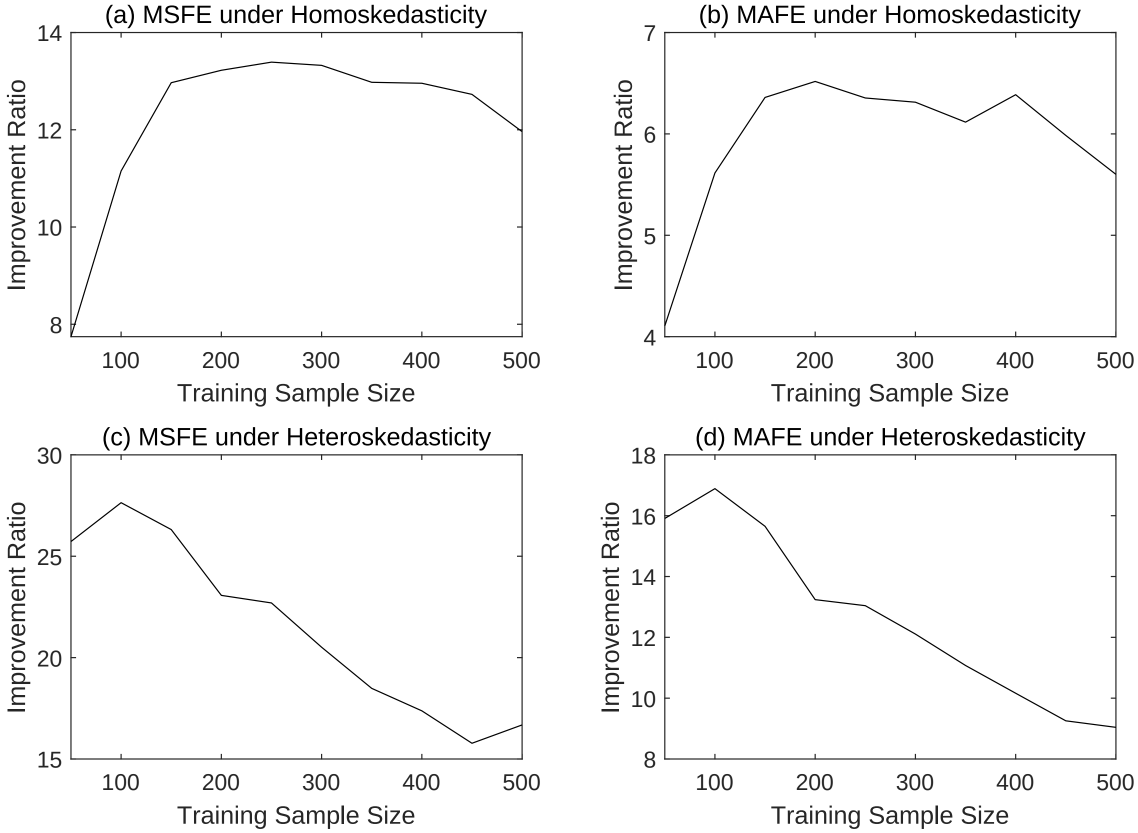
(d) MAFE under Heteroskedasticity

demonstrated in Figure 1. The horizontal axis represents the training sample size and the vertical axis stands for the estimated improvement ratio. The results also confirm that the improvement is more significant under heteroskedasticity for both prediction criteria.

5 Empirical applications

To illustrate the usefulness of the proposed methods, we now compare their performance with that of many existing methods using real data. Empirical applications to both macroeconomic variables and financial variables are considered. In particular, we consider forecasting three macroeconomic variables, inflation rates, real GDP growth rates, unemployment rates, and a financial variable, 3-month Treasury bill rate, based on surveys of professional forecasters. All four rates are essential to policy-makers and economic agents.

Figure 2: Improvement ratio by Mallows-type averaging



5.1 Data

When the Euro was launched in January 1999, the European Central Bank (ECB) started a Survey of Professional Forecasters (SPF) and collected the forecasters' views on future inflation rates, real GDP growth rates, and unemployment rates in the euro area. Genre et al. (2013) showed that a simple equally weighted pooling of forecasts performs well in practice relative to many alternative approaches, including those with estimated weights and the penalized LS.

In the following exercises, we first consider utilizing the forward-looking information from SPF, denoted as $\{x_{1t}, ..., x_{pt}\}$, to forecast the following three macroeconomic variables for the eurozone: (1) the inflation rate measured by the harmonized index of consumer prices (HICP); (2) the real GDP growth rate; (3) the unemployment rate. The raw data source from the official website.[7] From 1999Q1 to 2018Q4, the SPF collects one-year-ahead ($h = 4$) and two-year-ahead ($h = 8$) quarterly predictions of the above indicators from 119 different forecasters. However, an initial data cleaning is necessary since a specific forecaster may or may not submit a survey response each time throughout the whole

[7] http://www.ecb.europa.eu/stats/prices/indic/forecast/html/index.en.html

period. In the end, it boils down to 30 qualified forecasters for each indicator. Therefore, $p = 30$ and we use these 30 predictors to forecast each of the three macroeconomic variables. The detailed data polishing procedure is described in Appendix B. Note that our sample period extends that in Genre et al. (2013). Each forecaster may have his own private information to assist him in forecasting the three macroeconomic indicators.

As a financial application, we conduct a similar forecasting exercise on the 3-month Treasury bill nominal rate using the U.S. SPF data from the Federal Reserve Bank of Philadelphia.[8] The data span from 1992Q1 to 2019Q4 for one-year-ahead forecasts ($h = 4$) only. The same data polishing procedure leads to 21 qualified forecasters for predicting the Treasury bill rate. See Appendix B for complete details of data polishing.

We collect actual values of the considered indicators published by the officials.[9] The out-of-sample accuracy of each strategy is determined by comparing the actual values of the response variable with the forecasted values.

5.2 Nonlinearity

To motivate the implementation of machine learning methods and our proposed methods, it is helpful to first show that the assumption of a linear and additive relationship between y_t and $\boldsymbol{X}_t = [x_{1t}, ..., x_{pt}]^\top$ is too strong. The demonstration of a nonlinear relationship can explain the ineffectiveness of linear models. To do so, it would be ideal to consider a fully nonparametric function that relates y_t to $\boldsymbol{X}_t$. However, if the hypothetical relationship was imposed, one would face the curse of dimensionality for any nonparametric method because of the overwhelming 30 predictors. Therefore, we instead consider a partially linear model as

$$y_t = \boldsymbol{Z}_{1t}^\top \boldsymbol{\beta} + g(\boldsymbol{Z}_{2t}) + \epsilon_t, \tag{17}$$

where $\boldsymbol{Z}_{1t}$ is a $k \times 1$ vector, $\boldsymbol{\beta}$ is the associated $k \times 1$ coefficient vector, $\boldsymbol{Z}_{2t}$ is a $q \times 1$ vector (i.e., $q = p - k$), $g(\cdot)$ is an infeasible, possibly nonlinear function, and ϵ_t is the error term.

To contain the curse of dimensionality, a small q must be used. Following Li and Racine (2007), an infeasible estimator of $\boldsymbol{\beta}$ by the LS method is described by

$$\tilde{\boldsymbol{\beta}} = \left(\sum_{t=1}^{T} \tilde{\boldsymbol{Z}}_{1t} \tilde{\boldsymbol{Z}}_{1t}^{T} \right)^{-1} \sum_{t=1}^{T} \tilde{\boldsymbol{Z}}_{1t} \tilde{y}_t, \tag{18}$$

where $\tilde{\boldsymbol{Z}}_{1t} = \boldsymbol{Z}_{1t} - \mathbb{E}(\boldsymbol{Z}_{1t} | \boldsymbol{Z}_{2t})$ and $\tilde{y}_t = y_t - \mathbb{E}(y_t | \boldsymbol{Z}_{2t})$.

[8] https://www.philadelphiafed.org/research-and-data/real-time-center/survey-of-professional-forecasters

[9] The European data can be downloaded from https://sdw.ecb.europa.eu/, while the U.S. data source from https://fred.stlouisfed.org/.

In practice, the conditional expectations in (18) can be consistently estimated using the kernel method:

$$\begin{aligned}\hat{y}_t &\equiv \hat{\mathbb{E}}(y_t|\boldsymbol{Z}_{2t}) = T^{-1}\sum_{j=1}^{T} y_t K_h(\boldsymbol{Z}_{2t}, \boldsymbol{Z}_{2j}) \Big/ \hat{f}(\boldsymbol{Z}_{2t}),\\ \hat{\boldsymbol{Z}}_{1t} &\equiv \hat{\mathbb{E}}(\boldsymbol{Z}_{1t}|\boldsymbol{Z}_{2t}) = T^{-1}\sum_{j=1}^{T} \boldsymbol{Z}_{1j} K_h(\boldsymbol{Z}_{2t}, \boldsymbol{Z}_{2j}) \Big/ \hat{f}(\boldsymbol{Z}_{2t}),\end{aligned}$$

where $\hat{f}(\boldsymbol{Z}_{2t}) = T^{-1}\sum_{j=1}^{T} K_h(\boldsymbol{Z}_{2t}, \boldsymbol{Z}_{2j})$, $K_h(\boldsymbol{Z}_{2t}, \boldsymbol{Z}_{2j}) = \prod_{s=1}^{q} h_s^{-1} k\left(\frac{Z_{2ts} - Z_{2js}}{h_s}\right)$ with $k(\cdot)$ being the kernel function and h_s being the bandwidth for the s^{th} element in $\boldsymbol{Z}_{2t}$.

