Erasmus University Rotterdam

Robeco Institutional Asset Management

Master Thesis

Forecasting Bond Risk Premia with Machine Learning

Tobias Hoogteijling (408976)

University Supervisor: Prof. Dr. M. Van der Wel

Second reader: Dr. R. Lange

Company Supervisors: M. Martens & C. Zomerdijk

October 27, 2020

Abstract

This paper investigates the use of machine learning techniques for forecasting bond returns. A previous literature has documented that neural networks achieve positive predictive power, as opposed to principal-component regressions and tree-based methods. We show that this literature does not take into account issues such as nonstationarity and model uncertainty and is infeasible due to the use of information that is not available at the time of estimation. We propose an alternative setting that greatly improves the forecasting performance of both linear and machine learning techniques. Contrary to the existing literature, we find that all techniques yield positive predictive power. Linear techniques constitute the best performing methods and significantly outperform all machine learning techniques. Our results further indicate trading strategies based on linear forecasts produce significantly higher returns than those based on machine learning forecasts. Finally, machine learning does not provide alpha beyond what is contained in linear approaches.

The content of this thesis is the sole responsibility of the author and does not reflect the view of the supervisor, second assessor, Erasmus School of Economics or Erasmus University.

Contents

1 Introduction

Forecasting bond returns has been a major academic topic for decades. The Expectations Hypothesis (Fisher, 1896) postulates a constant risk premium, implying that there are no variables that have predictive power for excess bond returns. The optimal forecast is then simply the historical mean. The failure of this hypothesis has been well established in the literature by e.g. Fama and Bliss (1987) and Campbell and Shiller (1991). Cochrane and Piazzesi (2005) introduce a linear combination of forward rates that explains a substantial share of the variation in future bond returns. Ludvigson and Ng (2009) extract macro factors from a large set of variables and show that they have additional forecasting power beyond yield curve information. Many others[1] also find that future variation in the term structure is predictable to some extent, but their approaches can generally be characterized as linear. Recently, machine learning techniques have received more and more attention as an alternative to linear forecasting techniques, as the rise in computing power has made them more feasible. Machine learning techniques provide a model-free mapping between the dependent and independent variables and as such they can potentially capture complex nonlinear relations, which cannot be captured by traditional methods such as linear regression. Can machine learning approaches add to existing techniques in forecasting the bond market?

Bianchi, Tamoni, and Büchner (2020b) compare the performance of a broad range of machine learning techniques in forecasting bond risk premia. They study excess returns on US government bonds, defined as the 1 year holding period return of an n month bond minus the return on a 1 year bond. They compare forecasts of linear and machine learning techniques using forward rates to the benchmark of the Expectations Hypothesis. The authors find that artificial neural networks significantly outperform both other forecasting approaches and the benchmark in terms of out-of-sample R^2, for both a setting including and a setting excluding economic variables. They document that linear techniques and tree-based methods have no predictive power.[2]

However, in this paper, we argue that there are several issues with the work of Bianchi et al. (2020b), which make their setting unsuitable for evaluating forecasting performance. Firstly, the strong predictive power of neural networks found by Bianchi et al. (2020b) relies on macroeconomic information. The authors make use of a macroeconomic data set containing information which was not available in real time due to publication lags and data revision. As shown by

[1] e.g. Bansal and Shaliastovich (2013) and Joslin, Priebsch, and Singleton (2014).

[2] The original paper also reports strong positive predictive power for tree-based methods, but the corrigendum (Bianchi, Büchner, Hoogteijling, and Tamoni, 2020a) shows that future observations have been used to obtain these results. When excluding those, forecasting power decreases strongly for all methods, and tree-based methods no longer have predictive power.

Ghysels, Horan, and Moench (2018), using a final data set leads to a forward looking bias and the forecasting power of macro variables can be much smaller when using real time data. Secondly, Bianchi et al. (2020b) use forward rates to capture yield curve information. We show that forward rates are nonstationary, which is well-known to be potentially problematic for both linear regressions and machine learning techniques, and we illustrate how nonstationarity thwarts accurate model estimation. Thirdly, Bianchi et al. (2020b) do not take model uncertainty into account in their forecasts. This is an especially pressing issue because bond returns are only predictable to a very limited extent.

We propose the following changes to the methodology to deal with these issues. Firstly, we use vintage rather than final macro data to make the analysis feasible in real time. Furthermore, we replace forward rates by their first differences to obtain stationary explanatory variables. Finally, we take model uncertainty into account with a Bayesian approach. We use the historical sample mean as prior prediction and shrink all predictions to this benchmark. We examine the impact of each of these issues and investigate whether machine learning techniques outperform linear techniques in forecasting bond risk premia.

In our research, we consider excess returns on US government bonds over the period of August, 1971 till December, 2018, with the out-of-sample period starting in January, 1990. We compare the predictive power of linear regressions, tree-based methods and neural networks with two different sets of input variables. In the first setting, we use only yield curve information. In the second setting, we use yield curve information combined with a large set of macroeconomic variables.

Compared to the results of Bianchi et al. (2020a), we find improvements for all methods, but most strongly for linear regressions. Interestingly and contrarily to the findings of Bianchi et al. (2020b), we find that linear techniques constitute the best performing rather than worst performing methods, in all settings. In fact, simply replacing the forward rates with their first differences is already sufficient for regressions to outperform all machine learning techniques across all maturities. Also taking misspecification risk into account by imposing a prior generally improves predictions compared to the frequentist approach. Using vintage instead of final macro data decreases forecasting performance, mostly so for neural networks.

We also assess the economic significance of the predictive power of the three techniques, as a positive out-of-sample R^2 does not necessarily imply positive trading gains. We establish trading strategies that interpret model forecasts as signals and convert them into z-scores. We find that strategies based on linear regression forecasts outperform strategies based on machine learning forecasts both in terms of Information Ratios and Sharpe Ratios. Furthermore, we test whether machine learning techniques capture alpha beyond what is found by linear techniques with encompassing regressions. We regress machine learning trading returns on linear regression trading returns, and find no significantly positive alpha. Reversing these regressions indicates that linear techniques do yield positive and significant alphas beyond what is captured by machine learning.

We document that our main finding, that linear regressions outperform machine learning techniques, is very robust. It also holds in a different setting, where known predictors such as yield spread and momentum are used. It also extends to extreme trees, gradient-boosted trees and alternative configurations of neural networks. Finally, we set up a formal portfolio choice problem and show that linear regressions lead to significantly higher utilities for a mean-variance investor than machine learning techniques.

The remainder of this paper is organised as follows. An overview of relevant literature is given in Section 2. Methodology is discussed in Section 3 and data in Section 4. Main results are presented in Section 5 and additional results and robustness checks in Section 6. Section 7 concludes.

2 Literature

The Expectations Hypothesis states that the risk premium does not vary over time. This implies that future excess returns are not predictable, and that the optimal forecast of the future excess returns is simply the historical excess return. The empirical evidence against the Expectations Hypothesis is extensive. Fama and Bliss (1987) show that current forward rates can be used to forecast future excess returns. Campbell and Shiller (1991) show that the current yield spread can be used to forecast both future yields and future excess returns. Cochrane and Piazzesi (2005) find that a tent-shaped linear combination of one to five year forward rates, dubbed the CP factor after the names of the authors, can explain a share of up to 44% of the variation in future one year excess bond returns.

Ang and Piazzesi (2003) introduce macro factors in term structure models. They add principal components of groups of macro variables to VAR term structure models and find this substantially improves yield forecasts. In their setting, the macro factors are spanned by the yield curve, i.e. all information in the macro variables is contained in the yield curve. This Spanning Hypothesis is often rejected, as it also implies that conditioning on the yield curve, macro variables are uninformative about future macro variables (Shiller and McCulloch, 1990). Evidence for unspanned macro factors was found by Ludvigson and Ng (2009), who consider a set of 132 macro variables and find that combining principal components of those with the

CP factor increases forecasting performance. Joslin et al. (2014) formulate a model in which some output and inflation risks are not spanned by the yield curve and find that these risks explain future variation in the term structure. Other works on the inclusion of macro variables in term structure models are e.g. Cooper and Priestley (2009), Bansal and Shaliastovich (2013), Greenwood and Vayanos (2014) and Cieslak and Povala (2015). A different approach is taken by Baltussen, Martens, and Penninga (2020), who forecast excess returns using yield spread, momentum, equity returns and commodity returns. They convert movements in these explanatory variables into z-scores and show that each variable has significant predictive power.

The evidence on the failure of the Expectations Hypothesis and the Spanning Hypothesis is not free from criticism. Thornton and Valente (2012) report that much of the research on excess return predictability is subject to econometric issues such as a possible spurious regression problems. Furthermore, they document that the no-predictability benchmark is hard to beat out-of-sample. Finally, they show that improved forecasts do not necessarily translate in an increase in investor utility in a quadratic utility setting. Hamilton and Wu (2012) report the presence of identification issues for studies on unspanned macro variables. Ghysels et al. (2018) attribute the forecasting power of economic variables to the use of final data and show that real time macro variables explain a much smaller share of the variation in bond returns. Bauer and Hamilton (2018) claim that the used frameworks for unspanned macro factors are subject to small sample distortions and that the evidence against the Spanning Hypothesis is much weaker when using correct standard errors. Academic consensus on the Spanning Hypothesis has thus not been reached.

Machine learning has been introduced to financial forecasting only recently. Heaton, Polson, and Witte (2017) study portfolio construction using deep neural networks. Gu, Kelly, and Xiu (2020) compare machine learning techniques for asset pricing. Our work also relates strongly to Bianchi et al. (2020b), who forecast excess bond returns using a multitude of machine learning techniques. They consider both a yields-only setting, as in Cochrane and Piazzesi (2005), and a setting with unspanned macro variables, as in Ludvigson and Ng (2009). Machine learning techniques can potentially capture non-linear effects unspanned macro variables can have on the yield curve and their use can therefore contribute to the academic debate on the Spanning Hypothesis.

Whereas Bianchi et al. (2020b) use final macro data in forecasts, Feng, Fulop, and Li (2020) use real time macro data to forecast bond returns with machine learning. They report only minor forecasting power for both linear and machine learning techniques with vintage macro data. This does not translate to positive economic value for investors in a mean-variance setting. However, the differences between the results of Bianchi et al. (2020b) and Feng et al. (2020) cannot solely be attributed to the use of final versus vintage macro data. Firstly, Feng et al. (2020) do not include yield information in their forecasts, which was responsible for a substantial share of the predictability in the work of Bianchi et al. (2020b). This also implies that the paper does not offer evidence pro or contra the Spanning Hypothesis. As yield information was not included in the forecasts, we do not know if any of the macro variables' forecasting power was also contained in ('spanned by') the yield curve. Secondly, they only consider 56 'true' macro variables, such as employment and production, but disregard financial variables, such as price, stock market, bond market and exchange rate information. These variables were part of the 128 variable data set of Bianchi et al. (2020b) and were also important for forecasting. Finally, they consider a different time period, different maturities[3] and a different approach to some of the machine learning techniques. As the impact of using vintage macro data in machine learning is thus not yet clear, this is one of the core issues we examine.

3 Methodology

In Section 3.1, we outline how excess returns are constructed and forecast. In Section 3.2, we describe the linear approach and in Section 3.3 the machine learning techniques that we use. In Sections 3.4-3.6, we discuss the issues with the work of Bianchi et al. (2020b): real time macro information, nonstationarity and model uncertainty. Sections 3.7 and 3.8 are on the standard errors of the out-of-sample R^2's and pairwise tests of significance. Finally, a real time investment strategy is discussed in Section 3.9.

3.1 Problem description

We contrast the performance of linear methods and machine learning techniques in forecasting of bond risk premia in two settings. In the first setting, we only make use of information in the yield (or forward) curve itself, corresponding to the setting in Cochrane and Piazzesi (2005). We use the forward rates to capture this information,[4] but as the forward rates and yields contain the same information, we also refer to this as the yields-only setting. In the second setting, we make use of information in the yield curve and a wide range of macro variables, as in Ludvigson and Ng (2009).

[3] They only forecast bonds with maturities of up to 5 years, whereas the strongest predictability was found for 7 and 10 year bonds in Bianchi et al. (2020b).

[4] It is also possible to use yields instead. We use forward rates to make our work directly comparable to that of Bianchi et al. (2020a). Arguably, forward rates are more directly related to future excess returns than yields, and therefore more suitable for predictions.

Throughout our analysis, we use excess bond returns as dependent variable. The yearly excess bond return on an n month bond is the return of holding this bond for one year minus the one year yield. Excess bond returns are a common dependent variable in the literature and are suitable for three reasons. First of all, they have a clear interpretation as the risk premia earned by taking the risk of holding a bond for a shorter time period than the time to maturity. Secondly, excess bond returns are in real terms and need not be corrected for inflation or interest rate levels (Cochrane and Piazzesi, 2005). Thirdly, there is a clear and tough-to-beat benchmark in terms of the Expectations Hypothesis. The Expectations Hypothesis postulates a constant expected risk premium. This translates into a benchmark prediction that equals the historical sample mean (Shiller and McCulloch, 1990).

Let $p_t^{(n)}$ denote the log price of a zero coupon bond at time t (in months) with pay-off \$1 and maturity $t+n$. Its continuously compounded yield is denoted by $y_t^{(n)} = -\frac{12}{n} p_t^{(n)}$. The one year log excess returns on an n month bond (with $n \geq 12$) can then be computed as

$$\begin{aligned} xr_{t:t+12}^{(n)} &= p_{t+12}^{(n-12)} - p_t^{(n)} - y_t^{(12)} \\ &= -\frac{n-12}{12}\, y_{t+12}^{(n-12)} + \frac{n}{12}\, y_t^{(n)} - y_t^{(12)} \\ &= -\left(\frac{n}{12} - 1\right)\left(y_{t+12}^{(n-12)} - y_t^{(n)}\right) + \left(y_t^{(n)} - y_t^{(12)}\right), \end{aligned} \tag{1}$$

which is the accounting identity in Campbell and Shiller (1991).[5] We stack the excess returns $xr_{t:t+12}^{(n)}$ for $n = 24, 36, 48, 60, 84, 120$ in a vector $xr_{t:t+12}$, which are thus the excess returns that are earned between point t and point $t+12$. We describe the mapping from the information set at time t, denoted by x_t, to the excess returns between time t and time $t+12$ using the function $g(\cdot)$,

$$xr_{t:t+12} = g(x_t) + \varepsilon_t, \tag{2}$$

with ε_t being an error term. Following Bianchi et al. (2020b), we consider an expanding window setting, in which the relation in Equation (2) is re-estimated every period. The sample period is from August 1971, when the first 10 year US government bond was issued, till December 2018. The out-of-sample period starts in January 1990. In the most straightforward case only yield information (in the form of forward rates) is used to predict excess returns, corresponding to the results in Table 1 in Bianchi et al. (2020a). We thus have $x_t = \left[y_t^{(12)}, f_t^{(24)}, \ldots, f_t^{(120)}\right]$, with $f_t^{(n)}$ being the one year forward rate defined as

$$f_t^{(n)} = \frac{n}{12}\, y_t^{(n)} - \left(\frac{n}{12} - 1\right) y_t^{(n-12)},$$

[5] The equation illustrates that a forecast of a change in the bond yield $y_{t+12}^{(n-12)} - y_t^{(n)}$ can be translated in a forecast of the excess return $xr_{t:t+12}^{(n)}$, and vice versa (Bianchi et al., 2020b).

for n = 12, 24, 36, 48, 60, 84, 120. The predictive performance of the various methods is evaluated using the out-of-sample R^2 (R^2_{oos}) as proposed by Campbell and Thompson (2008):

$$R^2_{oos} = 1 - \frac{\sum_{t=1}^{T-12}(xr^{(n)}_{t:t+12} - \hat{xr}^{(n)}_{t:t+12}(M))^2}{\sum_{t=1}^{T-12}(xr^{(n)}_{t:t+12} - \bar{xr}^{(n)}_{t:t+12})^2}, \tag{3}$$

where $t = 1$ corresponds to the first out-of-sample observation, T is the length of the data set, $\hat{xr}^{(n)}_{t+12}(M)$ is the prediction of model M for time to maturity n and $\bar{xr}^{(n)}_{t:t+12} = \frac{1}{t-12}\sum_{s=1}^{t-12} xr^{(n)}_{s:s+12}$ is the benchmark prediction. A positive R^2_{oos} implies the method predicts better than the benchmark. The out-of-sample R^2 can be interpreted as the percentage reduction in Mean Squared Prediction Error compared to the benchmark. In the remainder of this paper, we sometimes omit the "oos" subscript for brevity.

3.2 Linear approach

In a linear approach, we explicitly specify $g(x_t)$ in Equation (2) to be of a linear form. In our first setting, this implies a simple principal-component regression. The information set at time t is captured by the principal components v_t of the forward rates $f^{(n)}_t$. Reversely, $f^{(n)}_t$ are a linear combination of v_t and we write

$$f_t = a_0 + Av_t + \varepsilon_t,$$

where a_0 is a constant, A is a matrix of coefficients and ε_t represents the error term. The error terms are subject to the standard regression assumptions of being uncorrelated with the regressors and having a zero mean. However, due to the overlapping returns, they exhibit autocorrelation and heteroskedasticity by construction. This is taken into account by using HAC standard errors. It follows from Equation (1) that the excess returns are also a linear combination of the principal components:

$$xr_{t:t+12} = b_0 + Bv_t + \eta_t, \tag{4}$$

where b_0 is a constant, B is a matrix of coefficients and η_t represents the error term.

In the second setting, we also consider a set of principal components m_t that has been extracted from a large set of macro variables. As Ludvigson and Ng (2009) and Bianchi et al. (2020b), we consider the first 8 principal components. However, adding all those to to the three yield-based principal components leads to an abundance of regressors and potentially to high estimation error (Heij et al., 2004).

We therefore follow the two-stage approach outline in Ludvigson and Ng (2009). First, we regress excess returns on individual principal components and a constant. We retain the components that are significant at a one percent level. Second, we consider regressions of excess returns on the yield-based principal components and a subset of components that were significant in the first stage.[6] We finally select the model with the lowest Bayesian Information Criterion (BIC) (Schwarz, 1978). This procedure is repeated after every observation, and the number of components included in the regression can thus vary over time. The model specification then becomes

$$xr_{t:t+12} = b_0 + Bv_t + Cm_t + \eta_t,$$

with C being a matrix of coefficients. Details on the principal-component regressions are found in Appendix C.1.

3.3 Machine learning

Machine learning techniques are techniques that improve themselves. There exist a multitude of techniques, some of which can be classified as ‘shallow’, e.g. support vector machines and linear discriminant analysis, and others as ‘deep learners’, such as regression trees and neural networks. A ‘shallow’ technique is one that does not allow for highly nonlinear relationships and is not very flexible. A ‘deep learning’ technique has more parameters and is thus more flexible in fitting the data. The distinction between deep and shallow techniques is not always black-and-white and often techniques can be tuned to become either more deep or more shallow. Choosing the depth of a technique involves a trade-off. Deeper techniques are more flexible and are therefore better able to more closely fit the data. However, this comes with the risk of overfitting (Hastie et al., 2009).

In machine learning, we also distinguish between supervised learning and unsupervised learning.[7] A supervised learning technique tries to find the best possible mapping between input data and output data. Tree-based methods and neural networks are examples of supervised learning. Unsupervised learning is concerned with data compression. A classic example of unsupervised learning is principal component analysis, although it is not so much ‘machine’ learning (Hastie et al., 2009). Bianchi et al. (2020b) find that neural networks and tree-based methods perform best all applications considered, so we focus on those.

There are both advantages and disadvantages to machine learning. A major advantage is that

[6] Ludvigson and Ng (2009) also consider square and cube terms, but we are specifically only interested in linear effects.

[7] Sometimes reinforcement learning is distinguished as a third class, but this is not a relevant distinction for our purposes.

machine learning techniques can be used to perform tasks that are too much work for humans. Machine learning techniques were introduced not because they could perform tasks better than human, but because they could perform them quicker (Daffodil Software, 2017). Consider the example of a spam filter. Arguably, humans are also able to identify spam mails and will maybe even classify less emails incorrectly as spam. However, people still like spam filters because they save them time and annoyance, even though they are not perfect. The vast majority of machine learning techniques used successfully in our lives are still used because they are quicker and cheaper than humans, rather than better. Examples are speech-to-text applications, virtual personal assistants, video surveillance, social media services, online customer support, product recommendations, etc. (Daffodil Software, 2017). Note that this advantage is not relevant for predictions of bond risk premia. Researchers care not so much for a quick prediction, but for an accurate one.

The use of machine learning techniques to do things better than humans only came much later and constitutes only a small share of all applications (Foote, 2019). In 2017 for example, AlphaZero was the first machine learning chess computer that surpassed human level (Silver et al., 2018).[8] AlphaZero relies on huge computing power to run many simulations of chess games. Other applications of machine learning that outperform humans are for instance videogames and art imitations (Steinberg, 2017). However, there is a major difference between these successful applications of machine learning and financial forecasting. These applications make use of either large data sets, or they are able to simulate a lot of data as there is perfect knowledge of the data generating process. In chess for example, the new situation on the board after every possible move is perfectly known beforehand. Furthermore, these applications are generally characterized by a high signal-to-noise ratio. If you lose your queen in a game of chess, this is in general obviously not a good sign. When forecasting excess bond returns, this is all different. We do not have access to a large amount of data, nor do we know the data generating process. In fact, we do not even know if this data generation process remains the same over time. Furthermore, signal-to-noise ratios are usually very low in finance. Nevertheless, there is a third advantage to the use of machine learning, which is very relevant for our purposes. Machine learning techniques are potentially able to capture (hidden) non-linear relationships that are largely missed by linear approaches (Mitchell, 2006).

We now turn to the drawbacks of machine learning. First of all, the methods are ‘black box’, which implies that the researcher does not know what happens ‘inside’ the algorithm. Machine learning is also very prone to overfitting, although several solutions for this exist,

[8] Chess computers that beat human players have been existing for a much longer time, but they do not rely on machine learning.

such as regularization, forecast averaging and early stopping. Overfitting is especially an issue when signal-to-noise ratios are low and data availability is limited, as in our case. Thirdly, the techniques can be very computationally intensive, especially in the case of neural networks.[9] Finally, machine learning techniques are very sensitive to the specific settings chosen by the researcher.

