Combining forecasts: Can machines beat the average?*

Tyler Pike[†]
Federal Reserve Board

Francisco Vazquez-Grande
Federal Reserve Board

October 21, 2020

Abstract

Yes. This paper documents the benefits of combining forecasts using weights built with non-linear models. We introduce our tree-based forecast combinations and compare them with benchmark equal weight combination as well as other nonlinear forecast weights. We find that nonlinear models can improve consistently upon the equal weight alternative–breaking the so-called "forecast combination puzzle"–and that our proposed methods compete well with other nonlinear methods.

Keywords: forecast combination; machine learning; random forest

[*] The views presented here are solely those of the authors and do not necessarily represent those of Board of Governors of the Federal Reserve or of the Federal Reserve System.

[†] Corresponding author. (Tyler.J.Pike@frb.gov). 1801 K St NW, Washington, DC 20006

1 Introduction

There exist a vast literature on optimal forecast combination (see for example Bates and Granger, 1969; Clemen, 1989; Timmermann, 2006), that propose theoretically optimal forecast weights for any common (symmetric or asymmetric) loss function. These optimal weights depend nonlinearly on past forecast errors. But in practice uniform weights are dominant, and empirical studies have shown that an equally weighted average across forecasts tends to outperform approaches based on estimated optimal weights. This is the so-called "forecast combination puzzle" (see Stock and Watson, 2004), which is usually explained by the importance of parameter estimation error in the combination of weights.

Elliott and Timmermann (2004) derive the optimal (population) constant and combination weights in a linear forecast combination model, for a general symmetric loss function, which in the special case of forecast errors having the same variance and pair-wise correlation the equally-weighted forecast combination is optimal (see Timmermann (2006)). However, and given that this assumptions are unlikely to hold in most practical applications, the study and construction of nonlinear weights is of importance.

In this paper we study the performance of nonlinear forecast combinations. We introduce two tree-based methods and compare them to uniform weights and median forecast combination, as well as to the nonlinear LASSO of Conflitti et al. (2015) and Diebold and Shin (2019). We show the effectiveness of tree-based combination techniques when the underlying forecast generating processes are available. To comprehensively compare these weight formation techniques, we conduct two out-of-sample forecasting exercises. First we combine one-year ahead forecasts from the European Survey of Professional Forecasters, a popular data source in the forecast combination literature. Second, we estimate and combine our own pool forecasts, using basic time series models, for a long history of monthly United States macroeconomic data. These experiments show that the nonlinear models we introduce can outperform uniformly-weighted forecast combinations breaking the "forecast combination puzzle."

There has been relatively little work on using nonlinear methods for forecast combinations in the academic literature. Historical exceptions include Donaldson and Kamstra (1996a) who use simple neural networks to combine volatility forecasts and show success and Deutsch et al. (1994) who use regime switching to combine forecasts with success. While contemporary exceptions are Montero-Manso et al. (2018) who demonstrate success using a modern nonlinear machine learning technique for forecast combinations in the M4 competition[1], and Diebold and Shin (2019) who develop both the partial egalitarian LASSO and the piece-wise linear average-best forecast combination techniques.

We will proceed in five sections: section 2 give details on the forecast combination techniques, section 3 describes our empirical exercises, section 4 presents our results, and section 5 concludes.

2 Forecast combination methods

We will compare the classical uniform weights and median forecast combination techniques to the nonlinear LASSO methods employed by Conflitti et al. (2015) and Diebold and Shin (2019), as well as, two tree-based methods that we introduce to the forecast combination literature.

2.1 Classical methods

Our baseline forecast combination technique is the canonical standard, uniform weights. We will also use the median forecast as a robustness check on the efficacy of our nonlinear combination techniques. The median forecast is used as a robust variation of the mean and has been shown to outperform uniform weights when noisy forecasts are present (see for example Genre et al. (2013)).

2.2 LASSO methods

We will first compare uniform weights to the well-known LASSO method, as in Conflitti et al. (2015), as well as, the partial-egalitarian LASSO (peLASSO) and average-best combinations of Diebold and Shin (2019).

[1] However these authors are attempting to create a generalized forecasting engine for use on 10,000 heterogeneous time series, and use a derived set of time series characteristics as an additional set of exogenous variables for determining the forecast combination weights, breaking from the tradition of using only forecast errors.

The weights of a LASSO regression are defined as:

$$\hat{\beta}_{LASSO} = \arg\min_{\beta} \left(\frac{1}{2} \sum_{t=1}^{T} \left(y_t - \sum_{i=1}^{K} \beta_i f_{it} \right)^2 + \lambda \sum_{i=1}^{K} |\beta_i| \right)$$

where t is the time index, k is the forecast index, and λ is the penalty factor. A thorough textbook discussion of the general method and its variants can be found in Hastie et al. (2009). In the forecast combination literature, Conflitti et al. (2015) find that the LASSO, with weights normalized to unity, can achieve highly accurate results when combining a survey of forecasts. However, we find that not imposing a unity restriction on the weights yields better results, and report the unconstrained LASSO in our analysis.