The presence of the random denominator $\hat{f}(\boldsymbol{Z}_{2t})$ can cause some technical difficulties when deriving the asymptotic distribution of the feasible estimator $\boldsymbol{\beta}$. We consider a simple approach that trims out observations for which the denominator is small and such a feasible estimator of $\boldsymbol{\beta}$ is defined by

$$\hat{\boldsymbol{\beta}} = \left(\sum_{t=1}^{T}(\boldsymbol{Z}_{1t} - \hat{\boldsymbol{Z}}_{1t})(\boldsymbol{Z}_{1t} - \hat{\boldsymbol{Z}}_{1t})^{\top}\right)^{-1} \sum_{t=1}^{T}(\boldsymbol{Z}_{1t} - \hat{\boldsymbol{Z}}_{1t})(y_t - \hat{y}_t)\mathbb{I}_t\left(\hat{f}(\boldsymbol{Z}_{2t}) \geq b\right), \tag{19}$$

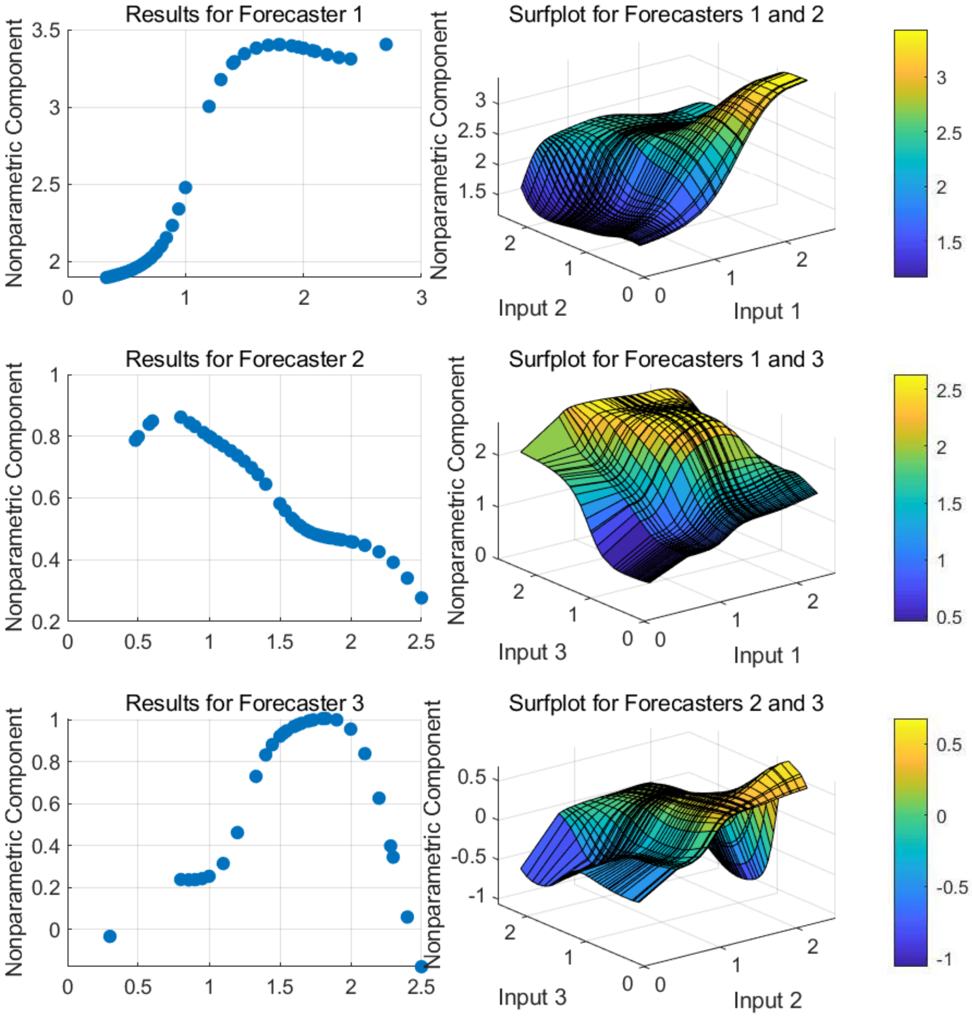
where $\mathbb{I}_t(\cdot)$ is an indicator function that equals one if the input argument is true and zero otherwise. The trimming parameter $b = b_n > 0$ and satisfies $b_n \to 0$ asymptotically. Once $\hat{\boldsymbol{\beta}}$ is obtained and the condition $\boldsymbol{Z}_{2t} = \boldsymbol{z}$ holds, the nonparametric components can be estimated consistently through

$$\hat{g}(\boldsymbol{z}) = \frac{\sum_{j=1}^{T}(y_t - \boldsymbol{Z}_{1t}^{\top}\hat{\boldsymbol{\beta}})K_h(\boldsymbol{z}, \boldsymbol{Z}_{2t})}{\sum_{j=1}^{T} K_h(\boldsymbol{z}, \boldsymbol{Z}_{2t})}. \tag{20}$$

In the following exercise, we concentrate on predicting the one-year-ahead HICP inflation ($h = 4$) as an example. We first obtain the top 3 forecasters (denoted as forecasts 1 to 3, respectively) based on RF, and then estimate the nonparametric components of various partially linear models. We use the Gaussian kernel with the optimal bandwidth $\hat{h}_z = 1.06\hat{\sigma}_z T^{-1/(q+4)}$ for each component of $\boldsymbol{Z}_{2t}$. Two possible scenarios are examined: (i) we choose one of the top 3 forecasters (i.e. $q = 1$) as the input variable for the nonparametric function; and (ii) we consider a full combination from any two of the top 3 forecasters (i.e. $q = 2$) as the predictors for the nonparametric function to generate a surface plot for each model. The complete results that verify nonlinearity and interactive effects for all the 30 predictors are presented in Section 2 of the online appendix.

The estimated $g(\boldsymbol{Z}_{2t})$ is plotted in Figure 3. The first and second columns of the subplots demonstrate the results in scenario (i) and (ii), respectively. Each subplot in the first column suggests that the relationship between the output and the single predictor is obviously nonlinear, while the subplots in the second column further sustain the finding of nonlinearity with some additional interactive effects uncovered between the two predictors. The above exercise points out clearly that it is inadequate to use a linear model, at least for our sample. This finding calls for the use of nonlinear methods such as the machine learning algorithms or our proposed methods. Nevertheless, we still include several conventional econometric methods assuming a linear relationship, as our reference methods for comparison.

Figure 3: Estimated nonparametric components by top 3 forecasters



5.3 Empirical results: macroeconomic indicators

In this section, we conduct one-year-ahead ($h = 4$) and two-year-ahead ($h = 8$) forecasting exercises for the three macroeconomic indicators using the data described in Section 5.1. We compare 31 forecasting methods fully described in Table 2. Principal tuning parameters and model settings for the machine learning techniques are[10]:

1. $\lambda = 0.5$ for LASSO, RIDGE, EN, SVRs, LSSVRs;
2. $\alpha = 0.5$ for EN;
3. Candidate model sets for the PMA and HPMA estimators are constructed by HEMS and HRMS, respectively;
4. We randomly pick 1000 models for CSR methods;
5. All the tree-type methods, with the exception of M5P-type, follow the settings (i) the minimum leaf size is one and (ii) the maximum number of splits is $T - 1$;
6. The minimum leaf size for the M5P-type algorithms is set at 5;
7. The number of learning cycles is set to $B = 100$ for all the ensemble methods;
8. The number of selected predictors is set to $\lfloor p/3 \rfloor$ for all the RF-type methods;
9. $\sigma_x^2 = 10$ for the Gaussian kernel;
10. $d = 10$, $p = 3$ for the polynomial kernel;
11. Candidate model sets are constructed following the generalized model screening (GMS) methodology for forecast combination machine learning algorithms.

One popular approach to constructing the candidate model set is to use a full combination of all p predictors, which leads to 1,073,741,824 candidate models in our exercise. Inspired by Xie (2017), we construct the candidate model set by the GMS method, which is a forward iterative procedure adding one predictor at a time according to pre-determined criteria. The generated post-screened model set is nested in sequence. See Appendix C for more details about the GMS method. Other hyperparameters not mentioned above follow conventional settings by defaults.

The window length is set to 40. The 31 companion methods are evaluated by MSFE, MAFE, SDFE, and Pseudo-R^2. The comparison results for the one-year-ahead ($h = 4$) and two-year-ahead ($h = 8$) HICP inflation rates are reported in Tables 3 and 4, respectively. Table 5 and 6 demonstrate the results for the real GDP growth, while the findings about

[10] We also consider alternative settings of tuning parameters. The results are qualitatively intact. See Appendix E for details.

Table 2: List of strategies

Method	Detailed description
Panel A: Benchmark and LS	
Benchmark	The equal weight pooling method recommended in Genre et al. (2013).
LS	Unrestricted ordinary least squares estimator.
Panel B: Penalized regression	
LASSO	The least absolute shrinkage selective operator by Tibshirani (1996).
RIDGE	The ridge regression.
EN	The elastic net method by Zou and Hastie (2005).
Panel C: Model averaging	
PMA	The prediction model averaging method by Xie (2015).
HPMA	The heteroskedasticity PMA method by Zhao et al. (2016).
CSR_{15}	The complete subset regression method by Elliott et al. (2013) with 15 selected predictors.
CSR_{20}	The complete subset regression method by Elliott et al. (2013) with 20 selected predictors.
Panel D: Classic machine learning	
RT	The regression tree method by Breiman et al. (1984).
BAG	The baggging tree method by Breiman (1996).
RF	The random forest method by Breiman (2001).
LSB	The LS RT boosting in Hastie et al. (2009, Chapter 10).
SVR_L	The support vector regression by Drucker et al. (1996) with linear kernel.
SVR_G	The support vector regression by Drucker et al. (1996) with Gaussian kernel.
SVR_P	The support vector regression by Drucker et al. (1996) with polynomial kernel.
Panel E: Advanced machine learning	
RT_M5P	The M5′ algorithm of Wang and Witten (1997) applied to RT.
BAG_M5P	The M5′ algorithm of Wang and Witten (1997) applied to BAG.
RF_M5P	The M5′ algorithm of Wang and Witten (1997) applied to RF.
LSSVR_G	The LS SVR method by Suykens and Vandewalle (1999) with Gaussian kernel.
LSSVR_P	The LS SVR method by Suykens and Vandewalle (1999) with polynomial kernel.
Panel F: Localized model averaging	
MAB	The model averaging tree leaf method applied to BAG by Lehrer and Xie (2018).
MARF	The model averaging tree leaf method applied to RF by Lehrer and Xie (2018).
*Panel G: Simple averaging machine learning (SAML)**	
BAG^SA	The simple averaging BAG method discussed in Section 3.
RF^SA	The simple averaging RF method discussed in Section 3.
$\text{LSSVR}_\text{G}^\text{SA}$	The simple averaging LSSVR method with Gaussian kernel discussed in Section 3.
*Panel H: Mallows-type averaging machine learning (MAML)**	
RT^MA	The Mallows-type averaging RT method discussed in Section 3.
BAG^MA	The Mallows-type averaging BAG method discussed in Section 3.
RF^MA	The Mallows-type averaging RF method discussed in Section 3.
$\text{LSSVR}_\text{G}^\text{MA}$	The Mallows-type averaging LSSVR method with Gaussian kernel discussed in Section 3.
$\text{LSSVR}_\text{P}^\text{MA}$	The Mallows-type averaging LSSVR method with polynomial kernel discussed in Section 3.