3.3.1 Parameters and hyperparameters

Within machine learning, it is important to distinguish between model parameters and hyperparameters. Model parameters are parameters of the model itself. Hyperparameters define how the technique learns about the model parameters, and how they relate to each other (Mitchell, 2006). As an illustration, consider a machine learning technique that considers a subset of observations, and assigns different importance weights to these observations in order to achieve the best fit. In this case, the size of the subset and the range of values these weights can take are hyperparameters. The weights are adjusted by the machine learning technique to obtain the best fit, and are thus model parameters.

The distinction between model parameters and hyperparameters is not always so clear-cut. The performance of machine learning techniques can be strongly dependent on the hyperparameters. For this reason, many machine learning techniques do not use the traditional splitting of data in a train and test sample. Instead they have the three consecutive stages of training, validation and testing. The training stage is used to determine the values of the model parameters, the validation stage to determine the values of the hyperparameters, and the testing stage to assess the model performance. Machine learning can also be highly sensitive to the way these hyperparameters are validated.

A common way to validate machine learning techniques is k-fold cross validation, which splits up the data in k different groups, fits the model on $k-1$ groups and validates on the remaining group. This procedure is repeated for all groups, and the hyperparameters are chosen that lead to the smallest average validation error. This is not suitable for our purposes however, due to the time series nature of our data. The data can contain temporal dependencies, and validating on data that precede the training data can distort the results. A solution can be to split the in sample data in a training and validation set, such that the training sample fully precedes the validation sample. This requires a decision on how to split the data in training and testing data. Following Bianchi et al. (2020b), we opt for a training-validation split of 85%-15%, which

[9] Bianchi et al. (2020b) report making use of supercomputing clusters provided by the University of Warwick, enabling them to run their code on over 2300 cores at the same time. For comparison: a normal computer has between 1 and 6 cores (but computational power is more complicated than the number of cores).

is common in the machine learning literature (Hastie et al., 2009).

It is important to note that some hyperparameters cannot be chosen by validation. The validation loss is not truly out-of-sample, as it is used to choose or tune hyperparameter values. For neural networks, many hyperparameters need to be chosen by the researcher ex ante. In some cases, this is due to the fact that the hyperparameters influence the overall order of magnitude of the validation error. For example, a larger number of nodes makes the network more flexible and generally leads to smaller losses but a higher risk of overfitting. In other cases, this is computationally too intensive. For example, some parameters cannot be chosen by numerical optimization but rather involve re-running the entire network.

For this reason, we report the results of various configurations of neural networks, choosing hyperparameters that have been found to be good choices empirically in other work. Nevertheless, our search for appropriate values of hyperparameters is far from exhaustive and the possibility always remains that different choices of parameters yield different results. For tree methods on the other hand, the full set of hyperparameters can be chosen using validation.

3.3.2 Tree-based methods

Tree-based methods are an ensemble method of individual decision trees. A decision tree is a tool that divides observations over several branches and subbranches, such that the observations in the same branch are as similar as possible, and observations in a different branch as different as possible. Decision trees make use of the Classification and Regression Tree (CART) Algorithm. At every split, the left subbranch contains all observations with a certain variable smaller than a certain threshold, whereas the right branch contains all observations for which that variable is larger than the threshold. The ends of the branches are called leafs (Hastie et al., 2009).

As a simple example, consider a decision tree with dependent variable $xr_{t-12:t}^{(n)}$ and explanatory variables $s_t^{(n)}$ for n=2, 3 and 5. A graphical illustration of such a tree can be found in Figure 1.

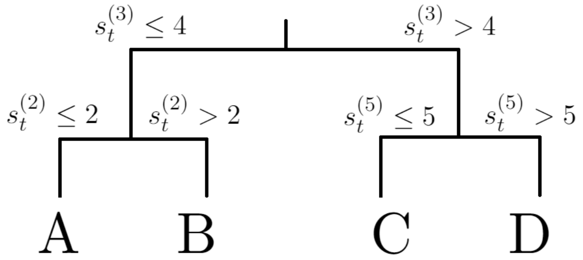

Figure 1: A simple example of a decision tree with explanatory variables $s_t^{(n)}$ for $n = 2,\ 3,\ 5$, with the subsets $A,\ B,\ C$ and D at the leafs.

To make a prediction for $xr_t^{(n)}$, the decision tree considers to which leaf observation t would belong, and takes the average of all excess returns in that leaf as prediction. The idea is that the historical observations that are the most similar to the current observation, are the most relevant for predictive purposes. However, individual decision trees are generally biased and have large variance. Therefore, researchers do not consider individual trees, but combine the results of many.

The simplest approach is growing multiple trees independently from each other and averaging their results. This approach is referred to as a random forest, which we will use as our main tree-based method. In a random forsest, each tree is fit using a bootstrap subsample of the data. The bootstrap sample is drawn with replacement from the full dataset, and may only consider a subset of all variables. Important hyperparameters to be validated are the number of variables to use in a bootstrap, the maximum depth of a tree and the number of trees. By considering only a subset of variables, the tree is able to identify different effects different variables can have, and prevents the tree from always being dominated by the same variables. Setting a maximum depth to the tree prevents overfitting and reduces variance, as this assures predictions are made based on a still reasonably sized subsample.[10] Using multiple trees reduces both bias and variance, as it averages predictions. When using tree-based methods, the values of all hyperparameters can be determined by the algorithm itself using cross validation. The researcher only needs to decide between which values of the hyperparameters the algorithm can choose. A full description of the computational details is given in Appendix C.2.

We also consider two other tree-based methods: extreme trees and gradient-boosted trees. A random forest is an example of a bagging procedure, which means that the trees are grown independently from each other. An extreme tree is an alternative bagging procedure. There are two main differences between extreme trees and random forests. Firstly, extreme trees do not use a bootstrap sample, but the data itself to build the forest. Secondly, the values used for splits are not optimized, but randomly decided (Hastie et al., 2009). The performance of extreme trees and random forests is generally comparable, although some claim that extreme trees perform a bit better in the presence of noisy data (Ceballos, 2019; Trip, 2019).

An alternative to bagging is boosting. In a boosting procedure, trees are grown recursively and depending on the success of the previous tree. The algorithm retains elements from previous trees that led to good fit and changes elements that did not lead to good fit (Géron, 2017). We consider a tree-based method that uses boosting called a gradient-boosted tree. Gradient-boosted trees generally work well in cases of high signal-to-noise ratios, but can perform poorly otherwise (Stephanie, 2019). We therefore expect gradient-boosted trees to perform worse than

[10] Alternatively, it is possible to set a minimum number of observations to be at each split or in each leaf.

random forests and extreme trees.

3.3.3 Neural networks

Artificial neural networks are among the most complicated of machine learning methods. A neural network is a connected system of nodes, inspired by neuron systems in animal brains (Goodfellow et al., 2016). An illustration of a simple neural network is given in Figure 2.

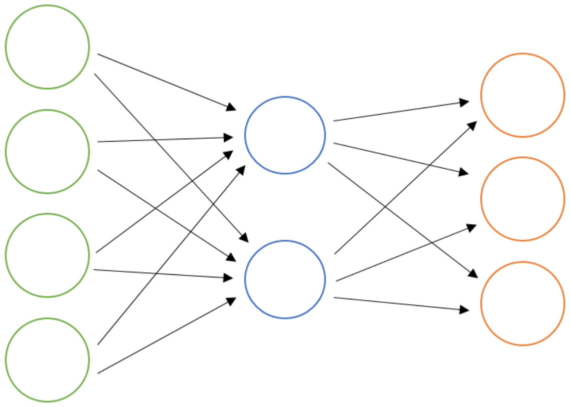

Figure 2: A graphical illustration of a neural network with 1 hidden layer. The layer on the left is the input layer, the middle layer is the so-called hidden layer and the layer on the right it the output layer. The nodes in the middle layer receive information from the input layer, perform a transformation and pass the information on to the output layer.

Neural networks try to find the most accurate mapping between input data and output data. The number of hidden layers and the number of nodes per layer are hyperparameters that are set by the researcher beforehand. Each node in a hidden layer or output layer receives input from the previous nodes, and assigns these nodes weight. Consequently, it performs a so-called activation function on the sum of these weighted inputs, and passes the result on to the next layer. As activation function for the hidden nodes we use the Rectified Linear Unit (ReLU), which is a common choice.[11] (Géron, 2017). The ReLU function is defined as $f(x) = \max(0, x)$. As activation function in the output layer we use a simple linear function, to allow the neural network to freely choose output. Other common activations functions in output layers are the Sigmoid and Softmax functions, but they restrict output to be between -1 and 1 or 0 and 1 respectively.

During optimization, the neural network keeps track of the training loss and the validation loss. The training loss is measured by a certain metric, and indicates the fit of the model on the training data. The network keeps adjusting parameters to minimize the training loss. The validation loss is similar to the training loss, but measured on the validation sample (Chollet,

[11] Other possibilities are Exponential Linear Unit (ELU) and Leaky Rectified Linear Unit (LReLU). Empirical evidence suggests neural networks are rather robust to different choices of activation functions (Géron, 2017).

2017).

Other important terms are epochs and batches. Epochs are the number of times the data is passed on to the model. Each epoch, the observations are split up in batches, which are fed to the network sequentially. For every batch, the model parameters are adjusted to decrease the training error over that specific batch. The model performance is sensitive to the batch size. A too large batch size requires a lot of computation power and memory storage. A too small batch size leads to very volatile model parameters, and as a result long running times and an increased possibility of being stuck in a local optimum. A common batch size is 32, which we adopt (Chollet, 2017). There exist various optimization procedures to minimize the loss metric, such as Root Mean Square Prop (RMSprop) and Adaptive Gradient Algorithm (Adagrad). We follow Bianchi et al. (2020b) and use Stochastic Gradient Descent (SGD) with Nesterov Momentum (Nesterov, 1983).[12]

Overfitting is a major issue for neural networks, which we combat using Early Stopping procedures. Without Early Stopping, the algorithm will keep altering model parameters in order to decrease the training loss until the maximum number of epochs is reached. After a while, the network will overfit on the training sample and the validation loss will deteriorate. We stop the model prematurely if the validation loss has not improved in 20 consecutive epochs,[13] in which case we restore the best model so far. A full description of the computational details can be found in Appendix C.3. The results reported in Section 5 are obtained with a neural network with 1 hidden layer of 3 nodes in settings 1 and 3 and with a neural network of 1 hidden layer of 32 nodes in setting 2.[14] We vary these settings in the robustness checks discussed in Section 6.5.

3.3.4 Validation

Consecutive yearly excess returns have 11 of their 12 months in common and exhibit an autocorrelation of about 92%. As a result, the excess return observed in period $t-1$ ($xr_{t-13:t-1}^{(n)}$) is extremely informative when predicting the excess return observable in period t ($xr_{t-12:t}^{(n)}$). As illustrated by the difference between the results in Bianchi et al. (2020b) and Bianchi et al. (2020a), machine learning techniques are able to exploit this information and generate very good forecasts artificially. However, we are forecasting one year ahead and the eleven observations preceding $xr_{t-12:t}^{(n)}$ are not available at the time of estimation.

[12] Stochastic Gradient Descent is an algorithm used to find the minimum of the loss function. The learning rate is a parameter that determines how much parameters are changed after each batch ('how fast the network learns'). Momentum allows the algorithm to 'remember' previous directions. If multiple previous steps are taken in the same direction, the algorithm speeds up in that direction, thereby generally decreasing running time and leading to a lower risk of being stuck in a local minimum. For momentum, we use a value of 0.9, which is a common choice (Chollet, 2017).

[13] This is referred to as the patience. Here too, we follow Bianchi et al. (2020b).

[14] These configurations are also studied by Bianchi et al. (2020b).

This implies that the train-validate-test split as proposed by Bianchi et al. (2020b) is potentially problematic. When the validation sample directly follows the training sample, this allows the machine learning techniques to incorporate the information in the overlapping returns in the tuning of the hyperparameters. However, these hyperparameters need not be optimal when forecasting with only real time information. For example, this can lead to selecting a too low value of the regularization parameter, because the model overestimates the forecasting power of the information in the explanatory variables. We therefore propose to drop the last 11 observations of the training sample, thereby preventing any overlap between the training and validation samples. Similarly, 11 observations are dropped between validation and testing. An illustration of the train-validate-test split over time can be found in Figure 3.

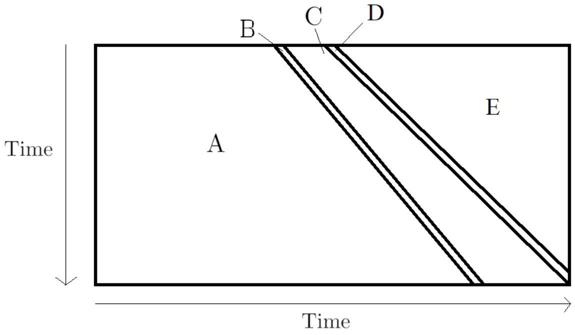

Figure 3: A graphical illustration of the split in training, testing and validation data. Area A corresponds to training data, area B consists of the 11 observations that are dropped between training and validation, area C is the validation data, area D consists of the 11 observations dropped between validation and testing and area E is the remaining testing data (of which only the first observation is considered every time).

3.4 Real time macro information

There arguably is a strong link between macroeconomic developments and the yield curve. For example, the interest rates set by central banks and the risk premia demanded by investors can be dependent on the state of the economy. However, market efficiency implies that if this information is public, it should be incorporated in the prices (and yields) of bonds. As such, the Spanning Hypothesis states that macro variables are not informative regarding future movements in the bond markets beyond the information contained in the yield curve. In other words, these macro factors are ‘spanned by’ the yield curve. However, some studies[15] report that adding macro variables to yield variables improves excess return forecasts. To allow for the existence of unspanned macro factors, we include a large macro data set. Following Bianchi et al. (2020b), we use the macro data set provided by McCracken and Ng (2016). This data set consists of 128

[15] Ludvigson and Ng (2009), Cooper and Priestley (2009), Joslin et al. (2014) to name a few.

variables that cover a wide range of economic indicators, and closely resembles the data set used by Ludvigson and Ng (2009).

Bianchi et al. (2020b) make use of the final data set as available at the end of the sample period. However, the information in this data set is reported with a one or two month publication lag. Moreover, the data is still subject to revisions after publication. The analysis in Bianchi et al. (2020b) is thus not feasible in real time. Ghysels et al. (2018) show that bond return forecasts can strongly benefit from using final rather than vintage data, and that the forecasting power of macro variables is much smaller in real time. This is an import issue to investigate, as the forecasting power reported in Bianchi et al. (2020a) is large when using macro variables, but much smaller when only using (real time) yield information. To assess the impact of using final rather than vintage macro data, we perform our research using both, and contrast the results. In the remainder of the paper, we then focus on the results obtained with vintage macro data, to ascertain that they are due to a forward looking bias.

Unfortunately, vintage data is only available from 1999:08 onwards. We therefore use the 1999:08 macro data set for the first 8 years of the out-of-sample period, thereby still taking into account the publication lag and only ignoring the effect of revisions in this period. The number of revisions in this period is small and results reported in Table 34 in Appendix B indicate that our conclusions also hold when starting the analysis in 1999:08 with only real time data.

3.5 Nonstationarity

Bianchi et al. (2020a) report that linear regressions[16] and tree-based methods produce negative R^2_{oos}'s in the yields-only setting. Positive out-of-sample R^2's are found for neural networks, but these are small and not always significant. Although only neural networks achieve positive predictive power, we argue below that these results do not necessarily imply that neural networks outperform linear methods in the forecasting of bond risk premia.

The forward rates that are used as explanatory variables are nonstationary and exhibit a downward trend since approximately 1985. It has been well established in the literature that using nonstationary regressors is potentially problematic. E.g. Uhlig (2009) and Onatski and Wang (2020) indicate that using nonstationary regressors in principal-component regressions potentially leads to spurious regression problems. Machine learning techniques can run into severe problems with nonstationary data as well, see e.g. Jung and Shah (2015) and Sugiyama et al. (2013). The intuition behind this is that the nonstationarity dominates the model estimation.

[16] Cochrane and Piazzesi (2005) find that in a linear fashion, current forward rates do hold strong predictive power for future bonds returns. However, their analysis relies mostly on in-sample results. Furthermore, their analysis uses data of up to 2003, whereas the out-of-sample period in Bianchi et al. (2020b) is 1990:01-2018:12.

Thus, using forward rates as input variables directly can potentially produce poor results for all techniques considered. We will illustrate the nonstationarity in Section 5.2, and also provide intuition why it thwarts model estimation.

We therefore transform the data by taking first differences,[17] to obtain the stationary time series $\Delta f_t^{(n)} = f_t^{(n)} - f_{t-12}^{(n)}$. Stationarizing data by taking first differences is an approach taken in e.g. Litterman and Scheinkman (1991) and Garbade (1996). If forecasting power greatly improves when using transformed data, this suggests an alternative interpretation of the findings reported in Bianchi et al. (2020a): neural networks do not necessarily outperform linear methods in forecasting bond risk premia, but are simply better in dealing with nonstationarity (in this case at least).

3.6 Model uncertainty & Bayesian shrinkage

In the frequentist approach we have discussed so far, we do not take into account the fact that our model might be misspecified. Excess bond returns are notoriously hard to predict and very noisy. As such, predictions of bond returns based on historical data are particularly subject to misspecification risk. To take this uncertainty into account, we propose a Bayesian approach. Turning back to the example in Equation (4), if the model is correctly specified, the optimal forecast is

$$\hat{xr}_{T:T+12}^{(n)} = \hat{b}_0 + \hat{B}p_T,$$

with $\hat{b}$ and $\hat{B}$ the OLS estimates. However, if p_t does not forecast $xr_{t:t+12}$, including it in the regression increases the variance of parameter estimates, leading to less accurate forecasts, a higher MSPE and thus a lower R^2_{oos} (Heij et al., 2004). If p_t does not forecast $xr_{t:t+12}$, Equation (4) becomes

$$xr_{t:t+12}^{(n)} = b_0 + \eta_t,$$

for $t = 1, \ldots, T-12$. The OLS estimate of b_0 is then simply the historical mean of $xr_{t:t+12}^{(n)}$, which in turn is also the optimal forecast of $xr_{T:T+12}^{(n)}$.[18] Furthermore, even if variables do forecast excess returns, increasing the number of explanatory variables makes it harder to accurately estimate parameters. As a result, it is possible that this increases rather than decreases prediction error. For this reason, a model that produces forecasts that are positively correlated with realized

[17] The differences have been taken with respect to a year earlier to match the fact that forecasts are made for one year ahead. Untabulated results illustrate that using one month first differences to forecast one month ahead leads to the same conclusions.

[18] The intuition behind this is as follows. If the model has no forecasting power, we have no information about the value of the excess return. Then any forecast that deviates from the mean by d can (at most) reduce the forecast error by d compared to the benchmark or increase the forecast error by d compared to the benchmark. However, as we square the forecast error, a decrease in d is not as beneficial for the squared prediction error as an increase in d is detrimental to it.

returns can still produce a negative R^2_{oos}. We propose a Bayesian solution to this issue. In the frequentist approach, it is assumed that everything we know about the future excess return $xr_{t:t+12}$ is captured by x_t. In a Bayesian fashion, we argue that even before observing x_t, we already have some information about $xr_{t:t+12}$. Excess returns are mean reverting and thus ex ante, an excess return very far from the mean is less likely than an excess return very close to the mean. As machine learning techniques are model-free, they do not allow for the usual Bayesian approach of specifying a probability distribution and putting a prior distribution on its parameters. Instead, we directly specify the posterior prediction as a weighted average of the prediction based on the prior and the prediction based on the data:

$$\hat{xr}^{(n)}_{t-12:t} = w\ \hat{xr}^{(n)}_{t-12:t,p} + (1-w)\hat{xr}^{(n)}_{t-12:t,d},$$

where w is the weight on the prior, $\hat{xr}^{(n)}_{t-12:t,p}$ is the prior prediction and $\hat{xr}^{(n)}_{t-12:t,d}$ is the prediction based on the data. For the prior prediction we take the historical sample mean, which is also the benchmark of the Expectations Hypothesis. Thus, by imposing a prior we shrink the predictions to the benchmark, reducing the risk of very inaccurate predictions. We consider $w = 0.25$, 0.50 and 0.75.

3.7 Standard errors

As addressed in Section 3.6, a forecast that deviates from the benchmark is 'risky' in terms of model and parameter uncertainty. Shrinking the forecasts to a prior expectation is a solution in terms of point estimates, but the uncertainty also needs to be accounted for in the computation of standard errors. As such it is possible that a model with outspoken forecasts and a low R^2_{oos} is significantly better than the benchmark, whereas a more conservative model with a higher R^2_{oos} is not. We use the Clark and West (2007) statistic to adjust for the model and parameter uncertainty. This approach makes use of the fact that models are nested, such that parameter restrictions reduce one model to the more parsimonious other model. In our case, the parsimonious model is the benchmark prediction, which is equivalent to a regression model with only a constant. For both machine learning and linear techniques, it is possible to restrict the parameters such that the model reduces to the benchmark.[19]

Following Clark and West (2007), we define

$$\hat{\sigma}_1^{2(n)} = \frac{1}{T-12}\sum_{t=1}^{T-12}(xr^{(n)}_{t:t+12} - \bar{xr}^{(n)}_{t:t+12})^2,$$

[19] For a regression, this requires all parameters except the constant to be zero. For tree methods, this implies a depth of 0. For neural networks, this means weights of 0 at all nodes.

$$\hat{\sigma}_2^{2^{(n)}} = \frac{1}{T-12}\sum_{t=1}^{T-12}(xr_{t:t+12}^{(n)} - xr_{t:t+12}^{(n)}(M))^2$$

and the adjusted measure

$$\hat{\sigma}_{2,adj}^{2^{(n)}} = \hat{\sigma}_2^{2^{(n)}} - \frac{1}{T-12}\sum_{t=1}^{T-12}(\bar{xr}_{t:t+12}^{(n)} - xr_{t:t+12}^{(n)}(M))^2.$$

This adjusted squared error can be interpreted as the part of the squared forecast error that is not also present in the more parsimonious model. The null hypothesis of equal R^2_{oos}'s is rejected when $\hat{\sigma}_1^{2^{(n)}}$ sufficiently exceeds $\hat{\sigma}_{2,adj}^{2^{(n)}}$. We can test this by regressing

$$(xr_{t:t+12}^{(n)} - \bar{xr}_{t:t+12}^{(n)})^2 - (xr_{t:t+12}^{(n)} - xr_{t:t+12}^{(n)}(M))^2 + (\bar{xr}_{t:t+12}^{(n)} - xr_{t:t+12}^{(n)}(M))^2$$

on a constant and considering the t-statistic.[20] The autocorrelation resulting from the overlapping excess returns is taken into account by using HAC standard errors. We also use the Clark and West (2007) statistic to assess if adding macro variables to the models significantly improves predictions.