The peLASSO is best described as an extension of the traditional LASSO designed specifically for forecast combinations. The weights estimated by the peLASSO are defined as:

$$\hat{\beta}_{peLASSO} = \arg\min_{\beta} \left(\sum_{t=1}^{T} \left(y_t - \sum_{i=1}^{K} \beta_i f_{it} \right)^2 + \lambda_1 \sum_{i=1}^{K} |\beta_i| + \lambda_2 \sum_{i=1}^{K} \left| \beta_i - \frac{1}{p(\beta)} \right| \right)$$

where t is the time index, k is the forecast index, and λ are penalty factors. The method can be thought of in two steps, first it selects the relevant forecasts by shrinking weights to zero, and then regularizes the surviving weights by shrinking them to uniformity. As the peLASSO is sensitive to the choice of hyperparamters λ, the authors also propose a simple method which emulates the intuition of the peLASSO, average-best combination. The average-best combination is constructed by reporting the mean forecast of the n-best, as measured by some loss function[2], models at any given point in time[3].

2.3 Tree methods

We will use two tree-based methods to combine forecasts, the boosted tree and the random forest. A single decision tree can be viewed as a function mapping each element of the domain to a neighborhood in the set of dependent variables, and then returning the average of that neighborhood. A bagged collection of trees, by extension, may be

[2] We will use the RMSE loss function.

[3] We use the entire history of errors in calculating the average best, so as to directly compare weighting methods, and not confound our results by also changing the window of observations.

thought of as a robust average of sampled averages from slightly varying neighborhoods. As a result, tree-based methods are obvious candidates for forecast combinations as they are designed to find robust averages, just as the mean and LASSO techniques do. Further, tree-based methods perform well "off-the-shelf" with little tuning, can easily handle nonlinear and sparse data, have intuitively evaluated variables of importance metrics, and can be extended to produce confidence intervals if desired (see Athey and Imbens (2019) for a discussion of tree-based methods for economics or Hastie et al. (2009) for a comprehensive statistically-oriented textbook treatment).

A single decision tree, first introduced by Breiman et al. (1984), is characterized by a collection of partitions called "decision nodes." Given a sample of outcomes Y_t and independent variables (also called "covariates") X_{tk} for $k = 1, \ldots, K$ and $t = 1, \ldots, T$, the algorithm starts by choosing a variable and a threshold to split the sample into two subsamples, according to whether the values of the chosen variable fall above or below the threshold. The estimated outcome of the tree for the observations on each subsample is then set to the average outcome for each of the subsamples. The algorithm proceeds by creating more splits sequentially on each of the subsamples on a single independent variable at a time. At each step, the algorithm selects the covariate and threshold that maximize the log-likelihood for the resulting subsamples. The size of the tree (the number of splits) represents the complexity of the tree and is a regularization parameter to be chosen to improve accuracy and avoid overfitting.

Random forests, first introduced by Breiman (2001), were proposed to improve the out-of-sample performance of the tree algorithm. By construction, trees provide a discontinuous predictive function: as values of the covariate cross thresholds the tree jumps to new predictive values. Random forests induce smoothness by averaging over a large number of trees. These trees differ from each other in two ways. First, each tree is based not on the original sample, but on a random subsample of the data, that selects randomly a subset of independent variables. Second, the splits at each stage are not optimized over all possible covariates, but over the random subset of the covariates that changes with every split. These two modifications lead to sufficient variation in the trees so that the average is relatively smooth and more importantly, has better predictive power than a single tree. The maximum number of splits of trees and the size of subsamples are the regularization parameter to be chosen in this algorithm[4].

[4] See Appendix A for the formal random forest algorithm.

Boosting is a general purpose technique to improve the performance of simple ML methods; here we apply it to trees, as in Friedman (2002). We start by forming predictions, based on outcomes Y_t and covariates X_{tk} with a very simple tree with at most two splits (this is our base learner). This algorithm does not have very high predictive performance as it only uses at most two of the many possible features of the independent variables. Boosting improves this base learner in the following way. Compute for all observations in the sample the residual from the prediction based on the simple tree. Now we apply the same base learner to the residuals as the outcome of interest (and with the same set of original covariates X_{tk}). The boosted prediction is then the sum of these two base learners. The algorithm uses the boosted tree to calculate a new residual, and then fits the base learner with covariates X_{tk} to the new residual. We repeat this process B times, and get a prediction based on summing the basic model predictions of updated residuals[5].

We use only use the most basic version of these tree-based methods, establishing a floor for the performance tree algorithms. In practice, one can use more accurate and sophisticated iterations, such as the local linear forest Friedberg et al. (2018) or the XGboost implementation of the boosted tree Chen and Guestrin (2016).

As both tree-based methods and LASSO-based methods require choosing hyperparameters, we do so via five-fold cross validation. There is an element of randomness to estimating the tree techniques so we replicate them 10 and 100 times in our first and second exercises, respectively, and report the mean to reduce noise in the analysis. Producing 100 forecasts at each point also has the advantages of producing a forecast distribution from which model uncertainty can be calculated and non-first order moments can be exploited, but we leave these strengths to be studied in future work.

2.4 Feature-based forecast model averaging

The last method we will test is the feature-based forecast model averaging (FFORMA) tree-based machine of Montero-Manso et al. (2018). This machine was originally built and tested in the M4 forecasting competition Makridakis et al. (2018), in which the goal was to minimize forecast point and interval errors across 100,000 demographic, economic, and financial series. As a result, FFORMA is unique in our list, as it focuses on optimizing

[5] See Appendix A for the formal tree-based stochastic gradient boosting machine (boosted tree) algorithm.

performance over many series and uses cross learning to do so. That is, the FFORMA algorithm is trained across a large set of potential forecast target time series, as opposed to just one as our other techniques do. Given the necessity to learn over many similar time series, we will use a random subsample of 1,000 time series drawn from the M4 competition's 100,000 time series to train our FFORMA machine.