* Note that each method in this panel are estimated under homoskedastic and heteroskedastic error terms, which are denoted by subscripts 1 and 2, respectively. The candidate model sets are also constructed using the GMS method under homoskedastic and heteroskedastic error terms, respectively.

the unemployment rate are contained in Tables 7 and 8.[11] In all cases, the best results are presented in boldface. Some poor-performing methods are omitted to conserve space. A more complete comparison of all the methods in Table 2 is reported in Section 3 of the online appendix.

To examine if the improvement in forecast accuracy is significant, we perform the Giacomini-White (GW) test of the null hypothesis that the column method performs equally well as the row method in terms of absolute forecast errors. For simplicity, we only present the comparison outcomes between selective methods that yield the lowest MSFE in their separate panels of Table 2. The corresponding p-values are presented in Tables 9 to 14, respectively.

Some findings are worth mentioning. First, although some penalized regression and model averaging methods yield lower MSFEs than the benchmark, the improvement is not statistically significant, which coincides with the findings in Genre et al. (2013).

Second, some classic machine learning methods yield better results than the benchmark. This spells the importance of nonlinear and interactive effects. Moreover, advanced machine learning methods have an overall improved performance over the classic machine learning methods under the same hyperparameters parameters. It is also interesting to see that $\mathrm{LSSVR}_{\mathrm{G}}$ dominates the benchmark significantly at the 5% level in certain cases.

It is also noticeable that all the averaging methods beat their non-averaging counterparts. This signifies the importance of acknowledging model uncertainty in practice even with machine learning estimators. It is worth performing averaging estimation instead of relying on a single model.

Finally and most importantly, almost all the forecast combination machine learning methods (both SAML and MAML) outperform the benchmark, and many of these methods surpass the benchmark at the 5% level. The best method for predicting the HICP and the GDP growth is $\mathrm{LSSVR}_{\mathrm{G2}}^{\mathrm{SA}}$, while the best method for forecasting the unemployment rate is $\mathrm{BAG}_{1}^{\mathrm{MA}}$. The above results sustain the superiority of the proposed machine-learning-based combination.

It is also informative to contrast the performance of SAML with that of MAML. Since the candidate model sets are screened and selected by GMS, it is reasonable to assume that each candidate model generates acceptable forecasts. Therefore, it is not a surprise that many SAML methods yield fair forecast accuracy. To examine their sensitivity to various candidate model sets, we construct a "tainted" candidate model set which incorporates many poorly performing candidate models. The related outcomes in Appendix D reveal that the SAML methods are sensitive to the choice of the candidate model set,

[11] It is apparent from our exercise that the real GDP growth is the most difficult to forecast especially for $h = 4$. Lahiri and Sheng (2010) provide explanations on why the real GDP growth is more difficult to predict than the inflation rate.

while the MAML methods are less affected. Therefore, to apply the computationally efficient SAML methods in practice, it is necessary to first employ a reliable model screening technique. To examine the sensitivity to tuning parameters, we use alternative values of tuning parameters and present the results for forecasting one-year ahead HICP ($h = 4$) as an example in Appendix E. It is shown that the results are not sensitive to the choice of tuning parameters.

To further illustrate the forecast accuracy improvement by combining machine learning methods, we plot the forecasts of the benchmark and the best performing forecast combination machine learning method against the actual data in Figure 4. Subplots (a) to (f) depict one-year-ahead ($h = 4$) and two-year-ahead ($h = 8$) results of the HICP inflation rate, the real GDP growth rate, and the unemployment rate, respectively. It is clear that combining machine learning methods has better performance relative to the benchmark. For each subplot, the forecasts by the benchmark are fairly smooth and fail to capture the large fluctuations around 2012. This problem is more severe for the two-year-ahead forecasts. On the other hand, the combined machine learning forecasts are capable of capturing most of the movements.

5.4 Empirical results: the 3-month Treasury bill rate

In this section, we replicate our analysis to forecast the 3-month Treasury bill rate using the U.S. SPF data described in Section 5.1. Strategies are identical to those listed in Table 2 with the exception of CSR_{20}. CSR_{20} is replaced with CSR_{10}, since we only have 21 valid predictors (qualified forecasters) for this application. The window length is set to 40. Principal tuning parameters and model settings are identical to those listed in Section 5.3.

Results from the forecasting analysis are presented in Table 15 and selected results of the GW test are shown in Table 16. A more detailed comparison of all methods is reported in Section 3 of the online appendix. The results keep demonstrating the significant gains by using the proposed forecast combination machine learning methods. Figure 5 shows a closer trace between the actual data and the line of the best performing forecast combination machine learning method.

6 Conclusion

Forecast combinations based on linear models have found a wide range of applications in economics and finance with the presence of model uncertainty. More recently, machine learning techniques start enjoying remarkable out-of-sample gains due to their ability at capturing a nonlinear relationship between the response variable and the predictor.

When model uncertainty and nonlinearity occurs at the same time, it is expected that forecast combinations based on machine learning techniques shall be able to improve on the performance of individual machine learning techniques. The same conclusion also applies to combined forecasts of linear models. In this paper, we provide novel methods to combine machine learning forecasts. A straightforward way is to take a simple average of machine learning forecasts, where an equal weight is assigned to forecasts from a set of candidate machine learning strategies. In this case, there is no need to estimate the weights and therefore it is easy to implement. Moreover, it applies to any machine learning method. When all candidate machine learning strategies are reasonable in terms of generating a fair forecast, the method is expected to work well.

It is shown that the predictions with specific predictors can be expressed as a weighted average of historical response variables. The above condition is satisfied by many popular machine learning methods, including RT, Bagging, RF, and LSSVR. We also propose to use a weighted average machine learning forecasts with the weights estimated by minimizing Mallows-type criteria.

The advantage of the proposed methods is demonstrated using both simulated and real data. We consider a forecasting analysis of three major macroeconomic variables and a key financial variable, that is, the inflation rate, the real GDP growth, the unemployment rate for the euro area and the 3-month Treasury bill rate in the US. Not only do we find evidence of the outstanding performance of the proposed methods, but we also answer the question posed in the title of Genre et al. (2013) – can anything beat the simple average (the benchmark in our exercise)? Our answer is that our proposed methods not only beat the simple average but also do so by a wide margin.

Figure 4: A comparison of forecasts for three macroeconomic indicators

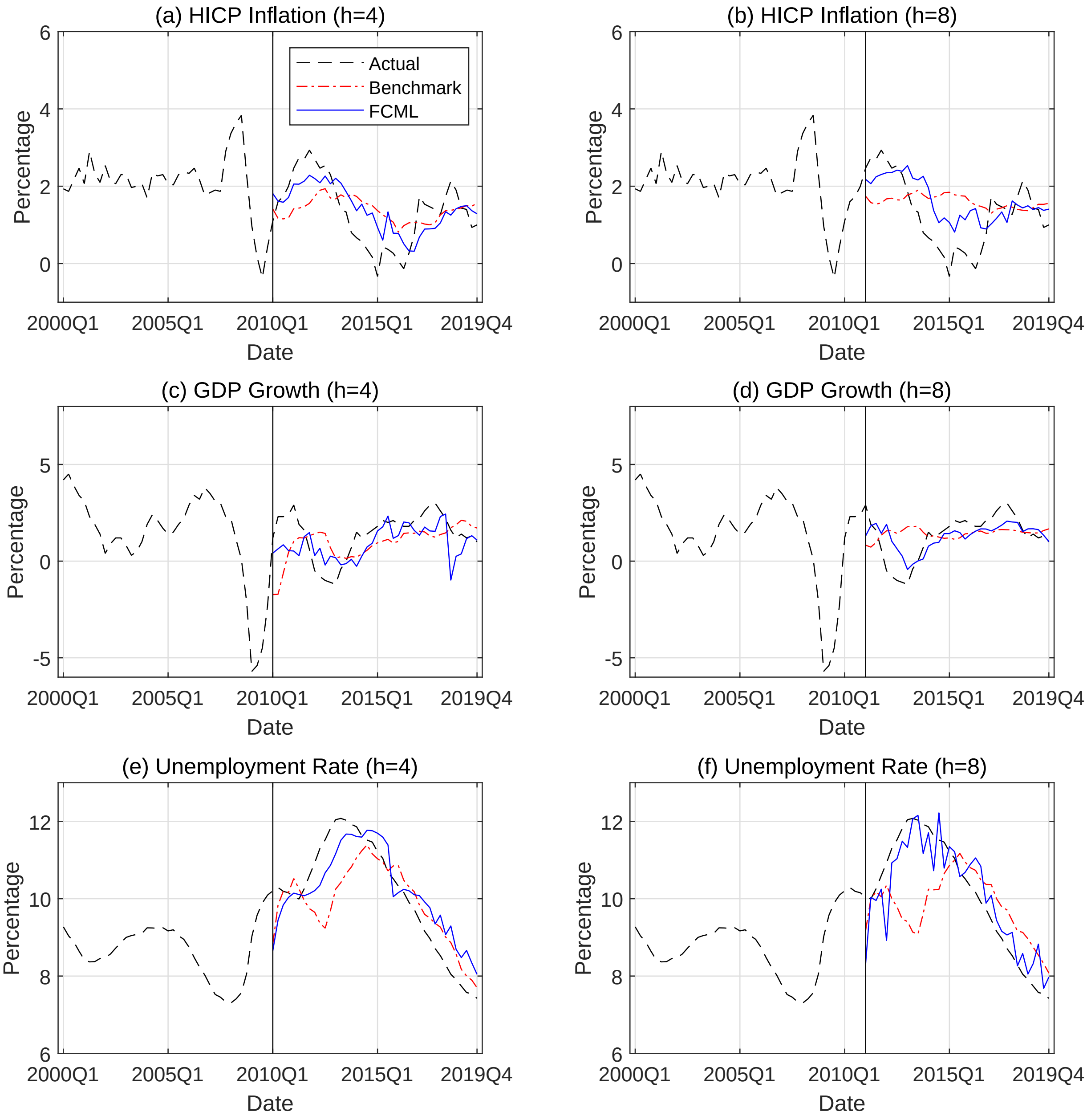


Figure 5: A comparison of forecasts for the 3-month Treasury bill rate

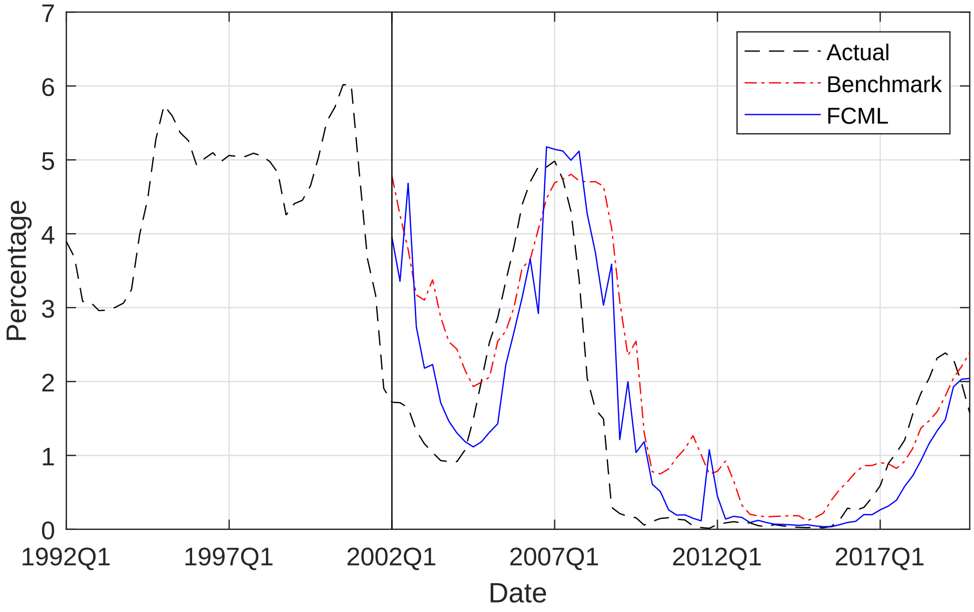


Table 3: Out-of-sample comparison of methods for forecasting the HICP inflation $(h = 4)$