3.8 Pairwise tests of significance

It is not only relevant to determine if the techniques significantly outperform the benchmark, but also if they significantly outperform each other. Following Bianchi et al. (2020b), we use the Diebold and Mariano (1995) test statistic, with the changes proposed by Harvey, Leybourne, and Newbold (1997). The null hypothesis is equal predictive accuracy, with a two-sided alternative. We define the series

$$d_t = (xr_{t:t+12}^{(n)} - xr_{t:t+12}^{(n)}(M))^2 - (xr_{t:t+12}^{(n)} - xr_{t:t+12}^{(n)}(N))^2 \tag{5}$$

as the difference in squared errors of models M and N, with mean $\bar{d}$. The Diebold-Mariano statistic is then defined as

$$DM = [\hat{V}(\bar{d})]^{-\frac{1}{2}}\bar{d}, \tag{6}$$

with $\hat{V}(\bar{d})$ being an estimator of the variance of $\bar{d}$. This estimator is adjusted for the autocorrelation in the forecast errors, which results from the overlapping yearly excess returns. Specifically,

$$\hat{V}(\bar{d}) = \frac{T+1-2h+\frac{h}{T}(h-1)}{T^2}\left[\hat{\gamma}_0 + \frac{2}{T}\sum_{k=1}^{h-1}(T-k)\hat{\gamma}_k\right],$$

[20] Note that this term is simply $\hat{\sigma}_1^{2^{(n)}} - \hat{\sigma}_{2,adj}^{2^{(n)}}$ for individual observations.

with h being the number of periods between the point in time at which a forecast is made and the point in time at which the true value is observed, which is 12 in our case, and $\hat{\gamma}_k$ the sample estimate of the kth order autocovariance. Here, the term in the fraction is the finite-sample correction proposed by Harvey et al. (1997). The term in square brackets takes heteroskedasticity and autocorrelation into account and reduces to $\hat{\gamma}_0$ when those are not present (similar to the usual HAC standard errors).

3.9 Trading strategy

Statistically significant forecasting power does not necessarily translate in significant trading gains (Thornton and Valente, 2012). We therefore design trading strategies based on our forecasts and compare the resulting returns. Bond investors can make strategic decisions in terms of country, maturity and amount of investment. As our forecasts are made for a given country and maturity, we focus on investment strategies with varying amounts of investments.

3.9.1 Information Ratios

The Information Ratio (IR) is a measure of risk-adjusted return compared to a specific benchmark (Bacon, 2008). It is computed as

$$IR = \frac{\mathrm{E}[R_a - R_b]}{\sqrt{\mathrm{Var}[R_a - R_b]}},$$

with R_a the return on the active strategy and R_b the return on the benchmark strategy. A Sharpe Ratio (SR) is an IR with the risk free asset as benchmark. To determine an IR, we do not need to specify a benchmark explicitly. Rather, we determine the investment strategy relative to the benchmark. We interpret our forecast of an excess return as a signal for investment. If we forecast a high excess return, the position is long in the $\frac{n}{12}$ year bond and short in the 1 year bond. If we forecast a low excess return, the reverse is true. We calculate a z-score at time T as

$$z = \frac{xr_{T:T+12}(M) - \bar{xr}_{t:t+12}(M)}{\mathrm{std}(xr_{t:t+12}(M))}, \tag{7}$$

with $xr_{T:T+12}(M)$ the forecast of model M, $\bar{xr}_{t:t+12}(M)$ the historical average forecast of model M and $\mathrm{std}(xr_{t:t+12}(M))$ the historical standard deviation of model M. The position will thus always yield z more excess returns than the benchmark strategy. As z is 0 on average, on average the active strategy has the same duration[21] exposure as the benchmark. Higher returns are thus

[21] Duration is the average maturity of all bonds in a portfolio. As the yield curve is generally upward-sloping, more exposure to duration implies a higher expected return. Duration is thus in some sense the bond market counterpart to β in the equity market.

not the results of taking more risk. Furthermore, this IR can be interpreted as the Sharpe Ratio of a strategy that has zero duration exposure on average. In the literature, it is suggested that an IR above 0.5 can be classified as 'good' and an IR of 1 as 'exceptional' (Grinold and Kahn, 1992; Jacobs and Levy, 1996).

3.9.2 Sharpe Ratios

The Information Ratio can be interpreted as the Sharpe Ratio of a strategy that has 0 exposure to excess returns on average. However, excess returns are positive on average, and investors might prefer a strategy that does earn these excess returns. Such an investor is long in long maturity bonds and short in short maturity bonds, such that the portfolio has positive exposure to duration. A positive Information Ratio does not automatically imply an increase in Sharpe Ratio compared to a benchmark. It is also interesting to see if our forecasts can improve the Sharpe Ratio of a benchmark strategy that has a positive exposure to duration, and earns the corresponding risk premium. We therefore consider a benchmark investment strategy that is line with our research setting. This benchmark strategy is to buy a $\frac{n}{12}$ year bond and to short a 1 year bond every month. A year later the position is closed. At each point in time, the portfolio is thus long in 12 bonds with times to maturity $n, \ldots, n-11$ and short in 12 bonds with times to maturity $12, \ldots, 1$. As the yield curve is upward sloping, this strategy earns a positive return on average. If excess bond returns are not not predictable, this strategy is optimal and it thus corresponds to the Expectations Hypothesis benchmark forecast.

Again, we compute z as in (7). Then, our active investment strategy is to buy $1+z$ of a $\frac{n}{12}$ year bond and to short $1+z$ of a 1 year bond. The strategy is thus to invest a bit more than the benchmark if a high return is forecast, and a bit less if a low return is forecast. Again, the duration exposure is equal to the benchmark on average, and increases in return can thus not be attributed to more duration exposure. For robustness, we also consider investing $1 + 0.5z$ and $1 + 2z$.

3.9.3 Encompassing regressions

For a machine learning technique to add value for an investor does not necessarily require more accurate forecasts than existing approaches. If machine learning techniques pick up different patterns than linear approaches, forecasts may have low correlation and a combination of forecasts might outperform each individual forecast. We consider a regression of the returns of a machine learning based investment strategy on the returns of linear investment strategy and an intercept. Thus

$$R_{m,t} = \alpha + R_{l,t} + \varepsilon_t, \tag{8}$$

for time $t = 1, \ldots, T$, with $R_{m,t}$ the return on a ML based investment strategy, α the intercept, $R_{l,t}$ the return on a linear investment strategy and ε_t the error term. If machine learning techniques add something to linear approaches, the variation in their returns is not fully described by the variation in the linear investment returns. This should then translate to a large and significant alpha.

4 Data

For yield information, we use the yield data set of Liu and Wu (2019).[22] The macro data set is provided by McCracken and Ng (2016).[23] It consists of 128 variables, divided over the categories (1) output, (2) labor market, (3) housing sector, (4) orders and inventories, (5) money and credit, (6) exchange and interest rates, (7) prices or price indices and (8) stock market. The authors provide an extensive description of the variables and the transformations that have been used. We make use of the final data set of January 2019 as well as vintage data sets starting in August 1998. A full list of the macro variables and their factor loadings on the first 8 principal components can be found in Tables 21 and 22 in Appendix A. Equity returns are obtained from the MSCI website and GSCI returns from Bloomberg.

Plots of the dependent variables, excess returns, are shown in Figure 4 for $n = 24$ and $n = 120$. The plots show that different excess returns strongly move together. This is also visible in the across-maturity correlations, which range between 82 and 99%. As also visible in the plots, the excess returns on high maturity bonds are more volatile than those on low volatility bonds, and also higher on average.[24] The forward rates are shown in Figure 5. They also strongly move together, with correlations between 80 and 99%. There is an upward trend until approximately 1985, and a downward trend thereafter. Descriptive statistics of excess returns, forward rates and changes in forward rates are given in Table 29 in Appendix A.

[22] Available at `https://sites.google.com/view/jingcynthiawu/`.

[23] Available at `https://research.stlouisfed.org/econ/mccracken/fred-databases/`.

[24] This supports the interpretation of duration as proxy for exposure to market risk.

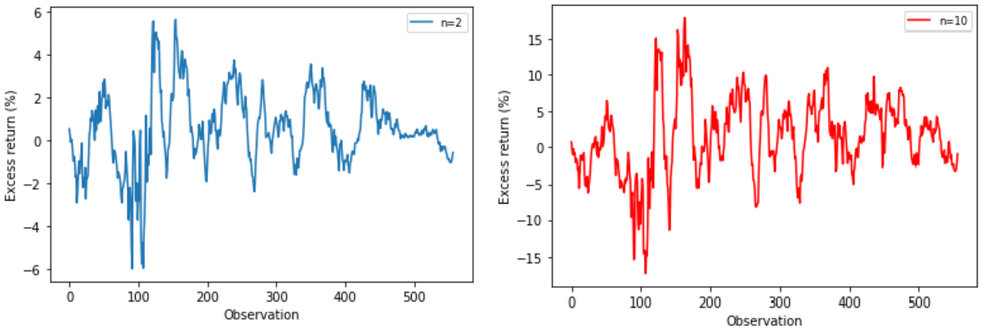

Figure 4: Plots of the 2 year (left) and 10 year (right) excess bond returns for the period 1971:08-2018:12.

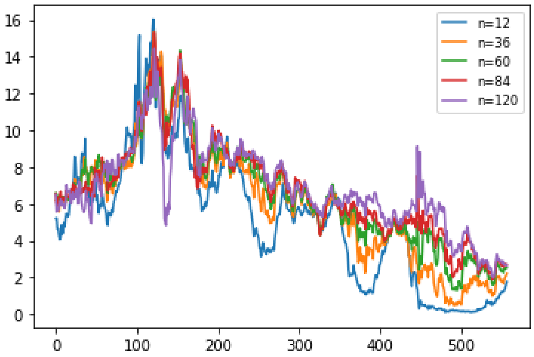

Figure 5: A plot of the one year forward rates (in %) for the period 1971:08-2018:12, for times to maturity of 12, 36, 60, 84 and 120 months.

5 Results

In Section 5.1, we discuss the impact of using macro data that is available in real time. The contrast between stationary and nonstationary input data is discussed in Sections 5.2. Section 5.3 is on model uncertainty and Bayesian shrinkage. The results for Information Ratios and Sharpe Ratios constitute Sections 5.4 and 5.5. The results of the encompassing regressions are given in Section 5.6. Finally, fully tabulated results of methods and techniques considered are provided in Tables 23-26 in Appendix A.

5.1 Real time macro data

The out-of-sample R^2's of forecasts using forward rates and both real time and final macro information are shown in Table 1. For neural networks, the R^2's are substantially reduced across all maturities when only using real time information. For 24 and 36 month bonds, the R^2's are no longer positive. For $n = 48$, the R^2 is still lower than in the yields-only case as reported in Bianchi et al. (2020b), such that the real time macro data does not contain predictive power beyond the forward curve. On the long end of the curve, neural networks still have strong predictive power, and also overall they are the best performing method. Nevertheless, it is an important finding that the forecasting power of neural networks is reduced in real time.

Macro data	Method	n=24	n=36	n=48	n=60	n=84	n=120	EW
Final	Regression	-46.0°	-36.8°	-28.3°	-7.9°	-7.5°	-1.6°	-10.5°
Vintage	Regression	-39.4°	-24.4°	-14.6°	-7.9°	-6.2°	1.7**	-7.1°
Final	Random forest	-20.7°	-19.9°	-20.6°	-19.3°	-20.3°	-18.7°	-20.7°
Vintage	Random forest	-22.7°	-19.4°	-20.0°	-16.2°	-13.9°	-8.7°	-15.7°
Final	Neural network	2.0**	9.9***	12.4***	15.4***	17.3***	20.9***	16.8***
Vintage	Neural network	-13.2°	-1.5°	2.3**	6.7**	10.6**	17.1***	9.0**

Table 1: The out-of-sample R^2 (in %) for the forecasting of excess bond returns for times to maturity n =24, 36, 48, 60, 84 and 120 and an equally weighted portfolio of those. Linear regressions, random forests and neural networks are considered. The explanatory variables are the forward rates $f_t^{(n)}$ in combination with (principal components of) a set of 128 (vintage or final) macro variables.

Also for regressions and random forests, there are some differences. However, comparing negative R^2's is not very meaningful, as they imply the methods offer worse predictions than the historical mean. By simply shrinking the methods to the mean, they always improve.[25]

5.2 Nonstationarity

A plot of the forward rates over the full sample period is given in Figure 5. The plot provides a clear indication that the forward rates are nonstationary, with an upward trend until the middle of the 1980s, and a downward trend thereafter. We formally test for nonstationarity using an Augmented Dickey-Fuller Test (Dickey and Fuller, 1979). To keep the analysis feasible in real time, we only consider the in-sample period.[26] As the mean of $f_t^{(n)}$ is not zero, we include a constant in the ADF regression,

$$f_t^{(n)} = \alpha + \beta t + \rho f_{t-1}^{(n)} + \sum_{j=1}^{k-1} \delta_j \Delta f_{t-j}^{(n)} + v_t,$$

where α is the drift term, β is the slope of the trend and v_t an error term. The null hypothesis of a unit root corresponds to $\rho = 0$ and the alternative hypothesis of stationarity to $\rho < 1$. We also consider the case without trend, corresponding to $\beta = 0$. The number of lags is chosen to minimize the Akaike Information Criterion.

The results of both tests are displayed in Table 2. Nonstationarity is rejected for none of the forward rates at any conventional significance level. To obtain more insight into why the linear approach fails when using forward rates, we consider the plot of its forecast in Figure 6. Remarkably, the forecasts show a strong downward bias. The nonstationarity and downward

[25] Contrary to methods with positive R^2's, which can improve by shrinkage, but do not necessarily always do so.

[26] For completeness, results for the full sample are given in Table 19 in Appendix A. Conclusions are similar in this case.

	No trend			Trend		
n	DF	p-value	Lags	DF	p-value	Lags
12	-2.23	0.194	11	-2.13	0.530	11
24	-2.04	0.271	11	-1.92	0.643	11
36	-1.84	0.362	0	-1.72	0.741	0
48	-1.83	0.364	0	-1.69	0.754	0
60	-1.73	0.414	1	-1.57	0.805	1
72	-1.84	0.358	6	-1.72	0.742	6
84	-1.73	0.417	0	-1.54	0.817	15
96	-1.72	0.422	1	-1.46	0.844	1
108	-1.71	0.425	0	-1.62	0.784	0
120	-2.56	0.102	2	-2.78	0.206	2

Table 2: This table reports the results of an Augmented Dickey-Fuller Test for a unit root in the n month forward rates in the period 1971:08-1989:12. The ADF test statistic is computed as $\frac{\hat{\rho}}{\text{se}(\hat{\rho})}$, where $\hat{\rho}$ is the estimate of ρ and se$(\hat{\rho})$ its standard error.

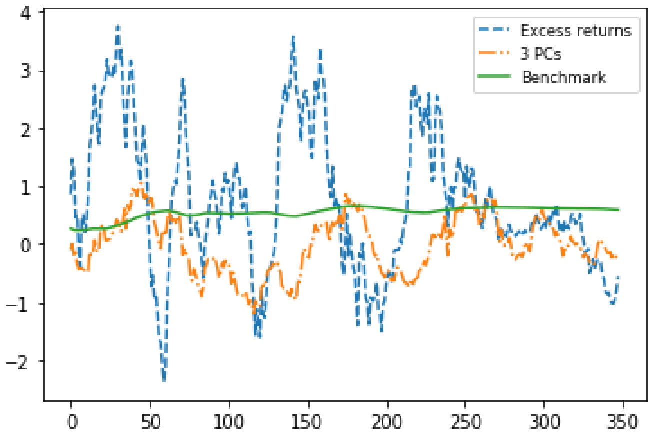

Figure 6: Plots of forecast and realized excess returns on 24-month treasury bonds over 1990:01-2018:12. The principal-component regression forecasts are made using the first three principal components of the forward rates. The benchmark is the historical sample mean.

trend thus dominate the principal-component regression forecast in Figure 6, leading to persistent underestimation of the excess return. It is no surprise that this forecast yields a negative R^2_{oos}. The fact that the models forecasts are almost always below the mean are a cause for concern, as the mean itself is stable. It is no wonder neural networks outperform this linear approach. Rather than dismissing linear regressions as a forecasting approach altogether, we investigate whether nonstationarity is an underlying issue that frustrates the methodology. The results of an Augmented Dickey-Fuller test for the changes in forward rates $\Delta f_t^{(n)}$ can be found in Table 20 in Appendix A. A unit root is rejected at a 1% significance level in all cases.

The results of forecasts using $f_t^{(n)}$ or $\Delta f_t^{(n)}$, possibly in combination with real time macro data, are displayed in Table 3. The results in rows 1-3 and 7-9 are obtained in a very similar fashion as those reported in Bianchi et al. (2020a). The findings are also comparable, with regressions and random forests producing negative R^2's, whereas neural networks generally produce positive R^2's and constitute the best performing method.[27] Comparing these results to the first three rows of Table 23, we find, as Bianchi et al. (2020a), that neural network forecasts improve on the long end of the curve when including macro information.

Input	Method	n=24	n=36	n=48	n=60	n=84	n=120	EW
$f_t^{(n)}$	Regression	-31.5°	-23.0°	-16.5°	-8.4°	-5.6°	3.3**	-8.3°
	Random forest	-19.1°	-16.4°	-15.0°	-11.9°	-12.0°	-8.9°	-13.1°
	Neural network	-2.1°	0.9°	1.6°	2.4°	3.1°	3.6*	2.7°
$\Delta f_t^{(n)}$	Regression	19.9***	19.7***	18.0***	15.7***	13.9***	12.5***	16.0***
	Random forest	-6.8°	-6.5°	-7.2°	-7.4°	-10.4°	-12.9°	-10.4°
	Neural network	-1.5°	3.1**	4.6**	5.4**	5.3**	6.6**	6.0**
$f_t^{(n)}$ + real time macro	Regression	-39.4°	-24.4°	-14.6°	-7.9°	-6.2°	1.7**	-7.1°
	Random forest	-22.7°	-19.4°	-20.0°	-16.2°	-13.9°	-8.7°	-15.7°
	Neural network	-13.2°	-1.5°	2.3**	6.7**	10.6**	17.1***	9.0**
$\Delta f_t^{(n)}$ + real time macro	Regression	10.7***	14.1***	17.7***	19.1***	18.6***	23.5***	18.9***
	Random forest	-18.5°	-10.7°	-6.8°	-1.8°	1.3**	7.2**	-0.5°
	Neural network	-34.5°	-14.4°	-8.4°	-2.2°	1.6**	10.5***	-1.2°

Table 3: The out-of-sample R^2 (in %) for the forecasting of excess bond returns for times to maturity n =24, 36, 48, 60, 84 and 120 and an equally weighted portfolio of those. Linear regressions, random forests and neural networks are considered. The explanatory variables are the forward rates $f_t^{(n)}$ or the first differences of the forward rates $\Delta f_t^{(n)}$, and (the principal components of) a set of 128 real time macro variables. *, ** and *** indicate that the R^2 is significantly larger than 0 at 10, 5 or 1% significance level respectively, and ◦ indicates no significance.

Overall, we find a remarkable result. When replacing the forward rates with the changes in forward rates, all methods show improvements (with the exception of neural networks in forwards + macro case). However, the rankings have now changed. With stationary input data, linear regressions constitute the best rather than worst performing method, across all maturities and in both the forwards only and the forwards + macro setting. The R^2's of linear regressions are highly significant, again across all maturities and settings. Thus, replacing the forward rates in principal-component regressions with the changes in forward rates leads to significant outperformance of the zero-predictability benchmark. This provides evidence against the Expectations Hypothesis.

The principal-component regression also outperforms all machine learning techniques, both

[27] There are some differences between the R^2_{oos}'s that we report and those reported in Bianchi et al. (2020a). For example, for the yields-only neural network configuration with 1 hidden layer and 3 nodes we have used here, they find R^2's of [0.026, 0.043, 0.045, 0.044, 0.044, 0.038, 0.044]. This difference can be attributed to differences in the hyperparameter search, most importantly for the regularization parameter. In this setting, the authors let the network choose between a value of 0.5 and 1 and in the macro setting between 0.001 and 0.01. However, we do not know ex ante what is the optimal grid for the network to choose from. We therefore perform a grid search over [0.001, 0.01, 0.1 and 1]. Furthermore, differences also arise as a results of the randomness in ML techniques. In the macro setting, neural network results can vary more in different configurations (number of hidden layers and number of nodes per layer) due to the use of a much larger data set. On average, those neural network configurations yield R^2's (ascending in maturity; %) of [-8.0, 3.5, 5.6, 9.1, 11.2, 16.0, 10.5]. That is lower than what we find with final macro data (as shown in Table 24 in Appendix A. However, for the specific configuration considered here (1 hidden layer of 32 nodes), they find R^2's of [3.4, 3.2, 8.6, 6.7, 21.1, 28.2, 19.6], higher than found by us. Such differences are not necessarily meaningful. In Tables 38 in Appendix A, we display neural network results for different configurations, which do not change our conclusions.

those using forward rates and those using the changes in forward rates as input, and with or without macro information. The results of pairwise tests of significance for 2 and 10 year excess returns are shown in Tables 27 and 28 in Appendix A. Linear regressions significantly outperform random forests and neural networks for at least one of the cases. Across-setting comparisons of methods are untabulated, but show an even stronger rejection of equal predictive power.[28] Interestingly and contrary to the findings of Bianchi et al. (2020a), with stationary input variables, random forests are able to generate positive predictive power when including macro information.