The formal FFORMA algorithm may be found in Appendix A, but we will also summarize the procedure here: First, a pool of forecasts are generated for each time series in a reference set of time series. Second, the forecast errors from each forecast method in the pool of forecasts, per time series in the reference set, are generated. Third, a collection of characteristics describing the time series in the reference set (e.g. strength of trend, spikiness, ect.) are calculated per time series. Fourth, the XGboost implementation of the boosted tree algorithm is trained to take in the time series characteristics as input and output the forecast combination weights for the pool of forecasts, per time series in the reference set, which minimizes the forecast errors per time series calculated in the second step.

3 Empirical exercises

We conduct two out-of-sample forecasting exercise to evaluate the real-time accuracy of several model averaging techniques, and in particular to test whether the so-called "forecast combination puzzle" (see Stock and Watson, 2004) is present when weighting models using nonlinear algorithms. Our estimation is conducted in real time, so our results can be extensible to relevant forecasting exercises. First we combine one-year ahead forecasts from the European Survey of Professional Forecasters, a popular data source in the forecast combination literature. Second, we estimate and combine our own pool forecasts, using basic time series models, for a long history of monthly United States macroeconomic data.

3.1 European survey of professional forecasters

We begin by combining one-year ahead forecasts for the Harmonized Index of Consumer Prices (HICP) growth rate and Real GDP growth rate from the European Survey of Professional Forecasters. The survey was begun at the outset of the European monetary union and has been collected quarterly since 1999. Representing a group of opaque and heterogeneous forecasts, the survey has been used extensively in forecast combinations literature (see for example Genre et al., 2013; Conflitti et al., 2015; Diebold and Shin, 2019).

Following the literature, we use data starting in 2000:Q1 and begin estimating our recursively updated forecast combinations in 2005:Q1. Further, as our techniques require balanced panels, we follow Conflitti et al. (2015) and use only forecasters with at least 5 submissions in a given information set during our out-of-sample exercise. Additionally, we fill in omitted entries by using the forecasters individual historical average entry. Forecasts run through 2019:Q4. We then use data as of Jan. 2020 to evaluate forecast results[6].

As we do not control the forecasting functions (i.e. forecasters) in the survey, we are not able to estimate the FFORMA machine due to an insufficient number of time series in the reference set, and therefore exclude it from this exercise.

3.2 United States macroeconomic data

While the European Survey of Professional Forecasters is the canonical choice for evaluating forecast combination techniques, it has a limited history and number of observations. To evaluate the efficacy of our forecast combination techniques in a setting with a longer history and larger number of observations, we forecast two important monthly US macro series, payroll employment and industrial production. However, instead of using a survey of forecasts, we use a pool of nine well known time-series forecasting models recently used with in the M4 Forecasting competition by Montero-Manso et al. (2018). Although, as the purpose of this paper is forecast combinations, details of the individual time series models can be found in the appendix B.

Our exercise uses first-read vintages from the Philadelphia FRB's Real-Time Data Research Center. Industrial production is measured as the period-over-period growth rate, while payroll employment is measured as the annualized period-over-period growth rate. Further, while the European SPF data only begins in 2000:Q1, payroll employment data begins in 1964:11 while industrial production data begins 1962:10. Our time series forecast begin in 1970:01, and forecast combinations begin 1975:01.

As a consequence of combining a pool of forecasts that we generate, we are able to test forecast combination techniques at various horizons. We explore this possible dimension

[6] Genre et al. (2013) do not find that vintage matters qualitatively

of differing accuracy by testing forecasts for payroll employment and industrial production at the one-, six-, ten-, and 24-month horizons.

4 Results

We conduct two exercises to evaluate the ability of nonlinear forecast combination techniques. Here we document results that suggest nonlinear models currently existing in forecast combination literature, and tree-based techniques we introduce, can outperform uniformly-weighted forecast combinations.

4.1 European Survey of Professional Forecasters

In our first exercise, forecasting with the European SPF, we find that nonlinear methods can outperform uniform weights and the median[7].

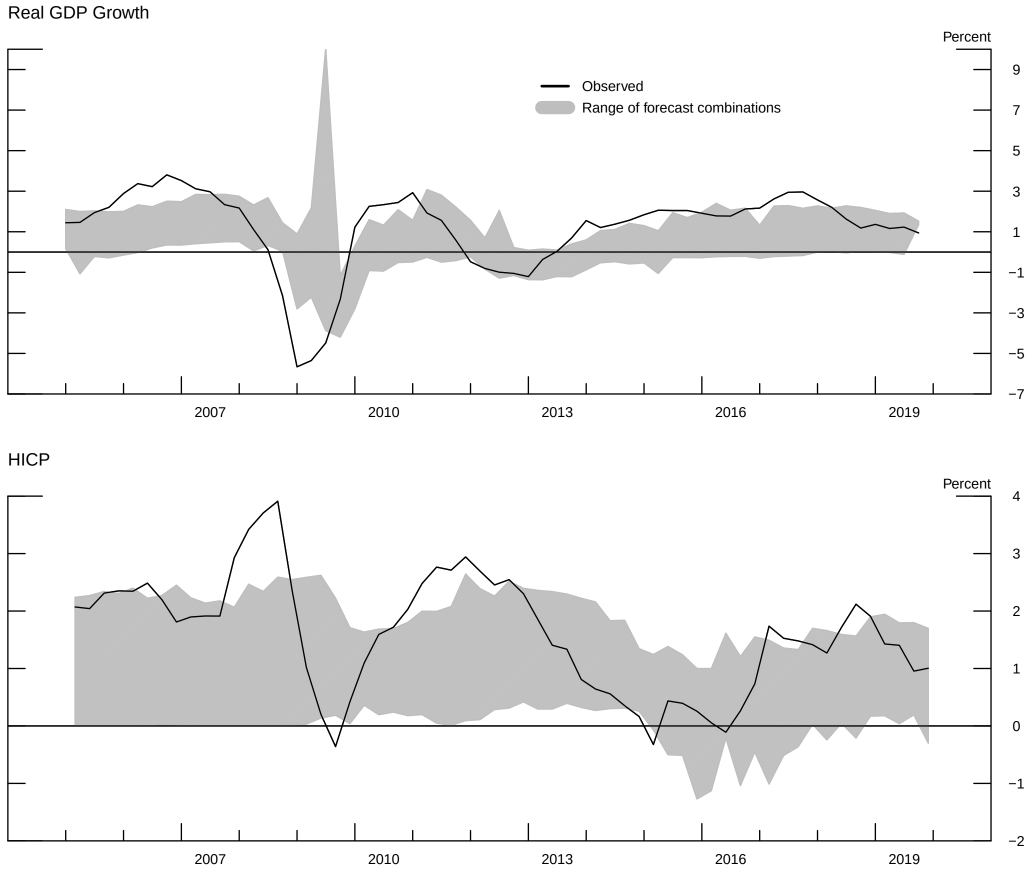
Figure 1 shows the one-year ahead forecast combinations for both changes in United States payroll employment and industrial production.