Method	MSFE	MAFE	SDFE	Pseudo R^2
Benchmark	0.6533	0.7076	0.8083	0.1393
EN	0.7869	0.7486	0.8871	-0.0366
CSR_{15}	0.6716	0.6859	0.8195	0.1152
BAG	0.6000	0.6642	0.7746	0.2096
RF	0.5621	0.6497	0.7497	0.2596
LSSVR_{G}	0.4840	0.6004	0.6957	0.3625
BAG_1^{SA}	0.4655	0.5510	0.6823	0.3867
BAG_2^{SA}	0.4607	0.5743	0.6788	0.3931
RF_1^{SA}	0.4677	0.5624	0.6839	0.3839
RF_2^{SA}	0.4656	0.5633	0.6823	0.3867
$\text{LSSVR}_{\text{G1}}^{\text{SA}}$	**0.3767**	0.5316	**0.6138**	**0.5037**
$\text{LSSVR}_{\text{G2}}^{\text{SA}}$	0.3816	0.5301	0.6177	0.4974
BAG_1^{MA}	0.4386	**0.5247**	0.6623	0.4222
BAG_2^{MA}	0.4751	0.5803	0.6893	0.3741
RF_1^{MA}	0.4529	0.5447	0.6730	0.4033
RF_2^{MA}	0.4748	0.5568	0.6891	0.3745
$\text{LSSVR}_{\text{G1}}^{\text{MA}}$	0.4505	0.5840	0.6712	0.4065
$\text{LSSVR}_{\text{G2}}^{\text{MA}}$	0.4451	0.5804	0.6672	0.4136

This table reports the out-of-sample results for predicting the one-year-ahead HICP inflation rate using various methods. The best result under each criterion is highlighted in boldface.

Table 4: Out-of-sample comparison of methods for forecasting the HICP inflation $(h = 8)$

Method	MSFE	MAFE	SDFE	Pseudo R^2
Benchmark	0.9991	0.8408	0.9996	-0.2134
EN	0.8559	0.7294	0.9251	-0.0394
CSR_{15}	0.9730	0.7737	0.9864	-0.1817
BAG	0.7900	0.6969	0.8888	0.0405
RF	0.7620	0.7005	0.8729	0.0746
LSSVR_{G}	0.7122	0.6940	0.8439	0.1350
BAG_1^{SA}	0.6234	0.6189	0.7895	0.2429
BAG_2^{SA}	0.5821	**0.5806**	0.7630	0.2930
RF_1^{SA}	0.5749	0.6033	0.7582	0.3018
RF_2^{SA}	0.6967	0.6514	0.8347	0.1539
$\text{LSSVR}_{\text{G1}}^{\text{SA}}$	0.5915	0.6389	0.7691	0.2817
$\text{LSSVR}_{\text{G2}}^{\text{SA}}$	**0.5321**	0.6005	**0.7294**	**0.3538**
BAG_1^{MA}	0.6573	0.6253	0.8107	0.2018
BAG_2^{MA}	0.6462	0.6194	0.8039	0.2152
RF_1^{MA}	0.5647	0.5983	0.7514	0.3142
RF_2^{MA}	0.7069	0.6593	0.8408	0.1415
$\text{LSSVR}_{\text{G1}}^{\text{MA}}$	0.6989	0.6888	0.8360	0.1511
$\text{LSSVR}_{\text{G2}}^{\text{MA}}$	0.6687	0.6755	0.8177	0.1879

This table reports the out-of-sample results for predicting the two-year-ahead HICP inflation rate using various methods. The best result under each criterion is highlighted in boldface.

Table 5: Out-of-sample comparison for forecasting the real GDP growth rate $(h = 4)$

Method	MSFE	MAFE	SDFE	Pseudo R^2
Benchmark	2.0836	1.1793	1.4435	-0.6455
EN	3.0558	1.3987	1.7481	-1.4132
CSR_{15}	14.5770	2.2942	3.8180	-10.5118
BAG	2.5869	1.3132	1.6084	-1.0430
RF	2.1886	1.2589	1.4794	-0.7284
LSSVR_G	1.3094	0.9447	1.1443	-0.0340
BAG_1^{SA}	2.6007	1.2955	1.6127	-1.0538
BAG_2^{SA}	2.8628	1.3730	1.6920	-1.2608
RF_1^{SA}	2.3190	1.2817	1.5228	-0.8314
RF_2^{SA}	2.6005	1.3299	1.6126	-1.0537
$\text{LSSVR}_{\text{G1}}^{\text{SA}}$	1.2622	0.9119	1.1235	0.0032
$\text{LSSVR}_{\text{G2}}^{\text{SA}}$	**1.1624**	**0.8693**	**1.0781**	**0.0821**
BAG_1^{MA}	2.4408	1.3007	1.5623	-0.9276
BAG_2^{MA}	2.4648	1.3356	1.5700	-0.9465
RF_1^{MA}	2.0742	1.2536	1.4402	-0.6380
RF_2^{MA}	2.3252	1.3397	1.5249	-0.8363
$\text{LSSVR}_{\text{G1}}^{\text{MA}}$	1.2687	0.9191	1.1264	-0.0019
$\text{LSSVR}_{\text{G2}}^{\text{MA}}$	1.2496	0.8891	1.1179	0.0132

This table reports the out-of-sample results for predicting the one-year-ahead real GDP growth rate using various methods. The best result under each criterion is highlighted in boldface.

Table 6: Out-of-sample comparison for forecasting the real GDP growth rate $(h = 8)$

Method	MSFE	MAFE	SDFE	Pseudo R^2
Benchmark	1.6051	0.9789	1.2669	-0.2110
EN	1.3928	0.9298	1.1802	-0.0508
CSR_{15}	3.7323	1.4609	1.9319	-1.8159
BAG	1.2910	0.8247	1.1362	0.0260
RF	1.0883	0.7888	1.0432	0.1789
LSSVR_G	0.7691	0.6819	0.8770	0.4197
BAG_1^{SA}	1.6854	0.8376	1.2982	-0.2716
BAG_2^{SA}	1.7388	0.8376	1.3186	-0.3119
RF_1^{SA}	1.5294	0.8306	1.2367	-0.1539
RF_2^{SA}	1.3813	0.8316	1.1753	-0.0422
$\text{LSSVR}_{\text{G1}}^{\text{SA}}$	0.7288	0.6303	0.8537	0.4502
$\text{LSSVR}_{\text{G2}}^{\text{SA}}$	**0.6949**	**0.6278**	**0.8336**	**0.4757**
BAG_1^{MA}	1.7272	0.8404	1.3142	-0.3031
BAG_2^{MA}	1.6301	0.7807	1.2768	-0.2299
RF_1^{MA}	1.4139	0.7672	1.1891	-0.0668
RF_2^{MA}	1.2482	0.7712	1.1172	0.0582
$\text{LSSVR}_{\text{G1}}^{\text{MA}}$	0.7459	0.6660	0.8636	0.4373
$\text{LSSVR}_{\text{G2}}^{\text{MA}}$	0.7673	0.6525	0.8760	0.4211

This table reports the out-of-sample results for predicting the two-year ahead real GDP growth rate using different methods. The best result under each criterion is highlighted in boldface.

Table 7: Out-of-sample comparison for forecasting the unemployment rate $(h = 4)$

Method	MSFE	MAFE	SDFE	Pseudo R^2
Benchmark	0.8257	0.6928	0.9087	0.5835
EN	1.3143	0.9854	1.1464	0.3369
CSR$_{15}$	1.9975	1.0162	1.4133	-0.0077
BAG	0.6248	0.6886	0.7904	0.6848
RF	0.7661	0.7738	0.8753	0.6135
LSSVR$_G$	0.7787	0.7727	0.8824	0.6072
BAG$_1^{SA}$	0.4821	0.6123	0.6943	0.7568
BAG$_2^{SA}$	0.5082	0.6325	0.7129	0.7436
RF$_1^{SA}$	0.5579	0.6550	0.7469	0.7185
RF$_2^{SA}$	0.5508	0.6450	0.7421	0.7222
LSSVR$_{G1}^{SA}$	0.6692	0.7171	0.8181	0.6624
LSSVR$_{G2}^{SA}$	0.6551	0.7028	0.8094	0.6695
BAG$_1^{MA}$	**0.4100**	**0.5487**	**0.6403**	**0.7932**
BAG$_2^{MA}$	0.5065	0.6007	0.7117	0.7445
RF$_1^{MA}$	0.4818	0.5908	0.6941	0.7569
RF$_2^{MA}$	0.4793	0.5942	0.6923	0.7582
LSSVR$_{G1}^{MA}$	0.7301	0.7404	0.8545	0.6317
LSSVR$_{G2}^{MA}$	0.7382	0.7497	0.8592	0.6276

This table reports the out-of-sample results for predicting the one-year ahead unemployment rate using different methods. The best result under each criterion is highlighted in boldface.