In the yields-only case, the principal-component regressions also outperform all 12 neural network configurations considered in Bianchi et al. (2020a), of which the R^2_{oos} ranges between 2.6 and 6.8%. It appears that neural networks do not necessarily outperform regressions, but rather that for them, nonstationary input data is not as problematic as it is for linear techniques. Random forests still fail to yield significantly positive R^2's, even with stationary input data. In the setting including macro information, the principal-component regression outperforms all neural networks considered in Bianchi et al. (2020a) for at least one maturity, and often for all maturities, despite the fact that the latter have been obtained using final data.

To determine if macro information adds something to the forecasts, we compare rows 4-6 to 10-12 of Table 3. All methods show clear improvements for the long maturities. The Clark and West (2007) statistic indicates all improvements in the R^2_{oos}'s of the regressions are statistically significant at a 5% level.[29] This is evidence for the existence of unspanned macro factors. However, the R^2's decrease on the short end, especially without a prior. This suggests that macro information is helpful only for predicting movements of bonds with long maturities. An explanation for this is that the prices of these bonds depend more on long term expectations than short maturity bonds. In turn, long term expectations are arguably more influenced by macro information.

5.3 Model uncertainty & Bayesian shrinkage

The R^2_{oos}'s of the methods including Bayesian shrinkage are shown in Table 4 for prior weights of 0.25 and 0.5.[30] Comparing these to the results in Table 3, all methods improve by imposing a prior of 0.25 on the forecasts, except the neural network with only forward rates. Thus, shrinking the forecasts a bit to the mean decreases their mean squared errors. Intuitively, this implies that

[28] This can be explained by the lower correlation in predictions across settings.

[29] For the machine learning techniques, the first setting is not truly nested in the second setting, and we thus cannot use the Clark and West (2007) statistic.

[30] Results for a prior weight of 0.75 are reported for in Tables 23-26 in Appendix A for completeness. Generally, a weight of 0.75 leads to R^2's closer zero, which is to be expected with such a large weight.

the model forecasts are a bit too outspoken. The forecasts improve by taking model uncertainty into account. The fact that neural network forecasts do not improve in the yields-only setting can be explained by the use of forecast averaging and regularization. As a result, their forecasts are already not that outspoken. Overall, the improvements are much larger when including macro variables than when only considering forward curve information. This is logical, as with a much larger data set there is a higher risk of overfitting.

Comparing regressions to machine learning techniques, we again observe that regressions clearly outperform random forests and neural networks, in both settings and across all maturities. Interestingly, the results also indicate that when taking model uncertainty into account, regressions do achieve some positive and significant R^2's even with nonstationary input variables. As shown in Table 25 in Appendix A, the R^2_{oos}'s of regressions are positive and significant across all maturities when a prior weight of 0.75 is imposed. In fact, when using only forward rates as input variables, all yields-only network configurations are outperformed for $n = 120$, both those reported here and those in Bianchi et al. (2020a).

Input	Method	Prior weight	n=24	n=36	n=48	n=60	n=84	n=120	EW
$f_t^{(n)}$	Regression	0.25	-17.3°	-12.3°	-7.1°	-1.3°	2.2*	9.8**	0.0°
	Random forest	0.25	-13.8°	-11.9°	-10.8°	-8.5°	-8.6°	-6.2°	-9.4°
	Neural network	0.25	-1.0°	1.0°	1.6°	2.1°	2.7°	3.0*	2.4°
	Regression	0.5	-7.4°	-5.0°	-1.3°	2.4°	5.7*	11.4**	4.1*
	Random forest	0.5	-8.9°	-7.6°	-6.9°	-5.4°	-5.4°	-3.8°	-5.9°
	Neural network	0.5	-0.3°	0.9°	1.3°	1.6°	2.0°	2.2*	1.8°
$\Delta f_t^{(n)}$	Regression	0.25	20.0***	19.4***	18.2***	16.1***	14.8***	13.4***	16.6***
	Random forest	0.25	-1.2°	-1.2°	-1.7°	-2.1°	-4.5°	-6.7°	-4.3°
	Neural network	0.25	0.3°	3.5**	4.6**	5.1**	5.0**	6.0**	5.7**
	Regression	0.5	16.7***	16.1***	15.2***	13.6***	12.8***	11.6***	14.2***
	Random forest	0.5	1.8*	1.7°	1.3°	0.9°	-0.8°	-2.4°	-0.5°
	Neural network	0.5	1.1°	3.1**	3.8**	4.0**	4.1**	4.7**	4.6**
$f_t^{(n)}$ and macro (real time)	Regression	0.25	-20.6°	-11.4°	-5.2°	-0.2°	2.1*	9.1**	1.4*
	Random forest	0.25	-13.5°	-12.3°	-13.0°	-10.7°	-9.3°	-5.9°	-10.4°
	Neural network	0.25	-3.9°	3.8*	6.2**	9.4**	12.2**	17.1***	11.4**
	Regression	0.5	-7.7°	-3.0°	0.4*	3.6*	5.9*	11.2**	5.4*
	Random forest	0.5	-6.7°	-6.7°	-7.3°	-6.1°	-5.4°	-3.4°	-6.0°
	Neural network	0.5	1.4°	5.8*	7.2**	9.2**	11.0**	14.3***	10.7**
$\Delta f_t^{(n)}$ and macro (real time)	Regression	0.25	13.7***	15.5***	18.2***	19.0***	19.5***	23.5***	19.6***
	Random forest	0.25	-8.1°	-2.8°	0.2*	3.6*	6.0**	10.1**	4.9**
	Neural network	0.25	-13.6°	-0.8°	3.2**	7.3**	10.5**	16.9***	8.9**
	Regression	0.5	12.9***	13.7***	15.4***	15.8***	16.7***	19.6***	16.7***
	Random forest	0.5	-1.5°	1.7°	3.7*	5.7*	7.3**	9.8**	6.8**
	Neural network	0.5	-0.9°	6.1*	8.5**	10.8**	13.2**	17.2***	12.5**

Table 4: The out-of-sample R^2 (in %) for the forecasting of excess bond returns for times to maturity n =24, 36, 48, 60, 84 and 120 and an equally weighted portfolio of those. Linear regressions, random forests and neural networks are considered with a prior weight of 0.25 or 0.5 on the historical sample mean. The explanatory variables are the forward rates $f_t^{(n)}$ or the first differences of the forward rates $\Delta f_t^{(n)}$. *, ** and *** indicate that the R^2 is significantly larger than 0 at 10, 5 or 1% significance level respectively, and $\circ$ indicates no significance.

5.4 Information Ratios

The Information Ratios of investment strategies based on forecasts of linear regressions, random forests and neural networks are shown in Table 5. Due to the conversion to z-scores, the prior weight only has a very small impact on the investment strategies. We therefore only report results based on forecasts without shrinkage. In both settings and for all maturities, regression forecast signals produce the highest IRs. In line with the results found for R^2_{oos}'s, the best results are obtained using both forward and macro information. Compared to the yields-only setting, the regression improves mostly on the long end of the curve. The IRs of the random forests also strongly improve when including macro information.

Setting	Method	n=24	n=36	n=48	n=60	n=84	n=120	EW
$\Delta f_t^{(n)}$	Regression	0.68	0.62	0.55	0.49	0.43	0.39	0.47
	Random forest	0.18	0.18	0.16	0.14	0.08	0.04	0.09
	Neural network	0.14	0.13	0.13	0.11	0.08	0.07	0.10
$\Delta f_t^{(n)}$ and macro	Regression	0.74	0.74	0.77	0.74	0.68	0.67	0.73
	Random forest	0.53	0.50	0.49	0.47	0.44	0.44	0.47
	Neural network	0.19	0.18	0.17	0.15	0.15	0.14	0.16

Table 5: The Information Ratios for timing strategies with no duration exposure on average. Excess bond returns are considered for times to maturity n =24, 36, 48, 60, 84 and 120 and an equally weighted portfolio of those. Model forecasts are interpreted as signal and a z-score is computed. Every month, an investment equal to the z-score is done and positions are held for twelve months, such that the portfolio consists of 12 bonds at each point in time. The explanatory variables are (1) the first differences of the forward rates $\Delta f_t^{(n)}$, (2) $\Delta f_t^{(n)}$ in combination with (principal components of) a set of 128 macro variables or (3) yield spread s_t, 12 month bond momentum $d_{t-12:t}^{(n)}$, 12 month equity returns $q_{t-12:t}$ and 12 month commodity returns $c_{t-12:t}^{(n)}$.

Interestingly, neural networks constitute the worst performing method in terms of IR, whereas their R^2_{oos}'s were generally higher than those of random forests and sometimes close to those of regressions. We investigate this further by considering the plot of regression forecasts and neural network forecasts in Figure 7. The plot shows that the neural network forecasts are often close to the mean and predictive power is concentrated in a small subset of observations. The regression forecasts appear to be more thin-tailed. This is also visible in the kurtoses, which are -0.47 and 0.83 for the regression and neural network forecasts respectively. The fact that the predictive power can be concentrated in a few observations might explain why a high R^2_{oos} does not necessarily translate to a high IR.

5.5 Sharpe Ratios

The Sharpe Ratios of investment strategies based on the three model forecasts are shown in Table 6. In each setting, the regression-based strategy produces the highest Sharpe Ratios of the three methods. For all maturities, linear forecasts using only yield information produce large

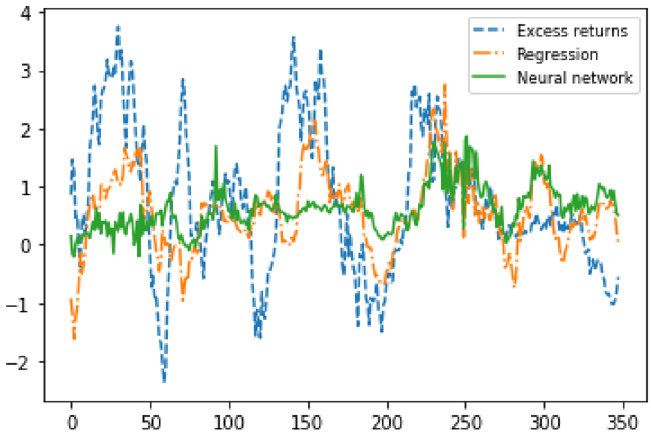

Figure 7: Forecasts of excess returns with time to maturity $n = 24$ with linear regressions and neural networks using $\Delta f_t^{(n)}$.

improvements in Sharpe Ratios compared to the benchmark. Also including macro information increases all Sharpe Ratios a bit more, again mostly for long maturities. Random forests and neural networks also provide improvements compared to the benchmark.

Setting	Method	n=24	n=36	n=48	n=60	n=84	n=120	EW
Benchmark	-	0.84	0.75	0.70	0.64	0.58	0.53	0.63
$\Delta f_t^{(n)}$	Regression	0.92	0.82	0.76	0.69	0.63	0.57	0.69
	Random forest	0.86	0.76	0.71	0.63	0.55	0.49	0.61
	Neural network	0.87	0.78	0.73	0.65	0.58	0.53	0.64
$\Delta f_t^{(n)}$ and macro	Regression	0.93	0.83	0.78	0.71	0.65	0.61	0.71
	Random forest	0.84	0.75	0.71	0.64	0.58	0.54	0.65
	Neural network	0.85	0.77	0.72	0.65	0.59	0.55	0.64

Table 6: The Sharpe Ratios for timing strategies with average positive duration exposure. The benchmark strategy is to buy a $\frac{n}{12}$ year bond and to sell a 1 year bond every month. After a year, the former is sold and the latter has expired, such that the portfolio consists of 12 bonds at each point in time. For the active strategies, model forecasts are interpreted as signal and a z-score is computed. Investment equals $1 + z$. The explanatory variables are the first differences of the forward rates $\Delta f_t^{(n)}$ and (principal components of) a set of 128 macro variables.

The improvements in Sharpe Ratios might seem modest compared to the rather large IRs. This can be explained by two reasons. First of all, simply earning the excess returns is a tough benchmark to beat. As explained in Section 6.1, holding a long position in a long term bond and a short position in a short term bond is expected to yield a positive return due to the carry effect. Secondly, our strategy only allows for variation in the amount of investment, not in the maturity of the bond or the country that issues them. There is thus limited room to exploit the forecasts. Moreover, varying the amount of investment can be expected to increase the volatility of the returns in the time series dimension,[31] which has a negative effect on the Sharpe Ratio.

Finally, it must be mentioned that the Sharpe Ratios are dependent on the amount of weight the investor allocates to the z-scores. For example, an investor can also choose to invest $1 + 0.5z$

[31] Unless of course, we can almost perfectly predict returns.

or $1+2z$ in bonds, rather than $1+z$. Table 35 in Appendix B shows that our results are robust to such decisions, with linear regressions outperforming the benchmark and the machine learning techniques. Only when taking extreme positions (such as $1+4z$) will the Sharpe Ratio of the linear regression strategy decrease below the benchmark. This is logical: the more weight is put on the z score, the more the SR starts to resemble an IR. Given that the IRs are below the SRs, this produces poorer outcomes.

5.6 Encompassing regressions

Correlations between the forecasts of linear regressions, random forests and neural networks are shown in Table 7. The correlations are high and positive, ranging between 0.5 and 0.75. The forecasts of all techniques are thus generally in the same direction, suggesting that they pick up similar patterns. If machine learning techniques were to find hidden non-linear patterns in the data, undetectable by linear techniques, we would expect a rather low correlation between forecasts.

Setting	Methods	n=24	n=36	n=48	n=60	n=84	n=120	EW
$\Delta f_t^{(n)}$	Regression & R. forest	0.56	0.57	0.57	0.57	0.56	0.55	0.56
	Regression & NN	0.56	0.61	0.63	0.64	0.66	0.67	0.65
	R. forest & NN	0.51	0.56	0.57	0.58	0.61	0.62	0.60
$\Delta f_t^{(n)}$ and macro	Regression & R. forest	0.57	0.56	0.59	0.61	0.61	0.60	0.61
	Regression & NN	0.41	0.43	0.48	0.50	0.53	0.52	0.51
	R. forest & NN	0.79	0.78	0.79	0.78	0.78	0.77	0.78

Table 7: Correlations between forecasts of excess returns of regressions, random forests and neural networks. The explanatory variables are (1) the first differences of the forward rates $\Delta f_t^{(n)}$, (2) $\Delta f_t^{(n)}$ in combination with (principal components of) a set of 128 macro variables or (3) yield spread s_t, 12 month bond momentum $d_{t-12:t}^{(n)}$, 12 month equity returns $q_{t-12:t}$ and 12 month commodity returns $c_{t-12:t}^{(n)}$.

In Table 8, the alphas are shown of regressions of returns of investment strategies based on machine learning strategies on returns of investment strategies based on linear regression forecasts. All alphas are negative, or positive but insignificant at a 5% level. In Table 30 in Appendix A, the results are shown of the reversed analysis, regressions of linear regression returns on machine learning returns. Those results show positive alphas for linear strategies, generally significant at a 5% level and always at a 10% level.

The correlations and alphas in Tables 7, 8 and 30 allow us to draw conclusions regarding linear vis-a-vis machine learning techniques. Forecasts of the different techniques are highly correlated and in part they capture the same patterns. Machine learning techniques do not have predictive power beyond what is captured by linear techniques, but reversely, linear techniques do detect significant alpha beyond what is found by machine learning techniques.

Setting	Method	n=24	n=36	n=48	n=60	n=84	n=120	EW
$\Delta f_t^{(n)}$	Random forest	-0.01	-0.01	-0.02	-0.02	-0.04	-0.06	-0.03
		(0.02)	(0.06)	(0.08)	(0.08)	(0.03)	(0.01)	(0.02)
	Neural network	0.00	0.00	0.00	0.00	0.00	-0.01	0.00
		(0.70)	(0.66)	(0.63)	(0.88)	(0.68)	(0.55)	(0.87)
$\Delta f_t^{(n)}$ and macro	Random forest	0.00	0.00	-0.01	-0.01	-0.02	-0.03	-0.01
		(0.74)	(0.93)	(0.56)	(0.41)	(0.36)	(0.33)	(0.39)
	Neural network	0.01	0.02	0.01	0.00	-0.01	-0.01	0.00
		(0.11)	(0.24)	(0.76)	(0.93)	(0.86)	(0.77)	(0.97)

Table 8: Estimated alphas in a regression of machine learning investment returns on linear regression investment returns. P-values for a two-sided test of alpha= 0 are given between brackets. The explanatory variables are the first differences of the forward rates $\Delta f_t^{(n)}$ and (principal components of) a set of 128 macro variables.

6 Additional results and robustness

In this section we present various additional results and robustness checks. In Section 6.1, we repeat our research with a different set of explanatory variables. Following Baltussen et al. (2020), we forecast excess returns using yield spread, momentum, equity returns and commodity returns. Section 6.2 is on time-varying prior weights, depending on model fit and similarity of observations. We discuss a mean-variance utility setting in Section 6.3 and the interpretation of principal components in Section 6.4. Finally, Section 6.5 is on the robustness checks.

6.1 Yield spread, momentum, equities & commodities

A disadvantage of using a large set of macro variables is the lack of interpretation, especially as the models are re-estimated every month, such that the influence of variables can be time-varying. We therefore consider a third setting, in which four variables are used for which the relation with future bond returns has clear economic interpretation. We adopt these variables from Baltussen et al. (2020). First, we consider the yield spread s_t, defined as the yield of a 10 year bond minus the yield of a 3 month bond. The yield spread has been documented to be a predictor of bond curve movements by Dyl and Joehnk (1981), Campbell and Shiller (1991) and many others. A high yield spread signals high expected excess returns for three reasons. Firstly, a high yield spread implies the difference in expected return between long term bonds and short term bonds is large. Secondly, when the yield spread is large, it is generally upward-sloping and an effect referred to as the roll-down is present. This implies that when an investor holds the bond, its yield 'rolls down' the curve. At the point of selling, a lower yield is required, such that the price of the bond becomes higher relative to its pay-off. These two effects are most clearly visible when the yield curve stays the same, and their combination is also called the carry. Thirdly, the yield spread tends to exhibit some mean reversion. A high yield spread signals lower yield spreads later, and outstanding bonds increase in value when yields fall (Martens et al., 2019).

For the second explanatory variable, we turn to Figure 4, which shows plots of the 2 and 10 year excess bond returns for the full sample period. The plots indicate strong autocorrelation, which is still present in the 12th lag. The twelfth order autocorrelations for the in-sample period are shown in Table 9. For completeness, results for the full sample are shown in Table 18 in Appendix A. To incorporate this strong autocorrelation, we include a momentum dummy variable $d^{(n)}_{t-12:t}$ in the analysis. This dummy variable indicates if the previous excess return $xr^{(n)}_{t-12:t}$ has been below or above average. In regressions, we use maturity-specific momentum, whereas machine learning techniques use the momentum of the equally weighted return.[32] Related variables have been studied by e.g. Cutler et al. (1991) and Ilmanen (1997).

Time to maturity	24	36	48	60	84	120	EW
ρ_{12}	0.209	0.176	0.164	0.135	0.092	0.049	0.114

Table 9: The twelfth order autocorrelation in excess returns over the period 1971:08-1990:12. We consider bonds with a time to maturity of 24, 36, 48, 60, 84 and 120 months and an equally weighted portfolio of those.

Our third variable is the past 12 month equity return $q_{t-12:t}$, which was proposed by Ilmanen (1995). The rationale is that investor risk aversion is decreasing in wealth. Negative past stock returns imply investors currently have low wealth and are risk averse. Therefore, they invest more in safe government bonds and drive up prices, such that negative equity returns are a positive signal for excess bond returns. Returns are taken from the MSCI US index. Fourthly, we consider the past 12 month commodity returns $c_{t-12:t}$. Negative commodity returns signal low inflation, which makes nominal bonds more safe, thus more attractive, and increases bond returns (Baltussen et al., 2020). Commodity returns are obtained from the S&P GSCI index.

In the third setting, Equation (2) reduces to

$$xr_{t:t+12}(n) = k_0 + k_1 s_t + k_2 d^{(n)}_{t-12:t} + k_3 q_{t-12:t} + k_4 c_{t-12:t} + \eta_t,$$

where k_j are coefficients for $j = 0, \ldots, 4$.

6.1.1 Results

The results of predictions based on yield spread, momentum, equity returns and commodity returns are shown in Table 10. Again, regression is the best performing technique, with strong predictive power especially for the long maturities. Random forests and neural networks fail to generate significant predictive power for all maturities except for $n = 120$. For all techniques,

[32] This difference stems from the fact that the machine learning techniques are a mapping to all excess returns jointly. Interestingly, specifying a mapping for each maturity individually strongly deteriorates the results.

including a prior leads to large improvements in predictive performance. This again underlines that all methods are subject to some overfitting.

Method	Prior weight	n=24	n=36	n=48	n=60	n=84	n=120	EW
Regression	-	-9.1°	-5.0°	-4.7°	-1.5°	3.8**	11.9***	1.3**
Random forest	-	-17.7°	-13.0°	-10.6°	-7.1°	-4.0°	1.7°	-5.9°
Neural network	-	-8.7°	-7.1°	-6.2°	-4.5°	-3.2°	0.6*	-3.8°
Regression	0.25	-0.9°	1.9*	2.9*	5.4**	10.0**	17.4***	8.3**
Random forest	0.25	-10.6°	-7.2°	-5.1°	-2.6°	0.2°	4.7°	-1.2°
Neural network	0.25	-5.6°	-4.5°	-3.6°	-2.4°	-1.2°	1.7*	-1.7°
Regression	0.5	3.4*	5.1*	6.2*	8.0*	11.4**	17.2***	10.4**
Random forest	0.5	-5.3°	-3.1°	-1.5°	0.1°	2.2°	5.4°	1.3°
Neural network	0.5	-3.2°	-2.4°	-1.8°	-1.0°	-0.1°	2.0*	-0.4°

Table 10: The out-of-sample R^2 (in %) for the forecasting of excess bond returns for times to maturity n =24, 36, 48, 60, 84 and 120 and an equally weighted portfolio of those. Linear regressions, random forests and neural networks are considered. The explanatory variables are the yield spread s_t, 12 month bond momentum $d^{(n)}_{t-12:t}$, 12 month equity returns $q_{t-12:t}$ and 12 month commodity returns $c^{(n)}_{t-12:t}$.