Table 1 reports the relative root mean squared error (RMSE) and relative mean squared error (MSE) of forecast combination techniques, where relative denotes a ratio of forecast combination errors to uniform weight errors. By construction, a relative RMSE or MSE less than one signals a forecast combination technique superior to using uniform weights. We use both RMSE and MSE, as together they help us detect the presence of outliers in the errors. Table 1 also reports the Diebold and Mariano (1995) statistic p-value, testing the statistical significance of the difference in forecast combination errors.

We find that the LASSO and peLASSO do not perform well, in terms of either RMSE or MSE. However, the LASSO-related N-best methods perform extremely well, with both RMSE and MSE being nearly eighty percent of those from uniform weights for forecasting HICP. Further, the N-best methods decrease forecast errors for the HICP at a statistically significant level, according to the Diebold-Mariano statistic. The N-best also achieve considerable success forecasting real GDP, reaching a relative MSE of 0.76. However, the forecast improvements generated by the N-best method for real GDP are not statistically significant according to the Deibold-Mariano statistic. We additionally find that the tree-methods do well, producing relative MSE's less than one, but they are not statistically

[7] The following results are robust to excluding the financial crisis

Figure 1: European SPF forecast combinations



Quarterly, one-year-ahead, forecast combinations are shown from 2005 through 2019. The thick black line denotes the observed series and the grey shaded region identifies the range of forecast combination point estimates.

significant. Although, it is notable that the boosted tree has a relative MSE of 0.78 for forecasting real GDP, second only to N1.

4.2 United States macroeconomic data

In our second exercise, forecasting United States macroeconomic data, we find further evidence that nonlinear methods can outperform uniform weights and the median forecast

Table 1: European Survey of Professional Forecasters combination results

Combination technique	HICP		Real GDP	
	Relative MSE	Diebold-Mariano	Relative MSE	Diebold-Mariano
Mean Forecast	1.00	1.00	1.00	1.00
Median Forecast	1.00	0.21	1.00	0.05°
Lasso	2.05	0.99	1.64	1.00
peLasso	2.27	1.00	2.09	0.98
N1	0.81	0.04°	0.88	0.15
N2	0.83	0.04°	0.90	0.15
N3	0.82	0.03°	0.91	0.14
N4	0.83	0.02°	0.91	0.13
Random Forest	1.02	0.27	1.06	0.38
Boosted Tree	0.97	0.13	0.96	0.21