Table 8: Out-of-sample comparison for forecasting the unemployment rate $(h = 8)$

Method	MSFE	MAFE	SDFE	Pseudo R^2
Benchmark	1.6665	1.0528	1.2909	0.2429
EN	2.1605	1.3516	1.4699	0.0185
CSR$_{15}$	3.0727	1.3871	1.7529	-0.3959
BAG	1.0079	0.8646	1.0040	0.5421
RF	1.3792	1.0635	1.1744	0.3734
LSSVR$_G$	0.7887	0.8182	0.8881	0.6417
BAG$_1^{SA}$	1.0154	0.9014	1.0077	0.5387
BAG$_2^{SA}$	1.4777	1.0520	1.2156	0.3287
RF$_1^{SA}$	1.0470	0.9175	1.0232	0.5244
RF$_2^{SA}$	1.0486	0.9184	1.0240	0.5236
LSSVR$_{G1}^{SA}$	0.8846	0.8498	0.9405	0.5982
LSSVR$_{G2}^{SA}$	0.8819	0.8486	0.9391	0.5993
BAG$_1^{MA}$	**0.7140**	**0.7562**	**0.8450**	**0.6756**
BAG$_2^{MA}$	0.7812	0.7839	0.8839	0.6451
RF$_1^{MA}$	0.7946	0.7957	0.8914	0.6390
RF$_2^{MA}$	0.8113	0.7996	0.9007	0.6314
LSSVR$_{G1}^{MA}$	0.7830	0.8148	0.8849	0.6443
LSSVR$_{G2}^{MA}$	0.7825	0.8145	0.8846	0.6445

This table reports the out-of-sample results for predicting the two-year ahead unemployment rate using different methods. The best result under each criterion is highlighted in boldface.

Table 9: Selected results of the GW test for the HICP Inflation $(h = 4)$

	Benchmark	EN	CSR_{15}	RF	LSSVR_{G}	MARF	$\text{LSSVR}^{\text{SA}}_{\text{G1}}$	$\text{RT}^{\text{MA}}_{2}$
Benchmark	-	-	-	-	-	-	-	-
EN	0.4651	-	-	-	-	-	-	-
CSR_{15}	0.7785	0.5241	-	-	-	-	-	-
RF	0.3444	0.0542	0.6888	-	-	-	-	-
LSSVR_{G}	0.0260	0.0040	0.3405	0.1772	-	-	-	-
MARF	0.0875	0.0048	0.3037	0.0118	0.8780	-	-	-
$\text{LSSVR}^{\text{SA}}_{\text{G1}}$	0.0022	0.0006	0.0624	0.0006	0.0123	0.0451	-	-
$\text{RT}^{\text{MA}}_{2}$	0.0176	0.0014	0.0478	0.0025	0.1091	0.0444	0.6571	-

The modified Giacomini-White test (Giacomini and White, 2006) is implemented to test the null hypothesis that the *row method* (in vertical headings) performs equally well as the *column method* (in horizontal headings) in terms of the absolute forecast error.

Table 10: Selected results of the GW test for the HICP Inflation $(h = 8)$

	Benchmark	EN	CSR_{15}	SVR_{L}	RF_{M5P}	MARF	$\text{LSSVR}^{\text{SA}}_{\text{G2}}$	$\text{RF}^{\text{MA}}_{1}$
Benchmark	-	-	-	-	-	-	-	-
EN	0.0776	-	-	-	-	-	-	-
CSR_{15}	0.5906	0.7061	-	-	-	-	-	-
SVR_{L}	0.0135	0.1121	0.0244	-	-	-	-	-
RF_{M5P}	0.0168	0.1827	0.2917	0.2011	-	-	-	-
MARF	0.0142	0.2387	0.2690	0.1985	0.9344	-	-	-
$\text{LSSVR}^{\text{SA}}_{\text{G2}}$	0.0027	0.0168	0.1033	0.7523	0.0669	0.0997	-	-
$\text{RF}^{\text{MA}}_{1}$	0.0079	0.0294	0.0949	0.7716	0.0883	0.0872	0.9542	-

The modified Giacomini-White test (Giacomini and White, 2006) is implemented to test the null hypothesis that the *row method* (in vertical headings) performs equally well as the *column method* (in horizontal headings) in terms of the absolute forecast error.

Table 11: Selected results of the GW test for real GDP growth rate $(h = 4)$

	Benchmark	LASSO	CSR_{20}	SVR_{G}	LSSVR_{G}	MARF	$\text{LSSVR}^{\text{SA}}_{\text{G2}}$	$\text{LSSVR}^{\text{MA}}_{\text{G2}}$
Benchmark	-	-	-	-	-	-	-	-
LASSO	0.1006	-	-	-	-	-	-	-
CSR_{20}	0.0188	0.1025	-	-	-	-	-	-
SVR_{G}	0.1605	0.0071	0.0261	-	-	-	-	-
LSSVR_{G}	0.0839	0.0040	0.0117	0.7495	-	-	-	-
MARF	0.2845	0.5371	0.0506	0.0865	0.0155	-	-	-
$\text{LSSVR}^{\text{SA}}_{\text{G2}}$	0.0215	0.0007	0.0059	0.3856	0.0419	0.0024	-	-
$\text{LSSVR}^{\text{MA}}_{\text{G2}}$	0.0323	0.0016	0.0075	0.4568	0.0491	0.0060	0.5060	-

The modified Giacomini-White test (Giacomini and White, 2006) is implemented to test the null hypothesis that the *row method* (in vertical headings) performs equally well as the *column method* (in horizontal headings) in terms of the absolute forecast error.

Table 12: Selected results of the GW test for real GDP growth rate $(h = 8)$

	Benchmark	EN	CSR_{20}	RF	LSSVR_{G}	MARF	$\text{LSSVR}^{\text{SA}}_{\text{G2}}$	$\text{LSSVR}^{\text{MA}}_{\text{G1}}$
Benchmark	-	-	-	-	-	-	-	-
EN	0.5883	-	-	-	-	-	-	-
CSR_{20}	0.0682	0.0094	-	-	-	-	-	-
RF	0.0300	0.0090	0.0027	-	-	-	-	-
LSSVR_{G}	0.0050	0.0009	0.0012	0.0862	-	-	-	-
MARF	0.1467	0.1579	0.0051	0.5434	0.1559	-	-	-
$\text{LSSVR}^{\text{SA}}_{\text{G2}}$	0.0032	0.0002	0.0003	0.0162	0.1801	0.0534	-	-
$\text{LSSVR}^{\text{MA}}_{\text{G1}}$	0.0034	0.0007	0.0010	0.0520	0.0711	0.1166	0.3371	-

The modified Giacomini-White test (Giacomini and White, 2006) is implemented to test the null hypothesis that the *row method* (in vertical headings) performs equally well as the *column method* (in horizontal headings) in terms of the absolute forecast error.

Table 13: Selected results of the GW test for the unemployment rate $(h = 4)$

	Benchmark	EN	CSR_{15}	BAG	BAG_{M5P}	MAB	$\text{BAG}^{\text{SA}}_{1}$	$\text{BAG}^{\text{MA}}_{1}$
Benchmark	-	-	-	-	-	-	-	-
EN	0.0032	-	-	-	-	-	-	-
CSR_{15}	0.0694	0.8528	-	-	-	-	-	-
BAG	0.9542	0.0000	0.0415	-	-	-	-	-
BAG_{M5P}	0.8402	0.0001	0.0397	0.7648	-	-	-	-
MAB	0.6305	0.0001	0.0278	0.1200	0.5045	-	-	-
$\text{BAG}^{\text{SA}}_{1}$	0.2779	0.0000	0.0158	0.0075	0.0631	0.1779	-	-
$\text{BAG}^{\text{MA}}_{1}$	0.0976	0.0000	0.0048	0.0002	0.0016	0.0027	0.0142	-

The modified Giacomini-White test (Giacomini and White, 2006) is implemented to test the null hypothesis that the *row method* (in vertical headings) performs equally well as the *column method* (in horizontal headings) in terms of the absolute forecast error.

Table 14: Selected results of the GW test for the unemployment rate $(h = 8)$

	Benchmark	RIDGE	CSR_{20}	LSB	LSSVR_{G}	MAB	$\text{LSSVR}^{\text{SA}}_{\text{G2}}$	$\text{BAG}^{\text{MA}}_{1}$
Benchmark	-	-	-	-	-	-	-	-
RIDGE	0.4332	-	-	-	-	-	-	-
CSR_{20}	0.8565	0.2072	-	-	-	-	-	-
LSB	0.0016	0.0002	0.0020	-	-	-	-	-
LSSVR_{G}	0.0621	0.0076	0.1494	0.0018	-	-	-	-
MAB	0.0246	0.0071	0.1232	0.0037	0.7646	-	-	-
$\text{LSSVR}^{\text{SA}}_{\text{G2}}$	0.0879	0.0233	0.2261	0.0021	0.2652	0.4319	-	-
$\text{BAG}^{\text{MA}}_{1}$	0.0094	0.0070	0.0716	0.0072	0.3011	0.3482	0.1178	-

The modified Giacomini-White test (Giacomini and White, 2006) is implemented to test the null hypothesis that the *row method* (in vertical headings) performs equally well as the *column method* (in horizontal headings) in terms of the absolute forecast error.

Table 15: Out-of-sample comparison for forecasting the 3-month Treasury bill rate $(h = 4)$

Method	MSFE	MAFE	SDFE	Pseudo R^2
Benchmark	1.6922	0.9356	1.3009	0.2127
EN	1.9962	0.9851	1.4129	0.0712
CSR_{10}	1.5222	0.8576	1.2338	0.2918
BAG	1.4343	0.7999	1.1976	0.3327
RF	1.4447	0.8177	1.2020	0.3278
LSSVR_{G}	1.4737	0.8695	1.2140	0.3143
BAG_1^{SA}	1.1096	0.7243	1.0534	0.4837
BAG_2^{SA}	1.1267	0.7049	1.0614	0.4758
RF_1^{SA}	1.1583	0.7438	1.0762	0.4611
RF_2^{SA}	1.0999	0.7237	1.0488	0.4882
$\text{LSSVR}_{\text{G1}}^{\text{SA}}$	1.3233	0.8048	1.1503	0.3843
$\text{LSSVR}_{\text{G2}}^{\text{SA}}$	1.3557	0.8335	1.1643	0.3692
BAG_1^{MA}	1.1541	0.7149	1.0743	0.4630
BAG_2^{MA}	1.1712	0.7162	1.0822	0.4551
RF_1^{MA}	1.0825	0.6981	1.0404	0.4963
RF_2^{MA}	**1.0605**	**0.7164**	**1.0298**	**0.5066**
$\text{LSSVR}_{\text{G1}}^{\text{MA}}$	1.4188	0.8418	1.1911	0.3399
$\text{LSSVR}_{\text{G2}}^{\text{MA}}$	1.4302	0.8460	1.1959	0.3346

This table reports out-of-sample results for predicting the one-year ahead 3-month Treasury bill rate using various strategies. The best result under each criterion is highlighted in boldface.