The best results are generally obtained with a prior weight of 0.5. With these prior weights, random forests and neural networks are significantly outperformed by regressions. Full results of pairwise tests of significance are displayed in Tables 27 and 28 in Appendix B. Comparing the results in this setting to those in Table 23, we observe that the yields-only predictions are better on the short end, but worse on the long end. The predictions using both forward and macro information in Table 25 are better for all maturities and all methods. Thus, yield spread, momentum, equity returns and commodity returns do not capture all yield and macro information that is relevant for bond predictions.[33]

Metric	Method	n=24	n=36	n=48	n=60	n=84	n=120	EW
IR	Regression	0.43	0.42	0.40	0.39	0.42	0.48	0.43
	Random forest	0.21	0.20	0.22	0.23	0.26	0.32	0.26
	Neural network	0.02	0.02	0.03	0.03	0.04	0.07	0.04
SR	Benchmark	0.84	0.75	0.70	0.64	0.58	0.53	0.63
	Regression	0.82	0.73	0.68	0.62	0.58	0.57	0.63
	Random forest	0.77	0.69	0.65	0.59	0.55	0.52	0.59
	Neural network	0.81	0.72	0.68	0.62	0.56	0.53	0.62

Table 11: The Information Ratios and Sharpe Ratios for timing strategies with no duration exposure on average. Excess bond returns are considered for times to maturity n =24, 36, 48, 60, 84 and 120 and an equally weighted portfolio of those. Model forecasts are interpreted as signal and a z-score is computed. Every month, an investment equal to the z-score is done and positions are held for twelve months, such that the portfolio consists of 12 bonds at each point in time. The explanatory variables are yield spread s_t, 12 month bond momentum $d^{(n)}_{t-12:t}$, 12 month equity returns $q_{t-12:t}$ and 12 month commodity returns $c^{(n)}_{t-12:t}$.

[33] Untabulated results show that when adding these four variables to the first three principal components of the changes in forward rates, the forecasting power improves. However, in that case, out-of-sample R^2's are still lower than when including the full set of macro variables. We conclude that these four predictors neither capture all relevant macro nor all relevant yield information.

Information Ratios and Sharpe Ratios for trading strategies based on forecasts using yield spread, momentum, equity returns and commodity returns are displayed in Table 11. These results are in line with those reported in Table 5. The highest IR and SR are obtained based on regression forecasts. As for the R^2_{oos}'s, they are lower than found when using $\Delta f_t^{(n)}$ or $\Delta f_t^{(n)}$ and macro information. For the machine learning techniques, the Sharpe Ratios do not exceed the benchmark for any maturity. The regression SR is higher than the benchmark for $n = 120$.

Method	n=24	n=36	n=48	n=60	n=84	n=120	EW
Random forest	0.00	-0.01	-0.01	-0.01	-0.01	-0.02	-0.01
	(0.16)	(0.11)	(0.17)	(0.21)	(0.29)	(0.34)	(0.23)
Neural network	0.00	0.00	-0.01	-0.01	-0.01	-0.01	-0.01
	(0.20)	(0.14)	(0.22)	(0.28)	(0.31)	(0.46)	(0.29)

Table 12: Estimated alphas in a regression of machine learning investment returns on linear regression investment returns. P-values for a two-sided test of alpha= 0 are given between brackets. The explanatory variables are yield spread s_t, 12 month bond momentum $d^{(n)}_{t-12:t}$, 12 month equity returns $q_{t-12:t}$ and 12 month commodity returns $c^{(n)}_{t-12:t}$.

The alphas of regressing machine learning strategy returns on linear regression strategy returns are given in Table 12. None of them are significant. Table 30 illustrates that the results of the reversed regression. Again, linear regressions do yield positive and significant alpha beyond what is captured by machine learning.

6.2 Time-varying weights

It can also be argued that the data and model at the time of estimation provide information on the probability of misspecification itself. A high validation loss might be an indication of a poor model fit. Furthermore, if x_T is very dissimilar to $x_{T-1}, x_{T-2}, \ldots, x_1$, it is more likely that a model based on those observations is a poor fit for x_T. We thus also consider prior weights that vary depending on model fit and similarity of an observation to previous observations.

We measure fit in terms of validation loss, and similarity in terms of Euclidean distance to the mean of all previous observations. Every time the model is fit, we sort all measures of fit and similarity we have obtained out-of-sample so far. The weight is a linear function of the rank of the current fit or similarity. I.e., if the current model has the best fit so far, the prior weight is set to 0.25, and if it has the worst fit, the prior weight is set to 0.75. In all other cases, the weight is between 0.25 and 0.75 and linearly dependent on the rank of the fit or similarity.

We only consider time varying weights for machine learning techniques and not for linear regressions. The fit of a linear regression will vary only by a very small amount over time, especially as we use an expanding window. Also the rationale for using time-varying weights depending on similarity to previous observations does not hold up for linear regressions. Whereas machine learning techniques can assign more weight to certain observations, linear regressions are always fit on all observations. In fact, due to minimizing the squared error, observations far from the mean have above-average influence on the model.

6.2.1 Results

Table 13 shows the results of the Bayesian approach with varying weights between 0.25 and 0.75, depending on the fit of the model or similarity of the current observation to the previous observations on which the model is fit. Interestingly, only for neural networks in the second setting, time-varying weights dependent on similarity appear to work rather well. In all other cases there is no improvement compared to the case with a constant prior of 0.5. This is a somewhat surprising result. Intuitively, it would be logical to put more faith in a model with a good fit, or one which has been fit on similar observations to the one that is to be forecast. However, these in-sample measures are apparently not informative for the out-of-sample performance in this case.

Input	Method	Prior weight	n=24	n=36	n=48	n=60	n=84	n=120	EW
$\Delta f_t^{(n)}$	Random forest	fit	-0.1°	-0.2°	-0.7°	-1.2°	-3.4°	-5.5°	-3.1°
	Neural network	fit	0.5°	3.5**	4.6**	4.9**	4.9**	5.7**	5.5**
	Random forest	similarity	0.8*	0.9°	0.7°	0.6°	-1.3°	-2.7°	-1.0°
	Neural network	similarity	0.8°	2.8**	3.7**	3.9**	3.8**	4.3**	4.3**
$\Delta f_t^{(n)}$ and macro (real time)	Random forest	fit	-7.3°	-2.1°	0.8*	4.0*	6.3**	10.2**	5.3**
	Neural network	fit	-12.8°	-0.6°	3.2**	7.3**	10.5**	16.7***	9.0**
	Random forest	similarity	-2.6°	0.7°	2.9*	5.0*	6.3**	8.8**	5.8**
	Neural network	similarity	3.3*	9.6**	12.0**	14.2***	15.6***	18.8***	15.4***
s_t, $d_{t-12:t}^{(n)}$ $q_{t-12:t}$, $c_{t-12:t}^{(n)}$	Random forest	fit	-9.5°	-6.2°	-4.2°	-1.8°	0.9°	5.2°	-0.5°
	Neural network	fit	-5.6°	-4.5°	-3.7°	-2.5°	-1.3°	1.5°	-1.8°
	Random forest	similarity	-6.2°	-3.5°	-1.4°	0.5°	2.9°	6.2*	1.7°
	Neural network	similarity	-4.4°	-3.8°	-3.1°	-2.3°	-1.5°	0.7°	-1.8°

Table 13: The out-of-sample R^2 (in %) for the forecasting of excess bond returns for times to maturity n =24, 36, 48, 60, 84 and 120 and an equally weighted portfolio of those. Linear regressions, random forests and neural networks are considered. The explanatory variables are (1) the first differences of the forward rates $\Delta f_t^{(n)}$, (2) the first differences of the forward rates $\Delta f_t^{(n)}$ and (principal components of) a set of 128 macro variables or (3) the yield spread s_t, 12 month bond momentum $d_{t-12:t}^{(n)}$, 12 month equity returns $q_{t-12:t}$ and 12 month commodity returns $c_{t-12:t}^{(n)}$.

6.3 Mean-variance utility investor

We have shown in Sections 5.4-5.6 that linear regression forecasts add more economic value for bond investors than machine learning techniques, in terms of Information Ratios and Sharpe Ratios. However, for completeness we also set up a formal portfolio choice problem to investigate the economic value of our forecasts to a mean-variance utility investor. Our approach is comparable to that of Della Corte et al. (2008), Thornton and Valente (2012) and Bianchi et al. (2020b). We consider an investor that can invest in the risk free 1 year bond and an n period bond. At each time t, the investor maximizes quadratic utility

$$\max_{w} \mathrm{E}[R_{p,t:t+12}] - \frac{\gamma}{2}\mathrm{Var}[R_{p,t:t+12}], \tag{9}$$

with w the portfolio weight of the n month bond, γ the risk aversion parameter and $R_{p,t:t+12}$ the gross portfolio return between t and $t+12$. Substituting $R_{p,t:t+12} = 1 + y_t^{(12)} + w\ xr_{t:t+12}^{(n)}$ in Equation (9) gives

$$\max_{w} 1 + y_t^{(12)} + w\ \mathrm{E}[xr_{t:t+12}^{(n)}] - \frac{\gamma w^2}{2}\mathrm{Var}[xr_{t:t+12}^{(n)}]. \tag{10}$$

The first order condition is then

$$\begin{aligned} \mathrm{E}[xr_{t:t+12}^{(n)}] - \gamma w \mathrm{Var}[xr_{t:t+12}^{(n)}] &= 0 \\ \Rightarrow \qquad\qquad w &= \frac{\mathrm{E}[xr_{t:t+12}^{(n)}]}{\gamma \mathrm{Var}[xr_{t:t+12}^{(n)}]}. \end{aligned}$$

Here $\mathrm{E}[xr_{t:t+12}^{(n)}]$ is simply the model forecast and as estimator for $\mathrm{Var}[xr_{t:t+12}^{(n)}]$ we use its historical sample average.[34] To make our results directly comparable to those of e.g. Thornton and Valente (2012) and Bianchi et al. (2020a), we use $\gamma = 5$. Following Fleming et al. (2001), we compute the realized utilities at time t based on model M as

$$u_{t,M} = R_{p,t-12:t} - \frac{\gamma}{2(1+\gamma)} R_{p,t-12:t}^2.$$

Consequently, for each model we compute the Certainty Equivalent Return (CER) values that lead to the same average utilities. To test if the CER values of models M and N are significantly different, we consider the regression

$$u_{t,M} - u_{t,N} = \alpha + \varepsilon_t,$$

[34] Bianchi et al. (2020b) employ a rolling sample estimator instead. This estimator is a weighted average of squared forecast errors, rather than squared deviations from the mean. We refrain from this for two reasons. First, there are no forecast errors for the in-sample period. This would imply that we either need to compute in-sample forecast errors based on in-sample model fit, or we need to use squared deviations from the mean for the in-sample observations. The former is clearly problematic, and likely causes severe downward bias in the variance estimate, whereas the latter is inconsistent and implies not all observations are equally important for utility computations. Secondly, this leads to very extreme weights (sometimes exceeding 1000). If the most recent predictions were accurate, this would strongly reduce the variance estimate, especially as forecast errors are squared. This problem is further amplified by the large autocorrelation in forecast errors. It is hard to justify such an extremely strong relation between the accuracy of a forecast made at $t-12$ and the variance of $xr_{t:t+12}^{(n)}$.

where α is an intercept and ε_t is an error term. If α is positive and significant (using HAC standard errors), model M leads to a significantly higher utility than model N.

6.3.1 Results

The percentage differences in Certainty Equivalent Returns of the various methods[35] are shown in Table 14 for $n = 24$ and $n = 120$. For each maturity and for each setting, regression forecasts provide the highest Certainty Equivalent Return to a mean-variance investor. The welfare gains compared to the Expectations Hypothesis benchmark are always significant at a 1%, and also the welfare gains compared to the machine learning methods are generally significant. Neural networks also realise welfare gains compared to the benchmark in all settings; random forests only when including macro information. Thus, we find that linear regressions provide economic value to investors. These welfare gains are significantly larger than found with machine learning techniques.

		$n = 24$			$n = 120$		
		Benchmark	Regression	Random forest	Benchmark	Regression	Random forest
$\Delta f_t^{(n)}$	Regression	23.6***			15.1***		
	Random forest	-9.5	-26.8***		-1.7	-14.6***	
	Neural network	14.0***	-7.8**	26.0***	7.8***	-6.3**	9.7***
$\Delta f_t^{(n)}$ and macro (real time)	Regression	15.7***			47.2***		
	Random forest	9.7**	-5.2		20.8***	-17.9***	
	Neural network	12.3*	-3.0	2.4	29.0***	-12.3*	6.8*
$\Delta f_t^{(n)}$ and macro (final data)	Regression	17.4***			42.8***		
	Random forest	13.8***	-3.1		24.8***	-12.6**	
	Neural network	16.8***	-0.5	2.7	34.3***	-5.9	7.6*

Table 14: Percentage change in Certainty Equivalent Return for mean-variance utility investors when using forecasts from the row method instead of the column method. *, ** and *** indicate that the R^2 is significantly larger than 0 at 10, 5 or 1% significance level respectively, and ◦ indicates no significance.

6.4 Factor interpretations

The first three principal components of the forward rates or yields have a clear interpretation as level, slope and curvature (Nelson and Siegel, 1987). The interpretation is not so clear-cut when considering the changes in the forward rates. An additional complication is the fact that the model is re-estimated every month, such that the loadings of the principal components can change over time. The factor loadings for the in-sample and the full sample period are displayed in Table 15. In both settings, the first principal component is approximately a weighted average

[35] The CERs of the benchmark are 4.38% for the 2 year bond and 3.78% for the 10 year bond. For the 2 year bond in the yields-only setting, the CER for the regression is 5.04%, leading to the $\frac{5.04-4.38}{4.38} = 23.6\%$ increase given in Table 14.

of all maturities. We can interpret this as the level change component. Interestingly, the interpretation of principal components 2 and 3 has changed during the out-of-sample period. Plots of these factor loadings for the 2 year and 10 year forward rates during the out-of-sample period can be found in Figure 8.[36] The interpretation of these components switches approximately after 200 observations. In the beginning of the out-of-sample period, the second principal component has a positive loading for both $n = 24$ and $n = 120$ and a negative loading for most maturities in between. At the end of the period however, the loading is negative for $n = 120$ and positive for $n = 24$. The third principal component shows the reverse pattern. We call the component with the positive loading for short maturities and negative loading for long maturities (which is the third PC in the beginning of the out-of-sample period and the second PC at the end) the change in slope component.

		In-sample			Full sample	
n	PC 1	PC 2	PC 3	PC 1	PC 2	PC 3
12	0.38	-0.12	0.70	0.35	-0.69	0.10
24	0.36	0.02	0.28	0.36	-0.36	-0.04
36	0.32	0.07	0.12	0.34	-0.14	-0.09
48	0.31	0.06	-0.00	0.33	0.01	-0.10
60	0.26	-0.13	-0.00	0.29	0.09	0.08
72	0.29	0.48	-0.17	0.31	0.21	-0.43
84	0.37	-0.11	-0.20	0.35	0.24	0.17
96	0.18	-0.04	-0.05	0.21	0.22	0.02
108	0.25	0.63	-0.26	0.25	0.30	-0.54
120	0.38	-0.57	-0.52	0.34	0.36	0.67

Table 15: This Table reports the factor loadings of the first principal components of the changes in forward rates $\Delta f_t^{(n)} = f_t^{(m)} - f_{t-12}^{(n)}$ for the period 1971:08-1990:12 and 1971:08:2018:12.

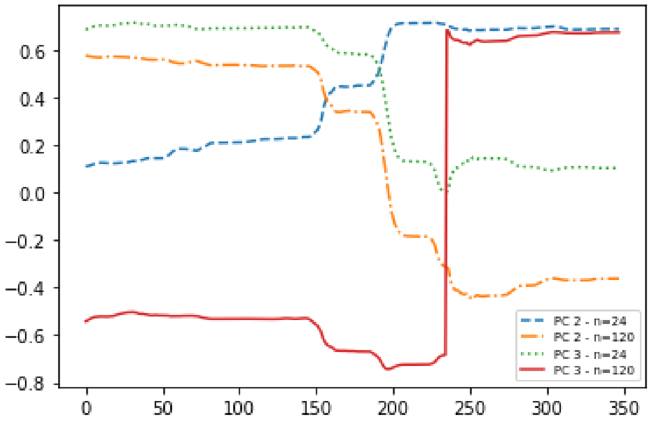

Figure 8: This Figure shows the loadings on the second and third principal components of the changes in forward rates $\Delta f_t^{(n)} = f_t^{(m)} - f_{t-12}^{(n)}$ over the period 1990:12-2018:12, for times to maturity $n = 24$ and $n = 120$.

[36] As the sign of the loadings is not identified, we have imposed the constraint that the loading on the first forward rate is always positive, such that the loadings are comparable over time.

The interpretation of the remaining principal component is not very clear. A typical curvature factor has positive loadings for short and long maturities and negative loadings in between. We do see this here, but the pattern is not smooth and mostly driven by the contrast between $n = 108$ and $n = 120$. To get more insight, we consider the full set of factor loadings and corresponding eigenvalues, as found in Table 31 in Appendix A. The loadings suggest that factors 3, 4 and 5 capture a combination of curvature and maturity-specific information. The average loadings of factors 3, 4 and 5 display a rather smooth curvature pattern (ascending in maturity): [0.24, -0.06, -0.16, -0.19, -0.14, -0.21, 0.00, 0.09, 0.10, 0.36]. As shown in Table 32 in Appendix A, the R^2_{oos} only marginally changes when replacing PC 3 with the average of PC 3, 4 and 5. We conclude that PC 3 relates to the curvature of the forward curve. Thus, PC 2 changes from curvature component to slope component throughout the out-of-sample period, and PC 3 experiences the reverse.

To assess forecasting power, we consider the regression of the excess returns on these components and a constant for the full sample period. The results are shown in Table 16. The intercept is large, positive and highly significant, and captures the fact that the excess returns are positive on average. The coefficient of the first PC is small and not significant. This indicates that changes in the level of the forward curve do not have significant predictive power for future excess return. The second principal component has a large and positive coefficient for all maturities. Thus, an increase in the slope of the yield curve is generally followed by large excess returns. This is conform expectations, as the literature suggests that a higher yield spread signals higher excess returns (as discussed in Section 6.1). The third principal component has a negative and significant coefficient for all maturities, but of smaller size. To confirm its interpretation as curvature component, we repeat the regression, replacing PC 3 with the average of PC 3, 4 and 5. Remarkably, the coefficients hardly change and the coefficients of this combination are also negative and significant. This supports the conclusion that the information in PC 3 relevant for bond predictions relates to changes in the curvature of the forward curve.

Thus, the regression coefficient suggests that if long and short forward rates increase by a lot relative to medium maturities (the forward curve becomes less concave), this signals lower excess returns. This is consistent with the work of Afonso and Martins (2012), who find that a decrease in concavity is followed by a yield level increase ceteris paribus. Rising yields are disadvantageous for current bond holders and therefore have a depressing effect on excess returns.

	n=24	n=36	n=48	n=60	n=84	n=120	EW
Const.	0.62 (8.72)	1.08 (8.38)	1.52 (8.37)	1.74 (7.69)	2.22 (7.10)	2.62 (6.05)	1.63 (7.40)
PC 1	0.02 (0.78)	0.03 (0.55)	0.03 (0.37)	0.01 (0.11)	-0.05 (-0.45)	-0.13 (-0.75)	-0.02 (-0.19)
PC 2	0.38 (7.84)	0.70 (8.22)	0.97 (8.16)	1.17 (8.02)	1.57 (8.13)	2.05 (7.89)	1.14 (8.22)
PC 3	-0.13 (-2.30)	-0.30 (-2.94)	-0.45 (-3.00)	-0.49 (-2.63)	-0.93 (-3.26)	-1.31 (-3.23)	-0.60 (-3.10)

Table 16: This Table reports the coefficients of a regression of excess returns on the principal components of $\Delta f_t^{(n)} = f_t^{(m)} - f_{t-12}^{(n)}$ and an intercept for the period 1971:08:2018:12. T statistics are between brackets, all significant at a 5% level except for PC 1.

6.4.1 Macro factors

The coefficients of predictive regressions including (real time) macro factors are shown in Table 17. Only macro factors 2, 3 and 6 hold significant preditive power for excess returns. PC 2 is included for all maturities except $n = 24$ and has a positive sign. The factor loadings in Table 22 suggest that this is largely a price level component, loading negatively on inflation variables. The positive coefficients imply that low inflation signals high excess returns. As discussed in Section 6.1, this is expected. Low inflation boosts the value of nominal bonds, and thus excess returns. The third variable loads negatively on yield variables, such as yield spread. This too is conform expectations, as the regression coefficients are negative. We have seen that yield spread is positively related to future excess returns. It is interesting to note that the yield variables in the macro data set add information that is not yet contained in the principal components of the changes in the forward rates. This implies that both the change in the slope and the slope itself hold predictive power for future excess returns. The sixth principal component is included only for $n = 120$. Its interpretation is not very clear. It loads negatively on employment, orders and inventories and can be interpreted as a business cycle factor. Its importance can be explained by a rationale similar to that for equity returns in Section 6.1. When investor risk aversion is declining in wealth, investors are more risk averse during economic downturns. They invest a larger share of their assets in safe bonds, driving up their prices and excess returns.

	n=24	n=36	n=48	n=60	n=84	n=120	EW
Constant	0.62 (8.76)	1.09 (8.64)	1.54 (8.77)	1.76 (8.17)	2.26 (7.66)	2.70 (6.70)	1.66 (7.90)
Forward - PC 1	0.01 (0.42)	-0.02 (-0.38)	-0.05 (-0.74)	-0.10 (-1.13)	-0.22 (-1.81)	-0.34 (-2.02)	-0.12 (-1.45)
Forward - PC 2	0.37 (7.56)	0.65 (7.66)	0.88 (7.55)	1.05 (7.36)	1.39 (7.45)	1.86 (7.39)	1.02 (7.56)
Forward - PC 3	-0.13 (-2.30)	-0.32 (-3.07)	-0.48 (-3.17)	-0.53 (-2.83)	-0.99 (-3.46)	-1.50 (-3.73)	-0.64 (-3.27)
Macro - PC 2		0.14 (3.83)	0.23 (4.65)	0.33 (5.21)	0.50 (5.92)	0.72 (6.11)	0.33 (5.56)
Macro - PC 3	-0.07 (-2.78)	-0.17 (-4.03)	-0.30 (-5.11)	-0.42 (-5.85)	-0.64 (-6.82)	-0.91 (-7.10)	-0.42 (-6.20)
Macro - PC 6						0.60 (2.74)	

Table 17: The coefficients of a regression of excess returns on the principal components of $\Delta f_t^{(n)} = f_t^{(m)} - f_{t-12}^{(n)}$, the principal components of a large set of macro variables and an intercept for the period 1971:08:2018:12. T statistics are between brackets, all significant at a 5% level except for PC 1.