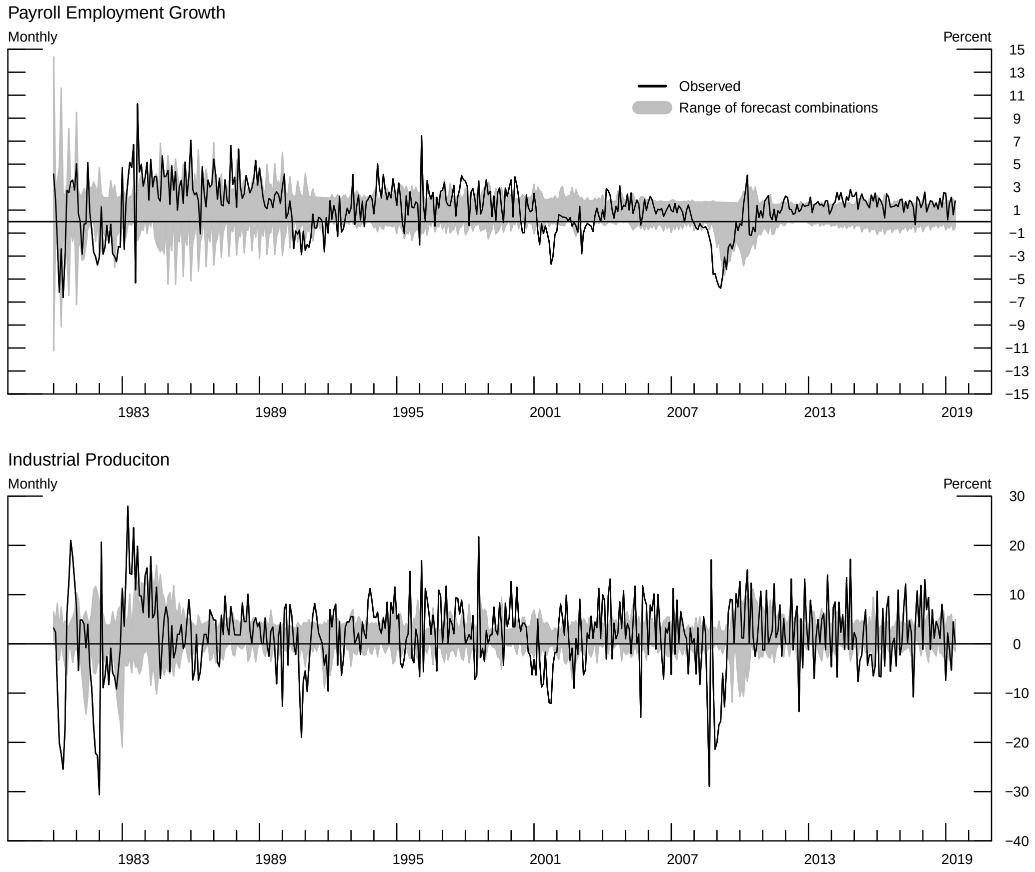
Notes: columns (2) and (4) report the RMSE of the combination method specified in column (1) divided by the RMSE the average survey forecast. A ratio less than one denotes the combination outperforming the uniform forecast. Columns (3) and (5) report the Diebold and Mariano (1995) p-value with the null hypothesis that the mean forecast is better than the model. $^{\odot}$, $^{\circ}$, * denote DM statistics significant at the ten-percent, five-percent, and one-percent confidence level, respectively.

combination.

Figure 2 shows the one-year ahead forecast combinations for both changes in United States payroll employment and industrial production.

Table 2 reports the relative RMSE of forecast combination techniques across all four horizons for both changes in payroll employment and industrial production. The time series characteristic driven FFORMA the clear standout star of this exercise. We find that it strictly dominates all other combination techniques across all but one series-horizon pair. In fact, FFORMA averages an error ratio of 0.58 across all eight exercises, achieving a minimum of 0.33 for one-year ahead industrial production forecasts. The key difference between FFORMA and the other combination methods is that FFORMA aims to provide the best combination weights across many time series, appropriatley regularizing the weights to prevent overfitting to one time series, reducing noise. Table 2 details results providing strong evidence that reducing noise through regularizing over many time series

Figure 2: United States macroeconomic series forecast combinations



Monthly, one-year-ahead, forecast combinations are shown from 1980 through 2019. The thick black line denotes the observed series and the grey shaded region identifies the range of forecast combination point estimates.

allows one to strictly dominate uniform combination weights.

We further find that for all but one monthly variable-horizon pair, both simple tree-based methods outperform both mean and median forecast combination techniques. Particularly of note is the fact that for both tree-methods, across both payroll employment and industrial production, there appears to be a positive correlation between relative performance and horizon length. The random forest and boosted tree achieve RMSE ratios of 0.67 and 0.72 respectively when forecasting payroll employment two years ahead, the

Table 2: United States macroeconomic data forecast combination results

Combination Technique	Employment				Industrial Production			
	H = 1	H = 6	H = 12	H = 24	H = 1	H = 6	H = 12	H = 24
Median Forecast	$0.91^{\star}$	$0.90^{\star}$	$0.90^{\star}$	$0.86^{\star}$	1.00	0.99	0.96	$0.96^{\star}$
peLasso	1.49	1.33	1.19	1.08	1.10	1.09	1.00	1.00
Lasso	1.49	1.33	1.32	1.24	1.10	1.09	1.00	1.00
N1	$0.94^{\odot}$	0.99	0.94	$0.82^{\star}$	1.03	0.99	0.93°	$0.92^{\star}$
N2	$0.93^{\odot}$	$0.94^{\odot}$	0.92°	$0.82^{\star}$	1.02	0.99	$0.93^{\star}$	$0.92^{\star}$
N3	$0.93^{\star}$	$0.92^{\star}$	$0.90^{\star}$	$0.82^{\star}$	1.01	0.99	0.93°	$0.92^{\star}$
N4	$0.92^{\star}$	$0.90^{\star}$	$0.90^{\star}$	$0.82^{\star}$	1.00	0.98	0.93°	$0.92^{\star}$
Random Forest	0.96	$0.83^{\star}$	$0.80^{\star}$	$0.67^{\star}$	0.99	0.96	0.91°	0.93°
Boosted Tree	0.94°	$0.86^{\star}$	0.83°	$0.72^{\star}$	0.98°	0.96	$0.92^{\odot}$	0.92°
FFORMA	1.19	$0.73^{\odot}$	0.64°	$0.48^{\star}$	$0.58^{\star}$	$0.37^{\star}$	$0.33^{\star}$	$0.37^{\star}$

Notes: forecast performance is reported as the ratio of the given forecast combination method's RMSE to the mean forecast's RMSE, such that a ratio less than one signals a forecast performance better than using uniform weights. $^{\odot}$, $^{\circ}$, * denote Diebold and Mariano (1995) statistics significant at the ten-percent, five-percent, and one-percent confidence level, respectively, testing that the given forecast combination technique improves upon using uniform weights.

two greatest error improvements of the exercise. Almost all tree-based errors are better than the baseline at the ten-percent confidence level, as measured by the Diebold-Mariano statistic. The N-best techniques again perform well and the LASSO and peLASSO do not.