Table 16: Selected results of the GW test for the 3-month Treasury bill rate $(h = 4)$

	Benchmark	EN	CSR_{10}	BAG	RF_{M5P}	MAB	RF_2^{SA}	RF_2^{MA}
Benchmark	-	-	-	-	-	-	-	-
EN	0.4944	-	-	-	-	-	-	-
CSR_{10}	0.2380	0.1024	-	-	-	-	-	-
BAG	0.0552	0.0005	0.4107	-	-	-	-	-
RF_{M5P}	0.0403	0.0012	0.3953	0.8933	-	-	-	-
MAB	0.0055	0.0001	0.0731	0.0003	0.0030	-	-	-
RF_2^{SA}	0.0023	0.0000	0.0539	0.0052	0.0029	0.9886	-	-
RF_2^{MA}	0.0045	0.0004	0.0533	0.0207	0.0336	0.7830	0.7512	-

The modified Giacomini-White test (Giacomini and White, 2006) is implemented to test the null hypothesis that the *row method* (in vertical headings) performs equally well as the *column method* (in horizontal headings) in terms of the absolute forecast error.

References

BARNARD, G. A. (1963): "New Methods of Quality Control," *Journal of the Royal Statistical Society. Series A (General)*, 126, 255–258.

BATES, J. M. AND C. W. J. GRANGER (1969): "The Combination of Forecasts," *OR*, 20, 451–468.

BELLONI, A. AND V. CHERNOZHUKOV (2013): "Least Squares After Model Selection in High-dimensional Sparse Models," *Bernoulli*, 19, 521–547.

BOX, G. E. P. (1976): "Science and Statistics," *Journal of the American Statistical Association*, 71, 791–799.

BREIMAN, L. (1996): "Bagging Predictors," *Machine Learning*, 26, 123–140.

——— (2001): "Random Forests," *Machine Learning*, 45, 5–32.

BREIMAN, L., J. FRIEDMAN, AND C. J. STONE (1984): *Classification and Regression Trees*, Chapman and Hall/CRC.

BUCKLAND, S. T., K. P. BURNHAM, AND N. H. AUGUSTIN (1997): "Model Selection: An Integral Part of Inference," *Biometrics*, 53, 603–618.

CHAUDHURI, P., W.-D. LO, W.-Y. LOH, AND C.-C. YANG (1995): "Bagging Predictors," *Generalized Regression Trees*, 5, 641–666.

CLEMEN, R. T. (1989): "Combining forecasts: A review and annotated bibliography," *International Journal of Forecasting*, 5, 559–583.

CLEMEN, R. T. AND R. L. WINKLER (1999): "Combining Probability Distributions From Experts in Risk Analysis," *Risk Analysis*, 19, 187–203.

COULOMBE, P. G., M. LEROUX, D. STEVANOVIC, AND S. SURPRENANT (2020): "How is Machine Learning Useful for Macroeconomic Forecasting?" *Working Paper*.

DE BRABANTER, K., J. DE BRABANTER, J. A. K. SUYKENS, AND B. DE MOOR (2011): "Approximate Confidence and Prediction Intervals for Least Squares Support Vector Regression," *IEEE Transactions on Neural Networks*, 22, 110–120.

DIEBOLD, F. X. (2017): "Forecasting in Economics, Business, Finance and Beyond," *Unpublished Manuscript*.

DIEBOLD, F. X. AND P. PAULY (1987): "Structural change and the combination of forecasts," *Journal of Forecasting*, 6, 21–40.

DRAPER, D. (1995): "Assessment and Propagation of Model Uncertainty," *Journal of the Royal Statistical Society. Series B (Methodological)*, 57, 45–97.

DRUCKER, H., C. J. C. BURGES, L. KAUFMAN, A. J. SMOLA, AND V. VAPNIK (1996): "Support Vector Regression Machines," in *Advances in Neural Information Processing Systems 9*, ed. by M. C. Mozer, M. I. Jordan, and T. Petsche, MIT Press, 155–161.

ELLIOTT, G., A. GARGANO, AND A. TIMMERMANN (2013): "Complete subset regressions," *Journal of Econometrics*, 177, 357 – 373.

ELLIOTT, G. AND A. TIMMERMANN (2016): *Economic Forecasting*, Princeton University Press.

FENG, G. AND J. HE (2019): "Factor Investing: Hierarchical Ensemble Learning," *Working Paper*.

FREUND, Y. AND R. E. SCHAPIRE (1997): "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, 55, 119 – 139.

GENRE, V., G. KENNY, A. MEYLER, AND A. TIMMERMANN (2013): "Combining expert forecasts: Can anything beat the simple average?" *International Journal of Forecasting*, 29, 108 – 121.

GIACOMINI, R. AND H. WHITE (2006): "Tests of Conditional Predictive Ability," *Econometrica*, 74, 1545–1578.

GU, S., B. KELLY, AND D. XIU (2020): "Empirical Asset Pricing via Machine Learning," *Review of Financial Studies*, 33, 2223–2273.

HANSEN, B. E. (2007): "Least Squares Model Averaging," *Econometrica*, 75, 1175–1189.

——— (2008): "Least-squares forecast averaging," *Journal of Econometrics*, 146, 342–350.

——— (2009): "Averaging Estimators for Regressions with A Possible Structural Break," *Econometric Theory*, 25, 1498–1514.

——— (2010): "Averaging Estimators for Autoregressions with A Near Unit Root," *Journal of Econometrics*, 158, 142 – 155, twenty Years of Cointegration.

HANSEN, B. E. AND J. S. RACINE (2012): "Jackknife model averaging," *Journal of Econometrics*, 167, 38–46.

HASTIE, T., R. TIBSHIRANI, AND J. FRIEDMAN (2009): *The Elements of Statistical Learning*, Springer Series in Statistics, New York, NY, USA: Springer New York Inc.

HOETING, J. A., D. MADIGAN, A. E. RAFTERY, AND C. T. VOLINSKY (1999): "Bayesian Model Averaging: A Tutorial," *Statistical Science*, 14, 382–401.

JAMES, G., D. WITTEN, T. HASTIE, AND R. TIBSHIRANI (2013): *An introduction to statistical learning*, vol. 112, Springer.

KIM, H. AND W.-Y. LOH (2003): "Classification Trees With Bivariate Linear Discriminant Node Models," *Journal of Computational and Graphical Statistics*, 12, 512–530.

LAHIRI, K. AND X. SHENG (2010): "Learning and heterogeneity in GDP and inflation forecasts," *International Journal of Forecasting*, 26, 265 – 292.

LEHRER, S. F. AND T. XIE (2018): "The Bigger Picture: Combining Econometrics with Analytics Improve Forecasts of Movie Success," *Working Paper*.

LI, Q. AND J. RACINE (2007): *Nonparametric Econometrics: Theory and Practice*, Princeton University Press, 1 ed.

LIU, Q. AND R. OKUI (2013): "Heteroskedasticity-robust C_p Model Averaging," *The Econometrics Journal*, 16, 463–472.

MAGNUS, J. R., O. POWELL, AND P. PRÜFER (2010): "A comparison of two model averaging techniques with an application to growth empirics," *Journal of Econometrics*, 154, 139 – 153.

MAKRIDAKIS, S. (1993): "Accuracy measures: theoretical and practical concerns," *International Journal of Forecasting*, 9, 527 – 529.

MULLAINATHAN, S. AND J. SPIESS (2017): "Machine Learning: An Applied Econometric Approach," *Journal of Economic Perspectives*, 31, 87–106.

PHILLIPS, L. D. AND M. C. PHILLIPS (1993): "Faciliated Work Groups: Theory and Practice," *The Journal of the Operational Research Society*, 44, 533–549.

QUINLAN, J. R. (1992): "Learning With Continuous Classes," World Scientific, 343–348.

RAFTERY, A. E., D. MADIGAN, AND J. A. HOETING (1997): "Bayesian Model Averaging for Linear Regression Models," *Journal of the American Statistical Association*, 92, 179–191.

RAPACH, D. E., J. K. STRAUSS, AND G. ZHOU (2009): "Out-of-Sample Equity Premium Prediction: Combination Forecasts and Links to the Real Economy," *The Review of Financial Studies*, 23, 821–862.

REID, D. J. (1968): "Combining Three Estimates of Gross Domestic Product," *Economica*, 35, 431–444.

SCHMITTLEIN, D. C., J. KIM, AND D. G. MORRISON (1990): "Combining Forecasts: Operational Adjustments to Theoretically Optimal Rules," *Management Science*, 36, 1044–1056.

SUYKENS, J. AND J. VANDEWALLE (1999): "Least Squares Support Vector Machine Classifiers," *Neural Processing Letters*, 9.

TIBSHIRANI, R. (1996): “Regression Shrinkage and Selection Via the Lasso,” *Journal of the Royal Statistical Society, Series B*, 58, 267–288.

TIMMERMANN, A. (2006): “Chapter 4 Forecast Combinations,” Elsevier, vol. 1 of *Handbook of Economic Forecasting*, 135 – 196.

ULLAH, A. AND H. WANG (2013): “Parametric and Nonparametric Frequentist Model Selection and Model Averaging,” *Econometrics*, 1, 157–179.

VAPNIK, V. N. (1996): *The Nature of Statistical Learning Theory*, New York, NY, USA: Springer-Verlag New York, Inc.

VENS, C. AND H. BLOCKEEL (2006): “A Simple Regression Based Heuristic for Learning Model Trees,” *Intell. Data Anal.*, 10, 215–236.

WAN, A. T., X. ZHANG, AND G. ZOU (2010): “Least Squares Model Averaging by Mallows Criterion,” *Journal of Econometrics*, 156, 277–283.

WANG, Y. AND I. H. WITTEN (1997): “Inducing Model Trees for Continuous Classes,” in *In Proc. of the 9th European Conf. on Machine Learning Poster Papers*, 128–137.

WIELAND, V., T. CWIK, G. J. MÜLLER, S. SCHMIDT, AND M. WOLTERS (2012): “A new comparative approach to macroeconomic modeling and policy analysis,” *Journal of Economic Behavior & Organization*, 83, 523 – 541, the Great Recession: motivation for rethinking paradigms in macroeconomic modeling.