As discussed in Section 5.1, a clear drop in forecasting power was found for neural networks when using real time instead of final macro data. As is shown in Tables 24 and 25 in Appendix A, the impact is much smaller for the principal-component regressions. This finding is not as much as it may seem in contrast with the work of Ghysels et al. (2018), who report large reductions in predictive power when using real time instead of final data. They focus on variables that add predictive power to the CP factor as well as on the first factor of Ludvigson and Ng (2009), which was dubbed the 'real factor' and captured employment and production. In our setting however, we use the first three principal components of the changes in forward rates to capture yield information, rather than the CP factor. The principal components of the macro variables that were added to this were often PC 2 and PC 3, rather than PC 1 (which in our setting also captures employment and production). As mentioned above, PC 2 loads on interest rate variables and PC 3 on price variables. Interest and price information is generally known in real time, which explains why in our setting, the difference between the results with vintage vis-a-vis final data is small for regressions.

6.5 Robustness checks

To ascertain our results are robust and not specific for certain machine learning techniques or the settings used in those techniques, we repeat our research using several other techniques and settings. In these checks, the same data, period, validation sample, etc. are used to ensure comparability of the results. As alternatives to random forests we consider extreme trees and gradient-boosted trees. For neural networks, we investigate different numbers of nodes and layers. In the forwards-only setting, we consider a setting with 1 hidden layer of 5 nodes, a setting of 2 hidden layers of 3 nodes and a setting of 3 hidden layers of 3 nodes. In the forward + macro setting we consider a setting of 1 hidden layer of 64 nodes, a setting of 2 hidden layers of 32 nodes and a setting of 3 hidden layers of 32 nodes. Increasing the number of nodes and layers makes the network more flexible and allows for a more nonlinear specification.

6.5.1 Results

The out-of-sample R^2's, Information Ratios and Sharpe Ratios of extreme trees, gradient-boosted trees and various configurations of neural networks are displayed in Figures 37-40 in Appendix B. The results are in line with our main results. In terms of R^2's, extreme trees and gradient-boosted trees appear to perform a bit better than random forests, with the clear exception of gradient-boosted trees in the setting with yield spread, momentum, equity returns and commodity returns. Yet, they are still outperformed by neural networks and regressions. The performance of the neural networks is remarkably similar to that of the base case. The strongest predictive power is again found for the longer maturities in the setting with forward and macro variables. However, in all settings and for all maturities, all machine learning techniques are still outperformed by the regression. Again, adding a prior weight of 0.25 or 0.5 generally improves all methods.

The Information Ratios are low overall, and again the predictive power of neural networks in terms of the R^2_{oos} does not translate in a high IR. None of the techniques achieves an IR higher than the regression in any of the settings and for any of the maturities considered. The Sharpe Ratios are also comparable to those reported in Section 5.5, with the techniques improving the benchmark by a small amount in some cases. None of the Sharpe Ratios are higher than those of linear regressions.

7 Discussion & conclusion

In this paper, we investigate the use of machine learning for forecasting bond risk premia. We predict excess returns on US government bonds between 1990:01 and 2018:12, using data from 1971:08 onwards to fit the models. We compare linear regressions, tree-based methods and neural networks with two different sets of explanatory variables. In the first setting, we only use yield information in our forecasts. In the second setting, we combine yield variables with a large set of macro data.

Earlier work by Bianchi et al. (2020b) shows that neural networks achieve positive predictive power, in contrast to principal-component regressions and tree-based methods. However, in this paper we illustrate that there are several issues to the work of Bianchi et al. (2020b), which each have substantial impact on forecasting performance. Firstly, part of the results rely on the use of final macro data, which was not available at the time of estimation. We perform our research using vintage data instead. Secondly, the forward rates that are used as explanatory variables are nonstationary. We solve this problem by taking their first differences. Finally, model and parameter uncertainty are ignored. We take a Bayesian view and shrink model forecasts to the historical mean to take this uncertainty into account.

Our research has yielded several remarkable findings. First and foremost, when using stationary instead of nonstationary explanatory variables, linear regressions are not the worst performing, but the best performing method. This holds across all settings and all maturities and is in stark contrast to the findings of Bianchi et al. (2020a). We thus offer an alternative interpretation of their findings, namely that nonstationary explanatory variables can be less problematic for neural networks than for regressions. However, with stationary input data, a linear regression outperforms all machine learning techniques, irrespective of the explanatory variables that they use.

Secondly, we find that imposing a prior on forecasts generally leads to improvements. This can change the ranking of the methods, e.g. even with nonstationary input data, a linear regression outperforms a neural network on the long end of the curve when a prior is imposed. Thirdly, forecasting power of macro data is lower in real time than when using final data, mostly so for neural networks. Fourthly, adding macro variables to yields-only models significantly improves the predictability of returns on long maturity bonds. This is evidence against the Spanning Hypothesis, which postulates that all information relevant for future bond returns is contained in the yield curve.

Our most important conclusion, that machine learning techniques are outperformed by linear regressions, is not unique to random forests or specific configurations of neural networks, but also holds for extreme trees, gradient-boosted trees and all other configurations of neural networks considered. This finding also holds when using a different set of predictors, consisting of yield spread, momentum, bond returns and equity returns. Moreover, we design simple trading strategies that interpret model forecasts as signals. We find that trading strategies based on linear regression forecasts achieve higher Information Ratios and Sharpe Ratios than strategies based on machine learning forecasts. We also investigate whether machine learning techniques can detect alpha beyond what is found by linear techniques via the use of encompassing regressions. We document that machine learning does not capture significantly positive alpha beyond what is found by regressions, but reversely, linear regressions do capture positive and significant alpha beyond what is found by machine learning. Finally, linear regressions provide significant welfare gains for mean-variance utility investors, compared to machine learning techniques and the Expectations Hypothesis benchmark.

Having established that machine learning techniques fail to outperform linear techniques in forecasting bond risk premia, the question remains why this is the case. The explanation that we offer for this finding is that the flexibility of machine learning is not only an advantage, but also a disadvantage. On the one hand, it is able to pick up many more patterns than simple linear techniques. On the other hand, it can also pick up much more noise than linear methods, resulting in poorer model fit. This can imply failure to select the right explanatory variables, or failure to accurately capture the relation between the dependent variable and the independent variables (or both). This disadvantage is especially pressing when, such as in our case, data is noisy and the dependent variable is generally very hard to predict.

It might be insightful to draw a parallel with the classical problem of selecting the right amount of explanatory variables in a linear regression. Just as a highly flexible neural network can provide a better model fit than a simple regression, an increase in the number of regressors can also improve model fit. However, this also increases the variance of parameter estimates, which can ultimately lead to selection of a poorer model. A related problem is overfitting. Additional regressors always improve in-sample fit, but this may very well come at the expense of out-of-sample forecasting performance.

The solution that we propose for this pressing issue is the following. Rather than dealing with model uncertainty and overfitting on an ex post basis (by e.g. model averaging or shrinkage), we can explicitly model them in the optimization problem. In our current approach, we select the model with the best in-sample model fit. This fit is sometimes so good, that we know almost certainly that it does not extend to out-of-sample predictions. Instead of selecting a model with a very good in-sample fit and hoping that this good fit at least in part carries over to the out-of-sample analysis, researchers could let machine learning techniques select configurations based on both model fit and robustness, rather than model fit alone. Alternatively, we can reduce the number of parameters in the model, or fix some parameters to a predetermined value. It is also possible to initiate machine learning optimization at a point "close to" the linear model, or to switch between machine learning and linear models, whichever has the best fit. Finally, the low signal-to-noise ratio could also be explicitly modelled.

Also outside of the bond market, there are still ample issues to look into. Machine learning techniques can be applied to situations where more data is available, such as the corporate bond market. Machine learning can deal with large data sets more naturally than linear techniques and more information can also prevent overfitting. Moreover, machine learning techniques can be used in situations where linear methods have very limited forecasting power (such as the stock market), implying that the benchmark is lower.

Bibliography

Afonso, A. and Martins, M. M. (2012). Level, slope, curvature of the sovereign yield curve, and fiscal behaviour. *Journal of Banking & Finance*, 36(6):1789–1807.

Ang, A. and Piazzesi, M. (2003). A no-arbitrage vector autoregression of term structure dynamics with macroeconomic and latent variables. *Journal of Monetary Economics*, 50(4):745–787.

Bacon, C. R. (2008). *Practical portfolio performance measurement and attribution*, volume 546. John Wiley & Sons.

Baltussen, G., Martens, M., and Penninga, O. (2020). Predicting international bond returns: 70 years of evidence. Working paper. *Available at SSRN 3631109.*

Bansal, R. and Shaliastovich, I. (2013). A long-run risks explanation of predictability puzzles in bond and currency markets. *The Review of Financial Studies*, 26(1):1–33.

Bauer, M. D. and Hamilton, J. D. (2018). Robust bond risk premia. *The Review of Financial Studies*, 31(2):399–448.

Bianchi, D., Büchner, M., Hoogteijling, T., and Tamoni, A. (2020a). Corrigendum: bond risk premiums with machine learning. *The Review of Financial Studies*, Forthcoming.

Bianchi, D., Büchner, M., and Tamoni, A. (2020b). Bond risk premiums with machine learning. *The Review of Financial Studies*, Forthcoming.

Campbell, J. Y. and Shiller, R. J. (1991). Yield spreads and interest rate movements: a bird's eye view. *The Review of Economic Studies*, 58(3):495–514.

Campbell, J. Y. and Thompson, S. B. (2008). Predicting excess stock returns out of sample: can anything beat the historical average? *The Review of Financial Studies*, 21(4):1509–1531.

Ceballos, F. (2019). *An intuitive explanation of random forest and extra trees classifiers.* Towards Data Science.

Chollet, F. (2017). *Introduction to Keras for researchers.* Keras.

Cieslak, A. and Povala, P. (2015). Expected returns in Treasury bonds. *The Review of Financial Studies*, 28(10):2859–2901.

Clark, T. E. and West, K. D. (2007). Approximately normal tests for equal predictive accuracy in nested models. *Journal of Econometrics*, 138(1):291–311.

Cochrane, J. H. and Piazzesi, M. (2005). Bond risk premia. *American Economic Review*, 95(1):138–160.

Cooper, I. and Priestley, R. (2009). Time-varying risk premiums and the output gap. *The Review of Financial Studies*, 22(7):2801–2833.

Cutler, D. M., Poterba, J. M., and Summers, L. H. (1991). Speculative dynamics. *The Review of Economic Studies*, 58(3):529–546.

Daffodil Software (2017). *9 applications of machine learning from day-to-day life*. Medium.

Della Corte, P., Sarno, L., and Thornton, D. L. (2008). The expectation hypothesis of the term structure of very short-term rates: statistical tests and economic value. *Journal of Financial Economics*, 89(1):158–174.

Dickey, D. A. and Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366a):427–431.

Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 20(1):134–144.

Dyl, E. A. and Joehnk, M. D. (1981). Riding the yield curve: does it work? *The Journal of Portfolio Management*, 7(3):13–17.

Fama, E. F. and Bliss, R. R. (1987). The information in long-maturity forward rates. *The American Economic Review*, 77(4):680–692.

Feng, G. G., Fulop, A., and Li, J. (2020). Real-time macro information and bond return predictability: does deep learning help? *Available at SSRN 3517081.*

Fisher, I. (1896). *Appreciation and interest: a study of the influence of monetary appreciation and depreciation on the rate of interest with applications to the bimetallic controversy and the theory of interest*, volume 11. American Economic Association.

Fleming, J., Kirby, C., and Ostdiek, B. (2001). The economic value of volatility timing. *The Journal of Finance*, 56(1):329–352.

Foote, K. (2019). *A brief history of machine learning*. Dataversity.

Garbade, K. D. (1996). *Fixed income analytics*. MIT Press.

Ghysels, E., Horan, C., and Moench, E. (2018). Forecasting through the rearview mirror: data revisions and bond return predictability. *The Review of Financial Studies*, 31(2):678–714.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Greenwood, R. and Vayanos, D. (2014). Bond supply and excess bond returns. *The Review of Financial Studies*, 27(3):663–713.

Grinold, R. C. and Kahn, R. N. (1992). Information analysis. *Journal of Portfolio Management*, 18(3):14–21.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 3(5):2223—-2273.

Géron, A. (2017). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.)*. O'Reilly Media, Inc.

Hamilton, J. D. and Wu, J. C. (2012). Identification and estimation of Gaussian affine term structure models. *Journal of Econometrics*, 168(2):315–331.

Harvey, D., Leybourne, S., and Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, 13(2):281–291.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Heaton, J., Polson, N., and Witte, J. H. (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1):3–12.

Heij, C., de Boer, P., Franses, P. H., Kloek, T., and van Dijk, H. K. (2004). *Econometric methods with applications in business and economics*. Oxford University Press.

Ilmanen, A. (1995). Time-varying expected returns in international bond markets. *The Journal of Finance*, 50(2):481–506.

Ilmanen, A. (1997). Forecasting US bond returns. *The Journal of Fixed Income*, 7(1):22.

Jacobs, B. I. and Levy, K. N. (1996). Residual risk: how much is too much? *Journal of Portfolio Management*, 22(3):10.

Joslin, S., Priebsch, M., and Singleton, K. J. (2014). Risk premiums in dynamic term structure models with unspanned macro risks. *The Journal of Finance*, 69(3):1197–1233.

Jung, K. and Shah, N. H. (2015). Implications of non-stationarity on predictive modeling using EHRs. *Journal of Biomedical Informatics*, 58:168–174.

Litterman, R. and Scheinkman, J. (1991). Common factors affecting bond returns. *Journal of Fixed Income*, 1(1):54–61.

Liu, Y. and Wu, J. C. (2019). Reconstructing the yield curve. *Available at SSRN 3286785.*

Ludvigson, S. C. and Ng, S. (2009). Macro factors in bond risk premia. *The Review of Financial Studies*, 22(12):5027–5067.

Martens, M., Beekhuizen, P., Duyvesteyn, J., and Zomerdijk, C. (2019). Carry investing on the yield curve. *Financial Analysts Journal*, 75(4):51–63.

McCracken, M. W. and Ng, S. (2016). FRED-MD: a monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4):574–589.

Mitchell, T. M. (2006). *The discipline of machine learning*, volume 9. Carnegie Mellon University, School of Computer Science, Machine Learning.

Nelson, C. R. and Siegel, A. F. (1987). Parsimonious modeling of yield curves. *Journal of Business*, 60(4):473–489.

Nesterov, Y. E. (1983). A method for solving the convex programming problem with convergence rate O($\frac{1}{k^2}$). In *Soviet Math Dokl*, volume 269, pages 543–547.

Onatski, A. and Wang, C. (2020). Spurious factor analysis. *https://doi.org/10.17863/CAM.52423*.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Shiller, R. J. and McCulloch, J. H. (1990). The term structure of interest rates. *Handbook of monetary economics*, 1:627–722.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

Steinberg, R. (2017). *6 areas where artificial neural networks outperform humans.* Venturebeat.

Stephanie, G. (2019). *Decision tree vs random forest vs gradient boosting machines: explained simply.* Data Science Central.

Sugiyama, M., Yamada, M., and du Plessis, M. C. (2013). Learning under nonstationarity: covariate shift and class-balance change. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(6):465–477.

Thornton, D. L. and Valente, G. (2012). Out-of-sample predictions of bond excess returns and forward rates: an asset allocation perspective. *The Review of Financial Studies*, 25(10):3141–3168.

Trip, T. K. (2019). *Random forest vs extra trees*. The Kernel Trip.

Uhlig, H. (2009). Comment on "How has the euro changed the monetary transmission mechanism?". In *NBER Macroeconomics Annual 2008*, pages 141–152. University of Chicago Press.

Appendix

A Tables

Time to maturity	24	36	48	60	84	120	EW
ρ_1	0.928	0.928	0.930	0.926	0.924	0.922	0.928
ρ_{12}	0.231	0.183	0.147	0.104	0.042	-0.005	0.078

Table 18: The first and twelfth order autocorrelation in excess bond returns over the period 1971:08-2018:12. We consider bonds with a time to maturity of 24, 36, 48, 60, 84 and 120 months and an equally weighted portfolio of those.

	No trend			Trend		
n	DF	p-value	lags	DF	p-value	lags
12	-1.77	0.395	11	-3.81	0.016	17
24	-1.35	0.605	11	-3.29	0.068	11
36	-1.23	0.659	0	-2.97	0.141	0
48	-1.02	0.746	5	-2.80	0.196	5
60	-0.97	0.765	5	-2.93	0.154	5
72	-1.00	0.755	14	-2.96	0.144	6
84	-0.92	0.781	15	-2.85	0.179	15
96	-1.15	0.695	1	-2.79	0.200	1
108	-1.36	0.601	3	-3.01	0.131	3
120	-0.95	0.772	15	-2.71	0.232	15

Table 19: The results of an Augmented Dickey-Fuller Test for a unit root in the n month forward rates in the period 1971:08-2018:12. The ADF test statistic is computed as $\frac{\hat{\rho}}{\text{se}(\hat{\rho})}$, where $\hat{\rho}$ is the estimate of ρ and se($\hat{\rho}$) its standard error. The null hypothesis is the existence of a unit root.

n	DF	p-value	lags
12	-4.53	0.000	19
24	-4.23	0.001	16
36	-5.13	0.000	12
48	-5.32	0.000	12
60	-4.91	0.000	15
72	-5.19	0.000	15
84	-6.20	0.000	12
96	-3.88	0.002	13
108	-4.73	0.000	17
120	-5.80	0.000	16

Table 20: The results of an Augmented Dickey-Fuller Test for a unit root in the first differences of the forward rates $\Delta f_t^{(n)}$ in the period 1971:08-2018:12. The ADF test statistic is computed as $\frac{\hat{\rho}}{\text{se}(\hat{\rho})}$, where $\hat{\rho}$ is the estimate of ρ and se($\hat{\rho}$) its standard error. The null hypothesis is the existence of a unit root. The tests are performed with intercept but without trend.

Variable	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Production								
Real Personal Income	0.082	0.067	0.011	0.097	-0.084	0.010	0.024	0.057
Real personal income ex transfer receipts	0.112	0.058	0.022	0.091	-0.073	-0.015	-0.018	0.023
Real personal consumption expenditures	0.060	0.083	-0.065	0.102	-0.051	0.071	0.147	-0.169
Real Manu. and Trade Industries Sales	0.120	0.043	-0.057	0.089	-0.103	0.042	0.106	-0.090
Retail and Food Services Sales	0.052	0.027	-0.071	0.040	-0.100	0.055	0.191	-0.207
IP Index	0.170	-0.017	-0.052	0.084	-0.115	-0.059	-0.016	0.037
IP: Final Products and Nonindustrial Supplies	0.159	-0.005	-0.068	0.097	-0.138	-0.065	0.014	0.065
IP: Final Products (Market Group)	0.148	-0.005	-0.067	0.104	-0.150	-0.069	0.008	0.078
IP: Consumer Goods	0.124	0.017	-0.126	0.111	-0.143	-0.078	0.003	0.084
IP: Durable Consumer Goods	0.128	0.017	-0.107	0.128	-0.081	-0.033	-0.027	0.082
IP: Nondurable Consumer Goods	0.079	0.007	-0.105	0.052	-0.156	-0.097	0.033	0.055
IP: Business Equipment	0.151	-0.025	0.039	0.060	-0.107	-0.042	0.027	0.055
IP: Materials	0.159	-0.024	-0.033	0.063	-0.083	-0.047	-0.040	0.010
IP: Durable Materials	0.164	-0.017	-0.010	0.076	-0.059	-0.041	-0.069	0.062
IP: Nondurable Materials	0.130	-0.015	-0.075	0.041	-0.074	-0.074	-0.028	0.010
IP: Manufacturing (SIC)	0.171	-0.011	-0.050	0.082	-0.110	-0.068	-0.028	0.062
IP: Residential Utilities	0.006	-0.005	-0.065	0.059	0.001	0.032	0.054	-0.021
IP: Fuels	0.021	-0.011	-0.011	-0.051	-0.062	0.017	-0.011	0.075
ISM Manufacturing: Production Index	0.169	-0.001	-0.058	0.086	-0.107	-0.071	-0.030	0.066
Capacity Utilization: Manufacturing	0.117	0.037	-0.064	0.009	0.053	0.092	-0.056	-0.001
Labor market								
Help-Wanted Index for United States	0.135	0.046	-0.076	-0.004	0.034	0.112	-0.122	0.005
Ratio of Help Wanted/No. Unemployed	0.031	-0.007	0.053	0.015	-0.004	-0.073	0.122	-0.052
Civilian Labor Force	0.129	-0.006	0.021	0.001	-0.012	-0.024	-0.036	-0.042
Civilian Employment	-0.133	0.002	0.036	0.015	0.005	-0.060	0.199	0.003
Civilian Unemployment Rate	-0.035	0.002	-0.155	0.060	-0.070	0.065	0.046	0.124
Average Duration of Unemployment (Weeks)	-0.044	-0.012	0.105	-0.037	0.007	-0.094	0.083	-0.086
Civilians Unemployed - Less Than 5 Weeks	-0.077	0.000	0.049	0.040	0.029	-0.023	0.048	0.013
Civilians Unemployed for 5–14 Weeks	-0.116	-0.021	-0.108	0.062	-0.014	0.013	0.209	0.118
Civilians Unemployed - 15 Weeks and Over	-0.080	-0.019	-0.017	0.073	0.037	0.001	0.172	0.113
Civilians Unemployed for 15–26 Weeks	-0.088	-0.016	-0.150	0.002	-0.057	0.010	0.135	0.060
Civilians Unemployed for 27 Weeks and Over	-0.093	-0.034	0.093	-0.083	0.000	-0.082	0.027	0.008
Initial Claims	0.169	-0.009	0.071	-0.023	-0.044	-0.053	-0.035	-0.076
All Employees: Total nonfarm	0.173	-0.023	0.054	0.001	-0.044	-0.046	-0.063	-0.040
All Employees: Goods-Producing Industries	0.012	-0.097	0.065	0.040	-0.068	-0.001	-0.016	-0.099
All Employees: Mining and Logging: Mining	0.129	0.036	0.054	0.006	-0.053	-0.063	0.013	0.030
All Employees: Construction	0.171	-0.036	0.040	0.000	-0.016	-0.033	-0.087	-0.045
All Employees: Manufacturing	0.168	-0.044	0.051	0.002	0.001	-0.006	-0.090	-0.031
All Employees: Durable goods	0.140	-0.005	0.000	-0.005	-0.059	-0.100	-0.059	-0.072
All Employees: Nondurable goods	0.125	0.003	0.075	-0.042	-0.033	-0.050	0.004	-0.096
All Employees: Service-Providing Industries	0.158	-0.009	0.039	-0.030	-0.022	-0.038	-0.081	-0.109
All Employees: Trade, Transportation & Utilities	0.133	-0.062	0.091	-0.058	-0.004	-0.035	-0.051	-0.097
All Employees: Wholesale Trade	0.134	0.029	0.002	0.001	-0.009	-0.039	-0.110	-0.073
All Employees: Retail Trade	0.106	-0.003	0.090	-0.077	0.043	0.043	0.065	-0.081
All Employees: Financial Activities	0.004	0.005	0.100	-0.055	0.008	-0.074	0.126	-0.005
All Employees: Government	0.114	-0.014	0.150	-0.048	-0.047	-0.029	0.072	0.076
Avg Weekly Hours: Goods-Producing	0.071	-0.016	-0.076	0.058	-0.021	-0.006	0.021	0.077
Avg Weekly Overtime Hours: Manufacturing	0.119	-0.018	0.160	-0.062	-0.019	-0.009	0.066	0.082
Avg Weekly Hours: Manufacturing	0.153	0.021	0.135	-0.095	0.059	0.093	0.101	0.007
ISM Manufacturing: Employment Index	0.102	0.055	0.130	-0.103	0.039	0.048	0.134	0.092
Housing								
Housing Starts: Total New Privately Owned	0.110	-0.010	0.132	-0.079	0.056	0.046	0.081	0.038
Housing Starts, Northeast	0.132	0.009	0.099	-0.061	0.025	0.112	0.071	-0.045
Housing Starts, Midwest	0.150	0.029	0.119	-0.099	0.079	0.074	0.077	0.012
Housing Starts, South	0.154	0.032	0.108	-0.097	0.068	0.106	0.092	0.009
Housing Starts, West	0.115	0.050	0.121	-0.111	0.068	0.055	0.109	0.087
New Private Housing Permits (SAAR)	0.129	0.015	0.116	-0.098	0.058	0.054	0.103	0.051
New Private Housing Permits, Northeast (SAAR)	0.117	0.013	0.073	-0.060	0.025	0.126	0.057	-0.040
New Private Housing Permits, Midwest (SAAR)	0.151	0.037	0.101	-0.093	0.093	0.089	0.076	0.012
New Private Housing Permits, South (SAAR)	0.063	-0.236	-0.100	0.034	-0.034	0.170	0.017	0.080
New Private Housing Permits, West (SAAR)	0.089	-0.012	-0.075	0.026	-0.056	0.047	0.057	-0.123
Orders & inventories								
ISM: PMI Composite Index	0.036	-0.021	-0.015	-0.010	-0.051	0.013	0.072	-0.112
ISM: New Orders Index	0.084	-0.130	0.139	-0.056	-0.031	0.090	0.010	-0.067
ISM: Supplier Deliveries Index	0.000	-0.154	0.186	-0.029	-0.018	-0.017	-0.024	-0.025
ISM: Inventories Index	-0.102	-0.037	0.101	-0.042	0.135	-0.050	-0.154	0.060
New Orders for Consumer Goods	-0.023	0.097	0.023	-0.050	-0.065	0.085	-0.154	-0.164
New Orders for Durable Goods	-0.023	0.093	0.023	-0.053	-0.108	0.036	-0.200	-0.059