It is also important to note that the mean is being driven in part by an outlier in the underlying forecasts (indicated by the significant improvement by using the median) and does very poorly. In contrast, the tree-based and N-best techniques were given the same forecasts and were able to successfully ignore the poorly performing outliers, acting as a sanity check, or an automatic pooling method, on forecast inputs. Additionally, it appears that the relative performance of the tree-techniques increases as the number of observations increases (comparing monthly to quarterly results). This phenomenon may be driving by relatively large number of parameters inside the tree-techniques, but to definite conclusion would require more experiments, which we will leave open for future work.

5 Conclusions

We conduct two exercises to evaluate the ability of nonlinear forecast combination techniques. First, using the European Survey of Professional Forecasters in two real-time forecast combination exercises, we document that nonlinear models currently in forecast combination literature can outperform uniformly-weighted forecast combinations. Second, document robust evidence across 8 different real-time forecast exercises of United States macroeconomic data that nonlinear techniques can consistently outperform both uniform weights and median forecasts combinations. The two tree-based forecast combination methods introduced in this paper strictly dominate the mean forecast combination, across all eight exercises. Further, these tree-based methods can serve as a useful sanity check on poorly performing forecasts in a way that the uniform weights cannot. Additionally, it appears that the relative ability of tree-based methods, in comparison to simpler nonlinear techniques, increases as the number forecast observations increases. We also show that when the underlying forecast generating processes are available, then the feature augmented model averaging technique can dominate all other forecast combination techniques. We attribute the success of the FFORMA to its ability to regularize forecast combination weights over many time series in such a way to reduce the possibility of over-fitting on one series, minimizing noise. In total, we document evidence that, in opposition to conventional forecasting wisdom, demonstrates that modern nonlinear forecast combination techniques produce more accurate forecasts than conventional approaches based on equal-weighed forecasts, breaking the so-called “forecast combination puzzle”.

References

Aiolfi, M. and A. Timmermann (2006). Persistence in forecasting performance and conditional combination strategies. *Journal of Econometrics 135*(1-2), 31–53.

Athey, S. and G. Imbens (2019, Mar). Machine Learning Methods Economists Should Know About. *arXiv e-prints*, arXiv:1903.10075.

Bates, J. M. and C. W. J. Granger (1969). The combination of forecasts. *OR 20*(4), 451–468.

Breiman, L. (2001). Random forests. *Machine learning 45*(1), 5–32.

Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen (1984). *Classification and regression trees.* CRC press.

Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM.

Clemen, R. T. (1989). Combining forecasts: A review and annotated bibliography. *International journal of forecasting 5*(4), 559–583.

Conflitti, C., C. De Mol, and D. Giannone (2015). Optimal combination of survey forecasts. *International Journal of Forecasting 31*(4), 1096–1103.

Deutsch, M., C. W. Granger, and T. Teräsvirta (1994). The combination of forecasts using changing weights. *International Journal of Forecasting 10*(1), 47 – 57.

Diebold, F. and R. Mariano (1995). Comparing predictive accuracy. *Journal of Business amp; Economic Statistics 13*(3), 253–63.

Diebold, F. X. and M. Shin (2019). Machine learning for regularized survey forecast combination: Partially-egalitarian lasso and its derivatives. *International Journal of Forecasting 35*(4), 1679 – 1691.

Donaldson, R. G. and M. Kamstra (1996a). Forecast combining with neural networks. *Journal of Forecasting 15*(1), 49–61.

Donaldson, R. G. and M. Kamstra (1996b, 1). Forecast combining with neural networks. *Journal of Forecasting 15*(1), 49–61.

Elliott, G. and A. Timmermann (2004, September). Optimal forecast combinations under general loss functions and forecast error distributions. *Journal of Econometrics 122*(1), 47–79.

Friedberg, R., J. Tibshirani, S. Athey, and S. Wager (2018). Local linear forests. *ArXiv abs/1807.11408*.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis 38*(4), 367–378.

Genre, V., G. Kenny, A. Meyler, and A. Timmermann (2013). Combining expert forecasts: Can anything beat the simple average? *International Journal of Forecasting 29*(1), 108 – 121.

Hastie, T., R. Tibshirani, and J. H. Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction, 2nd Edition*. Springer series in statistics. Springer.

Hyndman, R., G. Athanasopoulos, C. Bergmeir, G. Caceres, L. Chhay, M. O'Hara-Wild, F. Petropoulos, S. Razbash, E. Wang, and F. Yasmeen (2019). *forecast: Forecasting functions for time series and linear models*. R package version 8.7.

Hyndman, R., A. Koehler, R. Snyder, and S. Grose (2002, 02). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting 18*, 439–454.

Livera, A. M. D., R. J. Hyndman, and R. D. Snyder (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association 106*(496), 1513–1527.

Makridakis, S., E. Spiliotis, and V. Assimakopoulos (2018). The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting 34*(4), 802–808.

Montero-Manso, P., G. Athanasopoulos, R. J. Hyndman, and T. S. Talagala (2018). FFORMA: Feature-based forecast model averaging. Technical report.

Smith, J. and K. F. Wallis (2009). A simple explanation of the forecast combination puzzle. *Oxford Bulletin of Economics and Statistics 71*(3), 331–355.

Stock, J. H. and M. W. Watson (2004). Combination forecasts of output growth in a seven-country data set. *Journal of Forecasting 23*(6), 405–430.