XIE, T. (2015): “Prediction Model Averaging Estimator,” *Economics Letters*, 131, 5–8.

——— (2017): “Heteroscedasticity-robust model screening: A useful toolkit for model averaging in big data analytics,” *Economics Letters*, 151, 119–122.

XIE, T., J. YU, AND T. ZENG (2020): “Econometric Methods and Data Science Techniques: A Review of Two Strands of Literature and an Introduction to Hybrid Methods,” *Working Paper*.

ZHAO, S., X. ZHANG, AND Y. GAO (2016): “Model Averaging with Averaging Covariance Matrix,” *Economics Letters*, 145, 214 – 217.

ZOU, H. AND T. HASTIE (2005): “Regularization and variable selection via the elastic net,” *Journal of the royal statistical society: series B (statistical methodology)*, 67, 301–320.

Appendix

A Derivation of $P^{\text{LSSVR}}(X)$ in the case with an intercept

In line with De Brabanter, De Brabanter, Suykens, and De Moor (2011), if the formulation includes an intercept term β_0 such that

$$y_t = f(\boldsymbol{X}_t) + \epsilon_t = \beta_0 + \sum_{s=1}^{S} \beta_s h_s(\boldsymbol{X}_t) + \epsilon_t \quad \text{for } t = 1, ..., T, \tag{A1}$$

the optimization problem in LSSVR considers

$$\min_{\boldsymbol{\beta}} \; H(\boldsymbol{\beta}) = \sum_{t=1}^{T} (y_t - f(\boldsymbol{X}_t))^2 + \lambda \sum_{s=1}^{S} \beta_s^2$$

subject to (A1), where $\boldsymbol{\beta} = [\beta_0, ..., \beta_S]^\top = [\beta_0, \boldsymbol{\beta}_*^\top]^\top$. We can construct the Lagrangian equation

$$\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\alpha}) = H(\boldsymbol{\beta}) - \sum_{t=1}^{T} \alpha_t \left(\beta_0 + \sum_{s=1}^{S} \beta_s h_s(\boldsymbol{X}_t) - y_t \right),$$

where $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_T]^\top$ are Lagrange multipliers.

Taking the first-order conditions for optimization and substitute for $\boldsymbol{\beta}_*$, we obtain the following solution

$$\begin{bmatrix} 0 & \boldsymbol{\iota}^\top \\ \boldsymbol{\iota} & \boldsymbol{H}\boldsymbol{H}^\top + \gamma \boldsymbol{I}_T \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\boldsymbol{\alpha}} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix}, \tag{A2}$$

where $\boldsymbol{\iota} = [1, ..., 1]^\top$, $\boldsymbol{H}$ is the implicit basis matrix, and $\boldsymbol{H}\boldsymbol{H}^\top$ is the $T \times T$ kernel matrix with the $\{tt'^{th}\}$ element being the kernel function $K(\boldsymbol{X}_t, \boldsymbol{X}_{t'})$. For simplicity, we define

$$\boldsymbol{\Omega} \equiv (\boldsymbol{H}\boldsymbol{H}^\top + \lambda \boldsymbol{I}_T)^{-1}$$

and solve for $\hat{\beta}_0$ and $\hat{\boldsymbol{\alpha}}$ from (A2) such that

$$\begin{aligned} \hat{\boldsymbol{\alpha}} &= \boldsymbol{\Omega}(\boldsymbol{y} - \hat{\beta}_0 \boldsymbol{\iota}) \\ \hat{\beta}_0 &= \boldsymbol{\iota}^\top \boldsymbol{\Omega} \boldsymbol{y} / \boldsymbol{\iota}^\top \boldsymbol{\Omega} \boldsymbol{\iota}. \end{aligned}$$

The resulting LSSVR model then becomes

$$\begin{aligned} \hat{f}(\boldsymbol{X}) &= \boldsymbol{H}\boldsymbol{H}^\top \hat{\boldsymbol{\alpha}} + \hat{\beta}_0 \boldsymbol{\iota} \\ &= \boldsymbol{H}\boldsymbol{H}^\top \boldsymbol{\Omega}(\boldsymbol{y} - \hat{\beta}_0 \boldsymbol{\iota}) + \hat{\beta}_0 \boldsymbol{\iota} \\ &= \boldsymbol{H}\boldsymbol{H}^\top \boldsymbol{\Omega} \boldsymbol{y} + (\boldsymbol{\iota} - \boldsymbol{H}\boldsymbol{H}^\top \boldsymbol{\Omega} \boldsymbol{\iota}) \hat{\beta}_0 \end{aligned}$$

$$
\begin{aligned}
&= \left(\boldsymbol{H}\boldsymbol{H}^\top \boldsymbol{\Omega} + \frac{(\boldsymbol{\iota} - \boldsymbol{H}\boldsymbol{H}^\top \boldsymbol{\Omega}\boldsymbol{\iota})\boldsymbol{\iota}^\top \boldsymbol{\Omega}}{\boldsymbol{\iota}^\top \boldsymbol{\Omega}\boldsymbol{\iota}} \right) \boldsymbol{y}, \\
&= \boldsymbol{P}^{\text{LSSVR}}(\boldsymbol{X})\boldsymbol{y},
\end{aligned}
$$

where

$$
\boldsymbol{P}^{\text{LSSVR}}(\boldsymbol{X}) \equiv \boldsymbol{H}\boldsymbol{H}^\top \boldsymbol{\Omega} + \frac{(\boldsymbol{\iota} - \boldsymbol{H}\boldsymbol{H}^\top \boldsymbol{\Omega}\boldsymbol{\iota})\boldsymbol{\iota}^\top \boldsymbol{\Omega}}{\boldsymbol{\iota}^\top \boldsymbol{\Omega}\boldsymbol{\iota}}. \tag{A3}
$$

The no-intercept version of $\boldsymbol{P}^{\text{LSSVR}}(\boldsymbol{X})$ in (16) can be written as $\boldsymbol{\Omega}\boldsymbol{H}\boldsymbol{H}^\top$. Note that the $T \times T$ matrix $\boldsymbol{H}\boldsymbol{H}^\top \boldsymbol{\Omega}$ is symmetric, since

$$
\begin{aligned}
\boldsymbol{H}\boldsymbol{H}^\top (\boldsymbol{H}\boldsymbol{H}^\top + \lambda \boldsymbol{I}_T)^{-1} &= \boldsymbol{H}\boldsymbol{H}^\top \left((\boldsymbol{H}\boldsymbol{H}^\top)^{-1} - \lambda (\boldsymbol{H}\boldsymbol{H}^\top)^{-1} (\boldsymbol{H}\boldsymbol{H}^\top + \lambda \boldsymbol{I}_T)^{-1} \right) \\
&= \boldsymbol{I}_T - \lambda (\boldsymbol{H}\boldsymbol{H}^\top + \lambda \boldsymbol{I}_T)^{-1}
\end{aligned}
$$

following the Woodbury matrix identity. Therefore, the no-intercept $\boldsymbol{P}^{\text{LSSVR}}(\boldsymbol{X})$ is a special case of (A3) that does not include the second term on the right-hand-side of (A3).

B Data polishing procedure

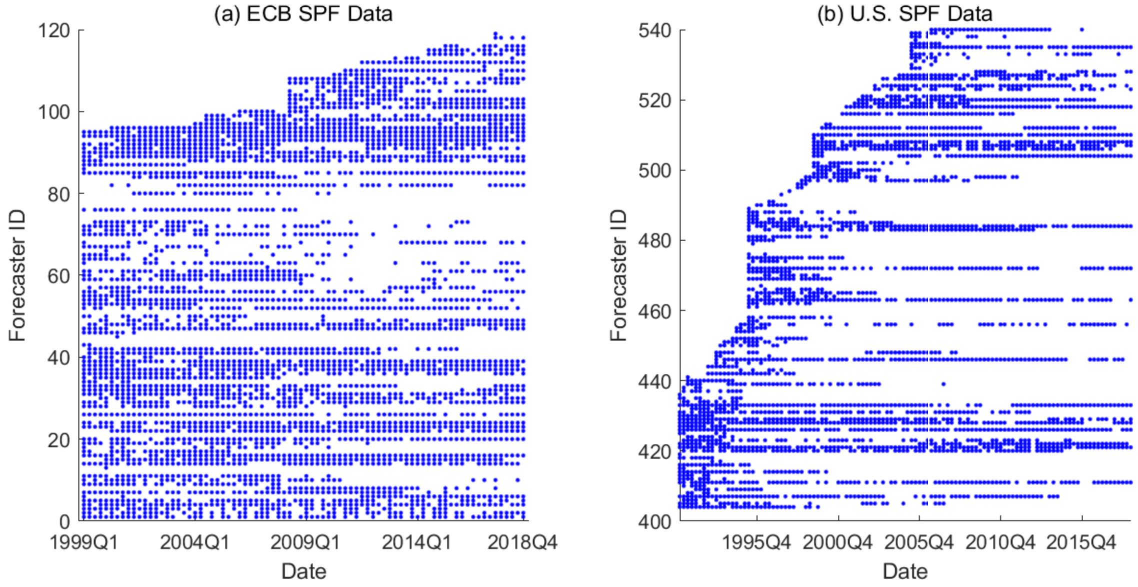
The Survey of Professional Forecasters (SPF) conducted by the European Central Bank (ECB) and the U.S. Federal Reserve Bank of Philadelphia collects the forecasters' views on macroeconomic and financial indicators in the respective region or country. However, a specific forecaster may not consistently submit a survey response throughout the sample period. Figure A1 describes the entries and exits of individual forecasters across the survey dates for the ECB and U.S. SPF data. A blue dot is marked for a specific forecaster (labeled in the vertical axis) if he or she submitted a survey response and a blank space indicates otherwise.

Both data clearly exhibit severe sparsity. To avoid the issues caused by missing observations, we follow Genre et al. (2013) to first remove irregular respondents if he or she misses more than near 50% of the observations. We narrow down to 30 qualified forecasters for the ECB SPF data and 21 qualified forecasters for the U.S. SPF data. Then the missing observations for each forecaster i are filled using the following approach proposed in Genre et al. (2013)

$$
\hat{y}_{it} - \bar{y}_t = \beta_t (\hat{y}_{i,t-1} - \bar{y}_{t-1}) + \epsilon_{it},
$$

where $\hat{y}_{it}$ is the prediction of forecaster i at time t and $\bar{y}_t$ is the equally weighted prediction over all forecasters at time t. For each forecaster i, we compute for $\hat{\beta}_t$ with all available observations and then fill in the gaps by a simple AR(1) process.