New Orders for Nondefense Capital Goods	0.038	0.189	-0.079	0.042	0.078	-0.006	-0.039	-0.045
Unfilled Orders for Durable Goods	-0.001	0.036	0.002	0.042	-0.009	0.081	-0.151	0.052
Total Business Inventories	-0.001	0.004	0.010	0.017	-0.016	0.090	-0.096	-0.060
Total Business: Inventories to Sales Ratio	-0.004	0.016	0.042	0.022	-0.060	0.064	-0.093	-0.057
Money & credit								
M1 Money Stock	0.033	-0.018	-0.062	0.013	0.056	0.018	0.062	-0.017
M2 Money Stock	0.029	0.029	-0.021	0.003	-0.028	-0.002	-0.049	0.053
Real M2 Money Stock	0.013	-0.012	-0.051	-0.048	0.073	-0.023	0.021	-0.237
St. Louis Adjusted Monetary Base	0.062	0.017	0.058	-0.110	0.157	-0.043	0.037	-0.177
Total Reserves of Depository Institutions	-0.003	0.156	-0.095	-0.016	-0.085	0.291	0.083	-0.056
Reserves Of Depository Institutions	0.000	0.149	-0.098	-0.017	-0.078	0.298	0.087	-0.057
Commercial and Industrial Loans	0.005	-0.164	0.094	0.020	0.090	-0.278	-0.097	0.051
Real Estate Loans at All Commercial Banks	-0.007	0.172	-0.118	0.010	-0.080	0.268	0.072	-0.047
Total Nonrevolving Credit	0.082	-0.150	0.000	0.075	0.036	0.125	0.074	0.007
Nonrevolving consumer credit to Personal Income	0.080	-0.177	-0.053	0.073	0.150	0.092	0.075	0.060
S&P's Common Stock Price Index: Composite	0.070	-0.174	-0.096	0.042	0.160	0.123	0.073	0.040
S&P's Common Stock Price Index: Industrials	0.071	-0.194	-0.096	0.060	0.170	0.116	0.043	0.046
S&P's Composite Common Stock: Dividend Yield	0.071	-0.193	-0.099	0.065	0.190	0.109	0.016	0.034
S&P's Composite Common Stock: Price-Earnings Ratio	0.060	-0.191	-0.101	0.050	0.186	0.084	-0.031	0.005
Effective Federal Funds Rate	0.050	-0.182	-0.093	0.045	0.187	0.070	-0.046	0.001
3-Month AA Financial Commercial Paper Rate	0.043	-0.198	-0.079	0.058	0.170	0.040	-0.056	0.005
3-Month Treasury Bill	0.022	-0.216	-0.027	0.071	0.136	0.014	-0.042	0.005
6-Month Treasury Bill	0.067	0.085	-0.080	0.010	0.227	-0.145	0.026	0.080
1-Year Treasury Rate	0.100	0.144	-0.104	-0.029	0.213	-0.054	0.007	0.024
5-Year Treasury Rate	0.096	0.156	-0.102	-0.026	0.217	-0.083	0.003	0.026
10-Year Treasury Rate	0.081	0.155	-0.128	-0.002	0.215	-0.106	-0.033	0.002
Moody's Seasoned Aaa Corporate Bond Yield	0.073	0.199	-0.121	-0.019	0.177	-0.118	-0.026	0.014
Moody's Seasoned Baa Corporate Bond Yield	0.075	0.204	-0.114	-0.025	0.173	-0.108	-0.017	0.022
3-Month Commercial Paper Minus FEDFUNDS	0.068	0.213	-0.112	-0.029	0.160	-0.109	-0.002	0.042
3-Month Treasury C Minus FEDFUNDS	0.046	0.223	-0.138	-0.012	0.142	-0.109	-0.013	0.033
6-Month Treasury C Minus FEDFUNDS	-0.005	-0.109	-0.117	0.099	0.004	-0.101	-0.014	-0.298
1-Year Treasury C Minus FEDFUNDS	0.009	-0.097	-0.125	0.069	-0.004	-0.055	-0.022	-0.272
5-Year Treasury C Minus FEDFUNDS	-0.021	-0.105	-0.079	0.071	0.022	-0.072	-0.019	-0.266
10-Year Treasury C Minus FEDFUNDS	0.002	0.060	0.103	-0.094	0.011	0.086	0.051	0.253
Moody's Aaa Corporate Bond Minus FEDFUNDS	0.017	-0.072	-0.046	0.076	0.040	-0.053	0.086	-0.002
Moody's Baa Corporate Bond Minus FEDFUNDS	0.000	-0.043	-0.072	-0.201	-0.019	-0.088	0.130	-0.111
Trade Weighted U.S. Dollar Index: Major Currencies	-0.001	-0.042	-0.080	-0.206	-0.016	-0.097	0.137	-0.115
Switzerland/U.S. Foreign Exchange Rate	0.019	-0.023	-0.087	-0.178	-0.073	-0.045	0.098	-0.085
Japan/U.S. Foreign Exchange Rate	0.000	-0.033	-0.094	-0.186	-0.040	-0.100	0.130	-0.098
U.S./U.K. Foreign Exchange Rate	0.005	-0.009	-0.026	-0.094	-0.034	0.002	-0.026	-0.003
Canada/U.S. Foreign Exchange Rate	0.005	0.035	0.018	-0.015	-0.009	0.072	-0.008	0.024
Prices								
PPI: Finished Goods	0.023	-0.084	-0.149	-0.247	-0.081	-0.032	-0.043	0.064
PPI: Finished Consumer Goods	0.015	-0.007	-0.079	-0.078	-0.056	0.080	-0.062	0.019
PPI: Intermediate Materials	0.013	-0.052	-0.099	-0.143	-0.032	0.014	-0.049	0.083
PPI: Crude Materials	0.008	0.006	0.023	0.016	-0.050	0.044	-0.054	0.018
Crude Oil, spliced WTI and Cushing	0.011	-0.078	-0.153	-0.269	-0.059	-0.056	0.023	0.016
PPI: Metals and metal products:	0.011	-0.033	-0.062	-0.052	-0.028	0.059	0.011	0.045
ISM Manufacturing: Prices Index	0.024	-0.048	-0.089	-0.065	-0.067	0.074	-0.063	0.062
CPI: All Items	0.027	-0.072	-0.114	-0.158	-0.069	0.035	-0.122	0.091
CPI: Apparel	0.013	-0.083	-0.157	-0.256	-0.047	-0.040	0.007	0.027
CPI: Transportation	0.023	-0.086	-0.148	-0.244	-0.085	-0.044	-0.042	0.072
CPI: Medical Care	0.016	-0.067	-0.138	-0.255	-0.050	0.011	-0.082	0.057
CPI: Commodities	0.014	-0.013	-0.029	-0.045	-0.015	0.100	-0.105	0.035
CPI: Durables	0.016	-0.074	-0.154	-0.254	-0.062	-0.055	0.010	0.029
CPI: Services	0.002	-0.016	-0.033	-0.102	-0.002	0.069	-0.139	0.065
CPI: All Items Less Food	-0.006	0.028	0.030	-0.027	0.056	0.136	-0.269	-0.035
CPI: All items less shelter	-0.023	0.022	0.026	-0.049	0.080	0.138	-0.242	-0.081
CPI: All items less medical care	0.001	0.012	0.004	-0.030	0.059	0.118	-0.241	0.020
Consumption								
Personal Cons. Expend.: Chain Index	0.000	0.085	-0.144	0.015	0.051	0.194	0.030	-0.094
Personal Cons. Exp: Durable goods	-0.035	0.101	0.067	-0.073	-0.091	0.021	-0.186	-0.074
Personal Cons. Exp: Nondurable goods	-0.001	-0.018	0.000	-0.020	0.068	0.002	-0.007	-0.155
Personal Cons. Exp: Services	0.002	-0.015	0.004	-0.037	0.059	-0.021	0.009	-0.227
Avg Hourly Earnings: Goods-Producing	-0.004	0.019	-0.028	0.026	-0.041	0.025	0.025	-0.004
Avg Hourly Earnings: Construction	-0.067	-0.047	0.025	0.039	-0.030	-0.135	-0.037	0.104

Table 21: The factor loadings of the 128 macro variables on the first 8 components. The sample period is 1971:08-1990:12. Variable descriptions are taken from McCracken and Ng (2016).

Variable	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Production								
Real Personal Income	0.063	0.069	0.016	0.046	0.035	0.026	-0.070	-0.071
Real personal income ex transfer receipts	0.085	0.082	0.020	0.042	0.037	-0.004	-0.078	-0.066
Real personal consumption expenditures	0.067	0.042	-0.034	0.066	0.003	0.174	-0.026	0.106
Real Manu. and Trade Industries Sales	0.119	0.058	-0.060	0.038	0.077	0.152	-0.012	-0.014
Retail and Food Services Sales	0.069	-0.044	-0.050	0.041	0.057	0.170	-0.055	0.066
IP Index	0.180	0.077	-0.062	-0.025	0.120	0.062	0.130	0.004
IP: Final Products and Nonindustrial Supplies	0.173	0.060	-0.061	-0.007	0.133	0.090	0.152	0.023
IP: Final Products (Market Group)	0.157	0.064	-0.056	-0.009	0.145	0.095	0.175	0.027
IP: Consumer Goods	0.125	0.066	-0.080	-0.012	0.116	0.125	0.215	0.073
IP: Durable Consumer Goods	0.119	0.081	-0.078	-0.018	0.104	0.098	0.138	-0.064
IP: Nondurable Consumer Goods	0.076	0.022	-0.044	-0.003	0.084	0.095	0.191	0.167
IP: Business Equipment	0.153	0.049	-0.016	0.003	0.132	0.021	0.064	-0.051
IP: Materials	0.158	0.080	-0.054	-0.036	0.089	0.029	0.089	-0.015
IP: Durable Materials	0.172	0.054	-0.067	-0.014	0.086	0.016	0.056	-0.073
IP: Nondurable Materials	0.113	0.047	-0.047	-0.032	0.046	0.060	0.080	0.002
IP: Manufacturing (SIC)	0.185	0.064	-0.068	-0.011	0.115	0.062	0.112	-0.036
IP: Residential Utilities	0.000	0.005	-0.014	-0.032	0.003	0.059	0.093	0.162
IP: Fuels	0.021	0.017	-0.001	-0.006	0.022	0.003	0.095	-0.008
ISM Manufacturing: Production Index	0.172	0.081	-0.098	-0.020	0.113	0.056	0.118	-0.052
Capacity Utilization: Manufacturing	0.072	0.007	-0.067	0.015	-0.008	-0.024	-0.045	-0.042
Labor market								
Help-Wanted Index for United States	0.105	0.032	-0.084	0.021	-0.016	-0.042	-0.055	-0.028
Ratio of Help Wanted/No. Unemployed	0.042	-0.040	0.049	0.028	-0.014	0.008	-0.010	-0.094
Civilian Labor Force	0.130	-0.012	0.007	0.027	-0.007	-0.063	-0.050	-0.070
Civilian Employment	-0.131	-0.032	0.054	0.000	-0.005	0.097	0.064	-0.024
Civilian Unemployment Rate	-0.041	0.019	-0.047	-0.037	0.019	0.145	0.045	-0.151
Average Duration of Unemployment (Weeks)	-0.022	-0.017	0.043	0.023	0.032	-0.041	0.000	0.025
Civilians Unemployed - Less Than 5 Weeks	-0.065	-0.016	0.047	-0.007	-0.024	0.013	0.033	0.034
Civilians Unemployed for 5–14 Weeks	-0.115	-0.019	0.013	-0.052	-0.031	0.190	0.082	-0.099
Civilians Unemployed - 15 Weeks & Over	-0.075	-0.011	0.036	-0.038	-0.048	0.097	0.029	-0.041
Civilians Unemployed for 15–26 Weeks	-0.092	-0.024	-0.020	-0.036	0.000	0.173	0.090	-0.097
Civilians Unemployed for 27 Weeks and Over	-0.087	-0.066	0.078	0.006	-0.015	-0.091	0.032	0.078
Initial Claims	0.190	-0.009	0.016	0.042	0.034	-0.105	-0.077	0.046
All Employees: Total nonfarm	0.190	0.022	-0.005	0.018	0.051	-0.142	-0.087	0.015
All Employees: Goods-Producing Industries	0.040	-0.018	0.059	-0.066	0.106	-0.061	-0.057	-0.023
All Employees: Mining and Logging: Mining	0.145	0.016	-0.001	0.081	-0.011	-0.083	-0.050	0.015
All Employees: Construction	0.183	0.030	-0.015	-0.015	0.069	-0.147	-0.077	0.011
All Employees: Manufacturing	0.181	0.030	-0.006	-0.021	0.058	-0.144	-0.073	-0.011
All Employees: Durable goods	0.143	0.024	-0.035	0.006	0.082	-0.123	-0.068	0.069
All Employees: Nondurable goods	0.157	-0.039	0.047	0.055	0.025	-0.062	-0.070	0.063
All Employees: Service-Providing Industries	0.168	-0.032	0.017	0.031	0.027	-0.103	-0.103	0.028
All Employees: Trade, Transportation & Utilities	0.161	-0.051	0.050	0.011	0.028	-0.121	-0.113	0.010
All Employees: Wholesale Trade	0.143	-0.027	0.010	0.041	0.012	-0.065	-0.060	0.025
All Employees: Retail Trade	0.123	-0.054	0.098	0.062	-0.019	-0.011	-0.047	0.069
All Employees: Financial Activities	0.020	-0.028	0.072	0.036	-0.037	0.027	0.059	0.073
All Employees: Government	0.064	0.063	-0.065	-0.004	0.040	-0.206	-0.135	0.090
Avg Weekly Hours: Goods-Producing	0.069	0.029	-0.066	-0.034	0.027	0.077	0.020	-0.056
Avg Weekly Overtime Hours: Manufacturing	0.065	0.063	-0.069	-0.006	0.024	-0.209	-0.130	0.074
Avg Weekly Hours: Manufacturing	0.138	-0.086	0.150	0.120	-0.171	0.071	0.072	0.010
ISM Manufacturing: Employment Index	0.110	-0.085	0.148	0.117	-0.123	0.067	0.067	-0.007
Housing								
Housing Starts: Total New Privately Owned	0.120	-0.071	0.136	0.102	-0.150	0.078	0.083	-0.024
Housing Starts, Northeast	0.132	-0.076	0.138	0.107	-0.161	0.062	0.058	0.024
Housing Starts, Midwest	0.134	-0.087	0.135	0.121	-0.177	0.058	0.063	0.017
Housing Starts, South	0.139	-0.078	0.124	0.122	-0.193	0.057	0.061	0.023
Housing Starts, West	0.121	-0.084	0.132	0.119	-0.156	0.058	0.064	0.002
New Private Housing Permits (SAAR)	0.128	-0.066	0.108	0.116	-0.180	0.059	0.076	-0.012
New Private Housing Permits, Northeast (SAAR)	0.120	-0.060	0.093	0.099	-0.172	0.038	0.036	0.040
New Private Housing Permits, Midwest (SAAR)	0.136	-0.083	0.134	0.118	-0.183	0.061	0.060	0.025
New Private Housing Permits, South (SAAR)	0.090	-0.136	-0.041	-0.181	0.065	0.135	-0.035	-0.138
New Private Housing Permits, West (SAAR)	0.078	0.016	-0.034	0.005	0.080	0.092	-0.018	-0.104
Orders & inventories								
ISM: PMI Composite Index	0.043	0.004	-0.006	0.008	0.066	0.057	-0.015	-0.062
ISM: New Orders Index	0.099	-0.073	0.103	-0.027	0.062	-0.064	-0.078	-0.055
ISM: Supplier Deliveries Index	0.073	-0.078	0.133	-0.050	0.092	-0.143	-0.100	-0.066
ISM: Inventories Index	-0.091	0.015	0.120	-0.028	-0.067	-0.206	-0.025	0.026
New Orders for Consumer Goods	-0.024	-0.001	0.010	0.075	0.005	-0.042	0.032	-0.031
New Orders for Durable Goods	-0.014	0.005	0.006	0.088	0.043	-0.117	0.081	-0.080