Timmermann, A. (2006). Forecast combinations. *Handbook of economic forecasting 1*, 135–196.

A Tree-based algorithms

Algorithm 1: Random Forest

```
for $b = 1$ to $B$ do
    Draw a bootstrap sample $Z^*$ of size $N$ from the training data;
    Grow a decision tree $T_b$ to the bootstrapped data, by recursively repeating the
      following steps for each terminal node of the tree, until the minimum node size
      $n_{min}$ is reached;
    while $n > n_{min}$ do
        Select $m$ variables at random from the $p$ variables.;
        Pick the best variable/split point among the $m$.;
        Split the node into two daughter nodes.;
    end
end
Output the ensemble of trees $T_{b_i}^B$;
```

Source: Hastie et al. (2009)

Algorithm 2: Tree-based stochastic gradient boosting machine (Boosted Tree)

```
Choose loss function $\Psi(y, f)$, learning rate $\lambda$, and tree depth $L$;
Instantiate simple decision tree $f(x)^{(0)}$;
for iteration $m = 1 \dots K$ do
    Compute the gradient $\tilde{y_i} = -(\frac{\partial \Psi(y_i, f(x_i)^{m-1})}{\partial f(x_i)^{m-1}})$ for all observations $i$;
    Sample from training data without replacement ;
    Train a tree model $h_i^{(m)}$ of depth $L$ on the random subset using the gradient as
      the outcome ;
    Update the model $f_i^{(m)} = f_i^{(m-1)} + \lambda h_i^{(m)}$ ;
end
Instantiate trained model $f^{(K)}$ ;
```

Source: Hastie et al. (2009)

Algorithm 3: Feature-based forecast model averaging (FFORMA)

```
Offline Phase: Train the learning model
Input
    $R$: a set of $N$ observed time series, $\{x_1, x_2, ..., x_N\}$, the reference set;
    $F$: a set of functions for calculating the time series features;
    $M$: a set of forecasting methods in the pool;
Output
    FFORMA meta-learner: A function from the extracted features to a set of M
  weights, one for each forecasting method ;

Prepare the meta-data
for $n$ = 1 to $N$ do
    Split $x_n \in R$ into a training period and test period;
    Calculate the set of features $f_n \in F$ over the training period;
    Fit each forecasting method $m \in M$ over the training period and generate
      forecasts over the test period;
    Calculate forecast losses $L_{nm}$ over the test period;
end

Train the meta-learner, w
Train a learning model based on the meta-data and errors, by minimizing:
  $\arg\min_w \sum_{n=1}^{N} \sum_{m=1}^{M} w(f_n)mL_{nm}$ ;

Online Phase: Forecast a new time series
Input
    FFORMA meta-learner from offline phase;
Output
    Forecast the new time series $x_{new}$;

Estimate forecasts
for each $x_{new}$ do
    Calculate features $f_{new}$ by applying F;
    Use the meta-learner to produce $w(f_{new})$ an $M$-vector of weights;
    Compute the individual forecasts of $M$;
    Combine individual forecasts using $w$ to generate final forecasts ;
end
```

Source: Montero-Manso et al. (2018)

B Time Series Forecasting Techniques

When conducting forecast combinations for US macroeconomic series, we estimate our own pool of constituent forecasts. We approximately follow Montero-Manso et al. and estimate eleven univariate time-series models; a constant[8], three random walks, two exponential smoothing models, four ARIMA models, and one artificial neural network. All models are linear, except for the single-layer feed-forward neural network. These methods are standard, simple, and flexible models that have been shown to be successful in a variety of forecasting exercises. The complete list is

- Mean
- Random walk
- Random walk with drift
- Seasonal random walk
- Automated exponential smoothing
- TBATS model
- Theta method
- AR(12)
- Automated ARIMA
- STLM-AR method
- Autoregressive neural network

The mean is the sample mean. Random walk is the previously observed growth rate. Random walk with drift is trend growth plus the previously observed deviation from trend. Seasonal random walk is the growth rate observed 12 periods past. The AR(12) is a standard 12-lag autoregressive forecasting regression. The automated ARIMA is a standard ARIMA model with p, d, and q chosen to minimize Akaike's information criterion. The

[8] Our experiments are all on stationary time series for which a constant is a reasonable forecast.

Table 3: Forecast Combination Results: United States Macroeconomic Series

Time Series Model	Employment				Industrial Production			
	H = 1	H = 6	H = 12	H = 24	H = 1	H = 6	H = 12	H = 24
Historical Mean	1.20	0.99	0.94	0.81	1.06	0.99	0.93	0.92
Random Walk	1.10	1.08	1.07	1.11	1.21	1.32	1.32	1.35
Random Walk (w. drift)	1.10	1.09	1.10	1.17	1.22	1.33	1.35	1.41
Random Walk (w. season)	1.26	1.10	1.07	1.11	1.38	1.34	1.32	1.35
Exponential Smoothing	0.92	0.93	0.99	1.00	1.02	1.15	1.18	1.19
Theta Method	5.06	5.06	5.20	5.98	1.51	1.32	1.20	1.20
TBATS Method	0.94	0.93	0.95	0.90	1.00	0.99	0.95	0.93
STLM-AR	0.98	0.93	0.93	0.85	1.09	1.02	0.95	0.94
ARIMA	0.95	0.93	0.95	0.86	1.01	0.99	0.94	0.92
AR(12)	0.94	0.91	0.90	0.82	1.01	1.00	0.94	0.92
AR-Neural Network	1.04	1.15	1.30	1.19	1.08	1.18	1.17	1.27

Notes: forecast performance is reported as the ratio of the given forecast's RMSE to the mean forecast's RMSE, such that a ratio less than one signals a forecast performance better than using an unweighted average of all forecasts.