Figure A1: An illustration of the entries and exits of individual forecasters



C Construction of the candidate model set by GMS

The candidate model set is crucial for any model averaging method in practice. A widely-adopted approach to constructing the candidate model set is through a full combination of all the p predictors, which leads to 2^p candidate models. One obvious drawback of this approach is that the total number of candidate models increases exponentially with p. In the case of our forecasting exercise, the above approach leads to 1,073,741,824 models, which are way too many to handle.

Inspired by Xie (2017),[12] we propose constructing the candidate model set through an forward iterative procedure which adds one predictor at a time according to certain criterion. This approach is termed the generalized model screening (GMS) method. The computational algorithm of GMS for a given method is summarized in the sequel.

1. We pick an initial model, denoted by $\mathbb{M}_{(0)}$, which can be a null model that includes no variables, or can be a model consisting of certain predictors of interest.

2. Each time we add one of the $q_{(0)}$ remaining regressors to $\mathbb{M}_{(0)}$, which generates $q_{(0)}$ candidate models. We then examine each candidate model by one of the following

[12] Xie (2017) proposed a homoskedasticity-efficient model screening (HEMS) method and a heteroskedasticity-robust model screening method (HRMS) to construct candidate models for the LS model averaging with a large number of predictors. Both HEMS and HRMS are forward iterative procedures which add one predictor at a time according to certain criteria.

criteria:

$$\begin{aligned}\text{Homoskedasticity} &: \|\boldsymbol{y}-\boldsymbol{P}_{(s)}\boldsymbol{y}\|^2+2\hat{\sigma}^2_{(s)}\sum_{t=1}^{T}P^{(s)}_{tt},\\ \text{Heteroskedasticity} &: \left\|\boldsymbol{y}-\boldsymbol{P}_{(s)}\boldsymbol{y}\right\|^2+2\sum_{t=1}^{T}(\hat{\epsilon}^{(s)}_t)^2P^{(s)}_{tt},\end{aligned}$$

for $s = 1, ..., q_{(0)}$, where $\boldsymbol{P}_{(s)}$ stands for the projection matrix $\boldsymbol{P}(\boldsymbol{x}, \boldsymbol{X})$ of the s^{th} candidate model, $\hat{\epsilon}^{(s)}_t$ is the t^{th} element of the residual with $\hat{\sigma}^2_{(s)}$ being the variance of estimated error terms (for homoskedasticity), and $P^{(s)}_{tt}$ represents the t^{th} diagonal term in $\boldsymbol{P}_{(s)}$.

3. We select the model denoted by $\mathbb{M}_{(1)}$ that yields the lowest value of the criterion under homoskedasticity or heteroskedasticity. Model $\mathbb{M}_{(1)}$ is taken as the initial model for the next round.

4. We repeat steps (2) to (3) iteratively until we draw the full model that consists of all q variables. We construct our candidate model set by incorporating the initial model $\mathbb{M}_{(0)}$ (if not null), all candidate models from step (2) and the full model.

The GMS method adds one and only one variable to the model of the previous step each time. Therefore, if there are q variables in total and our initial model $\mathbb{M}_{(0)}$ includes q_0 variables, we end up with only $(q - q_0 + 1)$ models that are nested in sequence. This number is much smaller than 2^q, especially for a large value of q.

D Results under a tainted candidate model set

In this main text, we have showed that both SAML and MAML perform well under screened candidate model sets by GMS. In this section, we examine the performance sensitivity of SAML and MAML to the composition of candidate model sets. The one-year-ahead HICP inflation forecast is illustrated as as an example. The original screened model set by GMS is denoted as $\mathbb{M}_0$.

A "tainted" candidate model set $\mathbb{M}_1$ is created partially from $\mathbb{M}_0$. This is achieved by having half of the models in $\mathbb{M}_1$ from $\mathbb{M}_0$ and replacing the other half with individual models that incorporate only one predictor. In this way, not all the models in $\mathbb{M}_1$ produce fair forecasts. We predict the one-year-ahead HICP inflation using SAML and MAML under $\mathbb{M}_0$ and $\mathbb{M}_1$ and the results are shown in panels A and B of Table A1, respectively. The results in panel A are identical to those in Table 3, which are reproduced here for comparison convenience.

Table A1: Forecast combination machine learning methods under different model sets

Method	MSFE	MAFE	SDFE	Pseudo R^2
Benchmark	0.6533	0.7076	0.8083	0.1393
Panel A: Results under $\mathbb{M}_0$				
BAG_1^{SA}	0.4655	0.5510	0.6823	0.3867
BAG_2^{SA}	0.4607	0.5743	0.6788	0.3931
RF_1^{SA}	0.4677	0.5624	0.6839	0.3839
RF_2^{SA}	0.4656	0.5633	0.6823	0.3867
$\text{LSSVR}_{\text{G1}}^{\text{SA}}$	0.3767	0.5316	0.6138	0.5037
$\text{LSSVR}_{\text{G2}}^{\text{SA}}$	0.3816	0.5301	0.6177	0.4974
BAG_1^{MA}	0.4386	0.5247	0.6623	0.4222
BAG_2^{MA}	0.4751	0.5803	0.6893	0.3741
RF_1^{MA}	0.4529	0.5447	0.6730	0.4033
RF_2^{MA}	0.4748	0.5568	0.6891	0.3745
$\text{LSSVR}_{\text{G1}}^{\text{MA}}$	0.4505	0.5840	0.6712	0.4065
$\text{LSSVR}_{\text{G2}}^{\text{MA}}$	0.4451	0.5804	0.6672	0.4136
Panel B: Results under $\mathbb{M}_1$				
BAG_1^{SA}	0.5792	0.6216	0.7611	0.2370
BAG_2^{SA}	0.5815	0.6233	0.7626	0.2339
RF_1^{SA}	0.5692	0.6206	0.7544	0.2502
RF_2^{SA}	0.5718	0.6235	0.7562	0.2467
$\text{LSSVR}_{\text{G1}}^{\text{SA}}$	0.5823	0.6372	0.7631	0.2329
$\text{LSSVR}_{\text{G2}}^{\text{SA}}$	0.5823	0.6372	0.7631	0.2329
BAG_1^{MA}	0.5332	0.6173	0.7302	0.2976
BAG_2^{MA}	0.5295	0.6100	0.7277	0.3024
RF_1^{MA}	0.5101	0.6105	0.7142	0.3280
RF_2^{MA}	0.4790	0.5818	0.6877	0.3769
$\text{LSSVR}_{\text{G1}}^{\text{MA}}$	0.4840	0.6004	0.6957	0.3625
$\text{LSSVR}_{\text{G2}}^{\text{MA}}$	0.4840	0.6004	0.6957	0.3625

This table reports the out-of-sample results for predicting the one-year-ahead HICP inflation using various methods shown in the first column. The results are based on the HICP inflation data ranging from 2000Q1 to 2019Q4. Panel A presents the outcomes for the original screened model set, while panel B contains the results for the "tainted" candidate model set. We use a rolling window of 40 observations to estimate the forecasts.

Both SAML and MAML outperform the benchmark under $\mathbb{M}_0$ and $\mathbb{M}_1$, although the results by all methods deteriorate under $\mathbb{M}_1$. It can be seen that the SAML methods are quite sensitive to the choice of candidate model sets. For example, the MSFE of the best performing SAML method under $\mathbb{M}_0$, $\text{LSSVR}_{\text{G1}}^{\text{SA}}$, increases by 54.58% under $\mathbb{M}_1$. This is not a surprise since $\mathbb{M}_1$ incorporates many poor-performing candidate models and the effect of bad models remains due to equal weighting.

In contrast, the estimated weights by MAML are determined by the performance of individual candidate models evaluated by Mallows-type criteria. Thus the MAML methods automatically assign lower weights to bad-performing candidate models compared to good-performing ones. This explains why the MAML methods are less sensitive to the composition of model sets. For instance, the MSFE of the best performing MAML method under $\mathbb{M}_0$, $\text{LSSVR}^{\text{MA}}_{\text{G2}}$, increases by 8.74% only under $\mathbb{M}_1$. The above results signify the necessity of first implementing reliable model screening techniques for forecast combination, which is especially so for the SAML methods.

E Results under alternative values of tuning parameters

In this section, we present the empirical results under alternative values of tuning parameters using the HICP inflation forecasts $(h = 4)$ as an example. We change the penalty coefficient, the number of selected predictors for RF-type methods, and the hyperparameters on kernels to the following values.

1. $\lambda = 1$ for LASSO, RIDGE, EN, SVRs, LSSVRs;
2. The number of selected predictors is set to $\lfloor p/2 \rfloor$ for all the RF-type methods;
3. $\sigma_x = 5$ for the Gaussian kernel;
4. $d = 100$, $p = 5$ for the polynomial kernel;

The results of forecast accuracy comparison are shown in Table A2. Our results are qualitatively unchanged.

Table A2: Forecast Accuracy Comparison HICP Inflation $(h = 4)$ under Different Tuning Parameters

Method	MSFE	MAFE	SDFE	Pseudo R^2
Benchmark	0.6533	0.7076	0.8083	0.1393
EN	0.8810	0.7569	0.9386	-0.1606
CSR_{15}	0.6767	0.6884	0.8226	0.1085
BAG	0.5997	0.6524	0.7744	0.2101
RF	0.5778	0.6482	0.7601	0.2388
LSSVR_G	0.4966	0.6086	0.7047	0.3458
BAG_1^{SA}	0.4726	0.5544	0.6875	0.3774
BAG_2^{SA}	0.4868	0.5667	0.6977	0.3587
RF_1^{SA}	0.4825	0.5688	0.6947	0.3643
RF_2^{SA}	0.4562	0.5618	0.6754	0.3991
$\text{LSSVR}_{\text{G1}}^{\text{SA}}$	0.4558	0.5690	0.6751	0.3996
$\text{LSSVR}_{\text{G2}}^{\text{SA}}$	0.4585	0.5717	0.6771	0.3960
BAG_1^{MA}	0.4656	**0.5258**	0.6824	0.3866
BAG_2^{MA}	0.4759	0.5482	0.6898	0.3731
RF_1^{MA}	0.4792	0.5638	0.6923	0.3687
RF_2^{MA}	**0.4417**	0.5320	**0.6646**	**0.4182**
$\text{LSSVR}_{\text{G1}}^{\text{MA}}$	0.4870	0.6116	0.6978	0.3585
$\text{LSSVR}_{\text{G2}}^{\text{MA}}$	0.4616	0.5936	0.6794	0.3919

This table reports the out-of-sample results for predicting the HICP one-year-ahead using different methods. The best result under each criterion is highlighted in the bold face.