STLM-AR applies a an AR (lag order chosen by standard information criterion) to a deseasonalized time series. The autoregressive neural network is the average of twenty single-layer feed-forward networks (see Hastie et al. (2009) for a textbook treatment of simple single-layer neural networks). The exponential smoothing model fits seasonal and trend components to a standard autoregressive state-space model (see Hyndman et al. (2002)). TBATS uses the aforementioned exponential smoothing model, after controlling for seasonal effects (see Livera et al. (2011)). The Theta method is a hybrid exponential smoothing and ARIMA model, as it decomposes time series and then uses either method to forecast the newly separated elements of the time series. Each model is re-estimated every period, using all available information up to date. Following Montero-Manso et al. (2018), our forecasting models are trained using the FORECAST package in R (Hyndman et al. (2019)) with default settings.

Table 3 shows the individual results for each constituent time series forecast in our model pool. As in our other analysis, all ratios are compared relative to the mean forecast combination. The Theta method is a clear outlier in its inability to forecast the monthly employment series. While all AR-based models (AR(12), Automated ARIMA, TBATs, and STLM-AR) are able to achieve success across all series and horizons. Lastly, the historical mean is able to minimize the forecast errors the most, compared to a simple average of all forecasts, when forecasting two-years ahead for both series. This may indicate that more sophisticated methods lose power or simply introduce too much noise as horizons extend.

C FFORMA informationn set robustness

Table 4: United States macroeconomic data forecast combination results — one-time model estimation

Combination Technique	Employment				Industrial Production			
	H = 1	H = 6	H = 12	H = 24	H = 1	H = 6	H = 12	H = 24
peLasso	2.13	1.89	1.46	1.2	1.19	1.29	0.96	1.03
Lasso	2.2	1.9	2.15	2.28	1.18	1.29	0.97	0.97
Random Forest	1.12	0.93	1.45	1.19	1.11	1.22	1.05	1.04
Boosted Tree	1.78	1.07	1.61	1.36	1.07	1.16	1.16	1.19
FFORMA	1.24	0.82	0.79	0.67°	0.49*	0.28*	0.29*	0.30*

Notes: forecast performance is reported as the ratio of the given forecast combination method's RMSE to the mean forecast's RMSE, such that a ratio less than one signals a forecast performance better than using uniform weights. $^{\odot}$, $^{\circ}$, * denote Diebold and Mariano (1995) statistics significant at the ten-percent, five-percent, and one-percent confidence level, respectively, testing that the given forecast combination technique improves upon using uniform weights. Machines are trained on forecast errors from 1970 through 1979, and forecast combinations from 1980 through 2019 are evaluated.

We train the FFORMA machine with 1,000 randomly selected series from the M4 competition, which ran in 2018. As a result, it is possible that the FFORMA is drawing its success from simply having seen enough macroeconomic series through 2018 that it can redraw industrial production and employment across our testing period. That is, the FFORMA machine may be contaminated with a future-biased information set. To evaluate the results of the FFORMA, expunging any possible future-bias, we replicate our primary experiment, but with nonlinear techniques estimated on data only running from 1) 1970 through 1979, and 2) 1970 through 2000. However, noting that this will severely restrict the number of observations available to FFORMA, we allow the machine to train over the approximately 12,000 or 9,000 series out of the 100,000 M4 competition series, with data available from 1970 through 1979 or 1970 through 2000, respectively.

Table 4 shows the results from our FFORMA robustness check when using training data form 1970 through 1980. Even when controlling for any possible future bias in the function estimation, FFORMA is by the best performing forecast combination method. In fact,

Table 5: United States macroeconomic data forecast combination results — one-time model estimation

Combination Technique	Employment				Industrial Production			
	H = 1	H = 6	H = 12	H = 24	H = 1	H = 6	H = 12	H = 24
peLasso	3.05	2.03	1.45	0.93	1.15	1.24	1	0.98
Lasso	2.99	2.01	1.49	1.07	1.16	1.24	1.02	0.99
Random Forest	2.03	2.21	2.77	2.52	1.08	1	1.31	0.90$^{\odot}$
Boosted Tree	5.68	3.89	2.59	2.15	0.98	0.99	1.13	0.91
FFORMA	1.92	1	0.73	0.59	0.36*	0.30*	0.29*	0.26*

Notes: forecast performance is reported as the ratio of the given forecast combination method's RMSE to the mean forecast's RMSE, such that a ratio less than one signals a forecast performance better than using uniform weights. $^{\odot}$, $^{\circ}$, * denote Diebold and Mariano (1995) statistics significant at the ten-percent, five-percent, and one-percent confidence level, respectively, testing that the given forecast combination technique improves upon using uniform weights. Machines are trained on forecast errors from 1970 through 1979, and forecast combinations from 1980 through 2019 are evaluated.

restricting the length of the training time series apears to have had no qualitative affect on the performance of the algorithm. Table 2 and Table 4 report very similar FFORMA results.

Table 5 shows the results from our FFORMA robustness check when using training data form 1970 through 2000. Similar to the smaller sample set, FFORMA is by the best performing forecast combination method. Again, restricting the length of the training time series appears to have had no qualitative affect on the performance of the algorithm. Table 2 and Table **??** report very similar FFORMA results.