New Orders for Nondefense Capital Goods	-0.022	0.122	-0.031	0.097	-0.081	-0.036	0.085	0.046
Unfilled Orders for Durable Goods	-0.006	0.037	0.017	0.004	0.005	-0.069	0.088	0.015
Total Business Inventories	0.005	0.067	0.054	-0.005	-0.010	-0.038	0.095	-0.021
Total Business: Inventories to Sales Ratio	-0.009	0.044	0.039	0.018	-0.007	-0.047	0.109	-0.004
Money & credit								
M1 Money Stock	0.024	-0.008	-0.030	-0.042	-0.038	-0.013	0.042	-0.054
M2 Money Stock	0.002	0.037	-0.002	-0.004	-0.016	-0.026	0.083	0.008
Real M2 Money Stock	0.010	-0.019	-0.030	-0.006	-0.019	0.004	0.020	0.076
St. Louis Adjusted Monetary Base	0.012	-0.015	-0.034	0.004	-0.098	-0.097	0.021	0.151
Total Reserves of Depository Institutions	0.010	-0.016	-0.086	0.133	0.035	0.216	-0.311	0.062
Reserves Of Depository Institutions	0.010	-0.018	-0.089	0.126	0.031	0.217	-0.310	0.053
Commercial and Industrial Loans	0.008	-0.004	0.091	-0.158	-0.026	-0.230	0.249	-0.049
Real Estate Loans at All Commercial Banks	-0.036	0.002	-0.083	0.114	-0.014	0.256	-0.182	0.035
Total Nonrevolving Credit	0.101	-0.033	0.014	-0.165	-0.032	0.029	-0.075	-0.037
Nonrevolving consumer credit to Personal Income	0.100	-0.018	-0.002	-0.233	-0.116	0.030	-0.046	-0.067
S&P's Common Stock Price Index: Composite	0.089	-0.027	-0.020	-0.231	-0.135	0.065	-0.096	-0.052
S&P's Common Stock Price Index: Industrials	0.092	-0.026	-0.014	-0.263	-0.138	0.057	-0.080	-0.059
S&P's Composite Common Stock: Dividend Yield	0.091	-0.021	-0.018	-0.271	-0.152	0.053	-0.078	-0.064
S&P's Composite Common Stock: Price-Earnings Ratio	0.074	-0.030	-0.026	-0.265	-0.154	0.059	-0.070	-0.053
Effective Federal Funds Rate	0.063	-0.028	-0.019	-0.254	-0.148	0.051	-0.052	-0.071
3-Month AA Financial Commercial Paper Rate	0.058	-0.021	0.011	-0.264	-0.121	0.009	0.004	-0.082
3-Month Treasury Bill	0.039	-0.003	0.065	-0.254	-0.095	-0.040	0.048	-0.051
6-Month Treasury Bill	0.005	0.122	-0.140	-0.022	-0.210	-0.080	0.050	-0.011
1-Year Treasury Rate	0.035	0.133	-0.202	0.043	-0.206	-0.064	-0.028	-0.008
5-Year Treasury Rate	0.040	0.136	-0.198	0.054	-0.225	-0.068	-0.009	-0.013
10-Year Treasury Rate	0.052	0.124	-0.177	0.055	-0.233	-0.038	0.007	-0.008
Moody's Seasoned Aaa Corporate Bond Yield	0.021	0.142	-0.206	0.088	-0.194	-0.030	0.038	-0.023
Moody's Seasoned Baa Corporate Bond Yield	0.007	0.150	-0.219	0.079	-0.169	-0.045	0.028	-0.035
3-Month Commercial Paper Minus FEDFUNDS	-0.018	0.154	-0.224	0.074	-0.155	-0.054	0.038	-0.029
3-Month Treasury C Minus FEDFUNDS	-0.041	0.152	-0.221	0.072	-0.144	-0.026	0.053	-0.030
6-Month Treasury C Minus FEDFUNDS	0.006	0.066	0.028	-0.158	-0.017	0.017	0.035	0.411
1-Year Treasury C Minus FEDFUNDS	0.014	0.041	0.001	-0.139	-0.029	0.065	-0.023	0.374
5-Year Treasury C Minus FEDFUNDS	0.004	0.010	0.012	-0.123	-0.014	0.057	-0.061	0.306
10-Year Treasury C Minus FEDFUNDS	0.007	-0.052	-0.012	0.105	0.007	-0.020	-0.048	-0.349
Moody's Aaa Corporate Bond Minus FEDFUNDS	0.007	0.081	0.048	-0.090	-0.022	-0.075	0.144	0.157
Moody's Baa Corporate Bond Minus FEDFUNDS	-0.002	-0.200	-0.129	0.017	-0.008	-0.040	0.024	0.046
Trade Weighted U.S. Dollar Index: Major Currencies	-0.003	-0.203	-0.133	0.018	-0.010	-0.039	0.025	0.049
Switzerland/U.S. Foreign Exchange Rate	0.008	-0.190	-0.138	0.031	0.014	-0.019	0.003	0.003
Japan/U.S. Foreign Exchange Rate	-0.008	-0.142	-0.094	0.024	-0.002	-0.025	0.049	0.027
U.S./U.K. Foreign Exchange Rate	-0.004	-0.106	-0.069	0.007	-0.016	-0.014	-0.003	0.031
Canada/U.S. Foreign Exchange Rate	0.001	-0.039	-0.041	0.037	0.002	0.017	-0.033	-0.090
Prices								
PPI: Finished Goods	0.010	-0.239	-0.161	-0.003	0.008	-0.055	0.066	0.028
PPI: Finished Consumer Goods	0.012	-0.029	-0.035	-0.010	0.007	0.033	0.005	0.047
PPI: Intermediate Materials	0.001	-0.223	-0.154	0.014	0.001	-0.025	0.038	0.012
PPI: Crude Materials	0.007	0.009	0.011	0.011	0.032	0.002	-0.016	0.007
Crude Oil, spliced WTI and Cushing	0.002	-0.245	-0.165	0.007	-0.003	-0.042	0.057	0.039
PPI: Metals and metal products:	0.001	-0.051	-0.034	-0.025	-0.008	0.017	0.016	0.043
ISM Manufacturing: Prices Index	0.022	-0.063	-0.042	-0.024	0.028	0.008	0.009	-0.039
CPI: All Items	0.012	-0.224	-0.154	-0.001	0.013	-0.040	0.054	0.022
CPI: Apparel	0.005	-0.243	-0.166	0.000	-0.002	-0.044	0.063	0.030
CPI: Transportation	0.012	-0.239	-0.162	-0.005	0.011	-0.052	0.070	0.021
CPI: Medical Care	0.008	-0.223	-0.154	0.006	0.005	-0.022	0.042	0.025
CPI: Commodities	0.007	-0.048	-0.033	-0.002	0.007	0.015	-0.005	0.023
CPI: Durables	0.004	-0.240	-0.164	0.009	-0.002	-0.043	0.059	0.035
CPI: Services	0.008	-0.059	-0.041	0.001	0.020	-0.018	-0.013	-0.048
CPI: All Items Less Food	-0.003	-0.003	0.004	0.024	-0.016	-0.050	-0.044	-0.150
CPI: All items less shelter	-0.022	-0.013	0.015	0.014	-0.046	-0.053	-0.051	-0.059
CPI: All items less medical care	0.007	0.005	-0.006	-0.002	-0.012	-0.034	-0.015	-0.151
Consumption								
Personal Cons. Expend.: Chain Index	-0.003	0.031	-0.056	0.015	-0.072	0.123	-0.188	0.103
Personal Cons. Exp: Durable goods	-0.030	0.005	0.015	0.132	0.067	-0.127	0.058	-0.057
Personal Cons. Exp: Nondurable goods	0.003	0.000	0.007	-0.021	-0.017	0.016	-0.001	0.067
Personal Cons. Exp: Services	0.006	0.004	0.003	-0.015	-0.013	0.008	-0.003	0.068
Avg Hourly Earnings: Goods-Producing	-0.004	-0.021	-0.024	0.004	0.011	0.024	0.033	-0.048
Avg Hourly Earnings: Construction	-0.091	0.012	0.040	-0.038	-0.004	0.007	0.226	-0.049

Table 22: The factor loadings of the 128 macro variables on the first 8 components. The sample period is 1971:08-2018:12. Variable descriptions are taken from McCracken and Ng (2016).

Input	Method	Prior weight	n=24	n=36	n=48	n=60	n=84	n=120	EW
$f_t^{(n)}$	Regression	-	-31.5°	-23.0°	-16.5°	-8.4°	-5.6°	3.3**	-8.3°
	Random forest	-	-19.1°	-16.4°	-15.0°	-11.9°	-12.0°	-8.9°	-13.1°
	Neural network	-	-2.1°	0.9°	1.6°	2.4°	3.1°	3.6*	2.7°
	Regression	0.25	-17.3°	-12.3°	-7.1°	-1.3°	2.2*	9.8**	0.0°
	Random forest	0.25	-13.8°	-11.9°	-10.8°	-8.5°	-8.6°	-6.2°	-9.4°
	Neural network	0.25	-1.0°	1.0°	1.6°	2.1°	2.7°	3.0*	2.4°
	Regression	0.5	-7.4°	-5.0°	-1.3°	2.4°	5.7*	11.4**	4.1*
	Random forest	0.5	-8.9°	-7.6°	-6.9°	-5.4°	-5.4°	-3.8°	-5.9°
	Neural network	0.5	-0.3°	0.9°	1.3°	1.6°	2.0°	2.2*	1.8°
	Regression	0.75	-1.6°	-0.9°	1.1°	2.9°	5.0*	8.2**	4.1*
	Random forest	0.75	-4.3°	-3.7°	-3.3°	-2.6°	-2.6°	-1.8°	-2.8°
	Neural network	0.75	0.0°	0.6°	0.8°	0.9°	1.1°	1.2*	1.0°
	Random forest	fit	-10.9°	-9.5°	-8.6°	-6.7°	-6.8°	-4.7°	-7.4°
	Neural network	fit	-1.3°	0.6°	1.1°	1.6°	2.2°	2.5*	1.8°
	Random forest	similarity	-10.2°	-8.6°	-7.7°	-5.9°	-6.0°	-4.2°	-6.6°
	Neural network	similarity	-1.5°	0.1°	0.5°	0.9°	1.3°	1.4°	1.0°
$\Delta f_t^{(n)}$	Regression	-	19.9***	19.7***	18.0***	15.7***	13.9***	12.5***	16.0***
	Random forest	-	-6.8°	-6.5°	-7.2°	-7.4°	-10.4°	-12.9°	-10.4°
	Neural network	-	-1.5°	3.1**	4.6**	5.4**	5.3**	6.6**	6.0**
	Regression	0.25	20.0***	19.4***	18.2***	16.1***	14.8***	13.4***	16.6***
	Random forest	0.25	-1.2°	-1.2°	-1.7°	-2.1°	-4.5°	-6.7°	-4.3°
	Neural network	0.25	0.3°	3.5**	4.6**	5.1**	5.0**	6.0**	5.7**
	Regression	0.5	16.7***	16.1***	15.2***	13.6***	12.8***	11.6***	14.2***
	Random forest	0.5	1.8*	1.7°	1.3°	0.9°	-0.8°	-2.4°	-0.5°
	Neural network	0.5	1.1°	3.1**	3.8**	4.0**	4.1**	4.7**	4.6**
	Regression	0.75	10.0***	9.6***	9.2***	8.3***	7.9***	7.1***	8.6***
	Random forest	0.75	2.2*	2.1°	1.9°	1.6°	0.7°	-0.2°	0.9°
	Neural network	0.75	1.0°	1.9**	2.3**	2.4**	2.4**	2.7**	2.7**
	Random forest	fit	-0.1°	-0.2°	-0.7°	-1.2°	-3.4°	-5.5°	-3.1°
	Neural network	fit	0.5°	3.5**	4.6**	4.9**	4.9**	5.7**	5.5**
	Random forest	similarity	0.8*	0.9°	0.7°	0.6°	-1.3°	-2.7°	-1.0°
	Neural network	similarity	0.8°	2.8**	3.7**	3.9**	3.8**	4.3**	4.3**

Table 23: The out-of-sample R^2 (in %) for the forecasting of excess bond returns for times to maturity n =24, 36, 48, 60, 84 and 120 and an equally weighted portfolio of those. Linear regressions, random forests and neural networks are considered. The explanatory variables are the forward rates $f_t^{(n)}$ or the first differences of the forward rates $\Delta f_t^{(n)}$. *, ** and *** indicate that the R^2 is significantly larger than 0 at 10, 5 or 1% significance level respectively, and $\circ$ indicates no significance.

Input	Method	Prior weight	n=24	n=36	n=48	n=60	n=84	n=120	EW
$f_t^{(n)}$ and macro (final data)	Regression	-	-46.0°	-36.8°	-28.3°	-7.9°	-7.5°	-1.6°	-10.5°
	Random forest	-	-20.7°	-19.9°	-20.6°	-19.3°	-20.3°	-18.7°	-20.7°
	Neural network	-	2.0**	9.9***	12.4***	15.4***	17.3***	20.9***	16.8***
	Regression	0.25	-19.9°	-14.6°	-9.8°	3.0**	4.2**	9.9***	2.6***
	Random forest	0.25	-11.6°	-11.6°	-12.2°	-11.5°	-12.4°	-11.6°	-12.5°
	Neural network	0.25	7.1**	11.9***	13.4***	15.4***	16.6***	19.2***	16.6***
	Regression	0.5	-3.5°	-1.1°	1.1**	8.0**	9.3**	14.0***	8.7**
	Random forest	0.5	-5.1°	-5.5°	-6.0°	-5.6°	-6.4°	-6.2°	-6.4°
	Neural network	0.5	8.4**	11.0***	11.7***	12.8***	13.5***	15.1***	13.7***
	Regression	0.75	3.1*	3.7**	4.4***	7.0**	7.9**	10.7***	7.8**
	Random forest	0.75	-1.3°	-1.6°	-1.9°	-1.8°	-2.3°	-2.3°	-2.2°
	Neural network	0.75	6.1**	7.0***	7.2***	7.7***	7.9***	8.7***	8.2***
	Random forest	fit	-9.6°	-9.6°	-10.2°	-9.7°	-10.8°	-10.1°	-10.7°
	Neural network	fit	5.7**	10.4***	11.9***	13.9***	15.1***	17.8***	15.0***
	Random forest	similarity	-5.5°	-5.4°	-5.5°	-4.9°	-5.7°	-5.1°	-5.6°
	Neural network	similarity	8.6**	11.0***	11.9***	13.3***	13.5***	15.0***	13.9***
$\Delta f_t^{(n)}$ and macro (final data)	Regression	-	13.7***	13.9***	10.3***	12.4***	16.5***	23.2***	15.9***
	Random forest	-	-16.8°	-8.8°	-4.9°	0.4**	3.9**	10.7***	2.2**
	Neural network	-	-35.8°	-15.0°	-9.9°	-2.4°	1.7***	11.6***	-1.1°
	Regression	0.25	17.6***	16.8***	15.2***	16.8***	20.4***	24.6***	20.1***
	Random forest	0.25	-6.9°	-1.3°	1.6*	5.3**	8.1**	12.9***	7.0**
	Neural network	0.25	-12.0°	0.4**	3.6**	8.3***	11.4***	18.3***	10.1***
	Regression	0.5	16.7***	15.4***	15.1***	16.2***	18.9***	21.2***	18.9***
	Random forest	0.5	-0.9°	2.6*	4.6*	6.9**	8.8**	11.8***	8.2**
	Neural network	0.5	1.9**	8.0**	9.7**	12.2***	14.4***	18.6***	14.0***
	Regression	0.75	10.8***	9.8***	10.0***	10.6***	12.1***	13.0***	12.2***
	Random forest	0.75	1.4°	3.1*	4.1*	5.1**	6.1**	7.5***	5.9**
	Neural network	0.75	5.9**	7.9**	8.5**	9.5***	10.6***	12.5***	10.6***
	Random forest	fit	-6.1°	-0.7°	2.1*	5.6**	8.2**	12.7***	7.2**
	Neural network	fit	-11.5°	0.2**	3.3***	7.9***	11.1***	17.8***	9.7***
	Random forest	similarity	-0.7°	2.5*	4.6*	6.7**	8.1**	10.8**	7.7**
	Neural network	similarity	7.6**	12.3***	14.1***	16.1***	17.0***	19.9***	17.1***

Table 24: The out-of-sample R^2 (in %) for the forecasting of excess bond returns for times to maturity n =24, 36, 48, 60, 84 and 120 and an equally weighted portfolio of those. Linear regressions, random forests and neural networks are considered. The explanatory variables are the forward rates $f_t^{(n)}$ or the first differences of the forward rates $\Delta f_t^{(n)}$ in combination with (principal components of) a set of 128 macro variables.

Input	Method	Prior weight	n=24	n=36	n=48	n=60	n=84	n=120	EW
$f_t^{(n)}$ and macro (real time)	Regression	-	-39.4°	-24.4°	-14.6°	-7.9°	-6.2°	1.7**	-7.1°
	Random forest	-	-22.7°	-19.4°	-20.0°	-16.2°	-13.9°	-8.7°	-15.7°
	Neural network	-	-13.2°	-1.5°	2.3**	6.7**	10.6**	17.1***	9.0**
	Regression	0.25	-20.6°	-11.4°	-5.2°	-0.2°	2.1*	9.1**	1.4*
	Random forest	0.25	-13.5°	-12.3°	-13.0°	-10.7°	-9.3°	-5.9°	-10.4°
	Neural network	0.25	-3.9°	3.8*	6.2**	9.4**	12.2**	17.1***	11.4**
	Regression	0.5	-7.7°	-3.0°	0.4*	3.6*	5.9*	11.2**	5.4*
	Random forest	0.5	-6.7°	-6.7°	-7.3°	-6.1°	-5.4°	-3.4°	-6.0°
	Neural network	0.5	1.4°	5.8*	7.2**	9.2**	11.0**	14.3***	10.7**
	Regression	0.75	-0.8°	0.8°	2.1°	3.7*	5.2*	8.2**	4.9*
	Random forest	0.75	-2.2°	-2.6°	-2.9°	-2.6°	-2.3°	-1.5°	-2.6°
	Neural network	0.75	2.7°	4.5*	5.1**	6.0**	6.9**	8.6***	6.9**
	Random forest	fit	-11.2°	-10.9°	-11.7°	-10.0°	-8.8°	-5.9°	-9.8°
	Neural network	fit	-4.3°	3.0*	5.4**	8.5**	11.3**	16.1***	10.4**
	Random forest	similarity	-9.9°	-8.9°	-9.6°	-7.8°	-6.7°	-4.1°	-7.5°
	Neural network	similarity	1.9°	6.5**	8.2**	10.4**	11.8***	14.9***	11.6***
$\Delta f_t^{(n)}$ and macro (real time)	Regression	-	10.7***	14.1***	17.7***	19.1***	18.6***	23.5***	18.9***
	Random forest	-	-18.5°	-10.7°	-6.8°	-1.8°	1.3**	7.2**	-0.5°
	Neural network	-	-34.5°	-14.4°	-8.4°	-2.2°	1.6**	10.5***	-1.2°
	Regression	0.25	13.7***	15.5***	18.2***	19.0***	19.5***	23.5***	19.6***
	Random forest	0.25	-8.1°	-2.8°	0.2*	3.6*	6.0**	10.1**	4.9**
	Neural network	0.25	-13.6°	-0.8°	3.2**	7.3**	10.5**	16.9***	8.9**
	Regression	0.5	12.9***	13.7***	15.4***	15.8***	16.7***	19.6***	16.7***
	Random forest	0.5	-1.5°	1.7°	3.7*	5.7*	7.3**	9.8**	6.8**
	Neural network	0.5	-0.9°	6.1*	8.5**	10.8**	13.2**	17.2***	12.5**
	Regression	0.75	8.3***	8.5***	9.3***	9.5***	10.2***	11.8***	10.2***
	Random forest	0.75	1.2°	2.6°	3.6*	4.5*	5.3**	6.5**	5.2**
	Neural network	0.75	3.7°	6.4*	7.4**	8.4**	9.7**	11.6***	9.5**
	Random forest	fit	-7.3°	-2.1°	0.8*	4.0*	6.3**	10.2**	5.3**
	Neural network	fit	-12.8°	-0.6°	3.2**	7.3**	10.5**	16.7***	9.0**
	Random forest	similarity	-2.6°	0.7°	2.9*	5.0*	6.3**	8.8**	5.8**
	Neural network	similarity	3.3*	9.6**	12.0**	14.2***	15.6***	18.8***	15.4***

Table 25: The out-of-sample R^2 (in %) for the forecasting of excess bond returns for times to maturity n =24, 36, 48, 60, 84 and 120 and an equally weighted portfolio of those. Linear regressions, random forests and neural networks are considered. The explanatory variables are the forward rates $f_t^{(n)}$ or the first differences of the forward rates $\Delta f_t^{(n)}$ in combination with (principal components of) a set of 128 macro variables.

Method	Prior weight	n=24	n=36	n=48	n=60	n=84	n=120	EW
Regression	-	-9.1°	-5.0°	-4.7°	-1.5°	3.8**	11.9***	1.3**
Random forest	-	-17.7°	-13.0°	-10.6°	-7.1°	-4.0°	1.7°	-5.9°
Neural network	-	-8.7°	-7.1°	-6.2°	-4.5°	-3.2°	0.6*	-3.8°
Regression	0.25	-0.9°	1.9*	2.9*	5.4**	10.0**	17.4***	8.3**
Random forest	0.25	-10.6°	-7.2°	-5.1°	-2.6°	0.2°	4.7°	-1.2°
Neural network	0.25	-5.6°	-4.5°	-3.6°	-2.4°	-1.2°	1.7*	-1.7°
Regression	0.5	3.4*	5.1*	6.2*	8.0*	11.4**	17.2***	10.4**
Random forest	0.5	-5.3°	-3.1°	-1.5°	0.1°	2.2°	5.4°	1.3°
Neural network	0.5	-3.2°	-2.4°	-1.8°	-1.0°	-0.1°	2.0*	-0.4°
Regression	0.75	3.7*	4.5*	5.2*	6.2**	8.1**	11.4***	7.6**
Random forest	0.75	-1.8°	-0.7°	0.2°	1.0°	2.2°	3.9°	1.7°
Neural network	0.75	-1.3°	-0.9°	-0.6°	-0.2°	0.4°	1.4*	0.2°
Random forest	fit	-9.5°	-6.2°	-4.2°	-1.8°	0.9°	5.2°	-0.5°
Neural network	fit	-5.6°	-4.5°	-3.7°	-2.5°	-1.3°	1.5°	-1.8°
Random forest	similarity	-6.2°	-3.5°	-1.4°	0.5°	2.9°	6.2*	1.7°
Neural network	similarity	-4.4°	-3.8°	-3.1°	-2.3°	-1.5°	0.7°	-1.8°

Table 26: The out-of-sample R^2 (in %) for the forecasting of excess bond returns for times to maturity n =24, 36, 48, 60, 84 and 120 and an equally weighted portfolio of those. Linear regressions, random forests and neural networks are considered. The explanatory variables are the yield spread s_t, 12 month bond momentum $d^{(n)}_{t-12:t}$, 12 month equity returns $q_{t-12:t}$ and 12 month commodity returns $c^{(n)}_{t-12:t}$.

P-value (%)	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥

Setting	Method	Prior	Reg	Rf	NN	Reg	Rf	NN	Reg	Rf	NN	Reg	Rf	NN	Rf	NN	Rf
			-	-	-	0.25	0.25	0.25	0.5	0.5	0.5	0.75	0.75	0.75	fit	fit	sim
$\Delta f_t^{(n)}$	Random forest	-	🟦														
	Neural network	-	🟦														
	Regression	0.25		🟥	🟥												
	Random forest	0.25	🟦	🟥		🟦											
	Neural network	0.25			🟥	🟦											
	Regression	0.5		🟥	🟥		🟥	🟥									
	Random forest	0.5				🟦			🟦								
	Neural network	0.5				🟦			🟦								
	Regression	0.75		🟥	🟥		🟥	🟥	🟦	🟥	🟥						
	Random forest	0.75				🟦			🟦			🟦					
	Neural network	0.75				🟦			🟦			🟦					
	Random forest	fit	🟦	🟥		🟦			🟦			🟦					
	Neural network	fit			🟥	🟦			🟦			🟦					
	Random forest	similarity				🟦			🟦			🟦					
	Neural network	similarity				🟦			🟦			🟦					
$\Delta f_t^{(n)}$ and macro (real time)	Random forest	-	🟦														
	Neural network	-	🟦														
	Regression	0.25		🟥	🟥												
	Random forest	0.25	🟦	🟥		🟦											
	Neural network	0.25			🟥												
	Regression	0.5		🟥	🟥		🟥										
	Random forest	0.5		🟥		🟦	🟥		🟦								
	Neural network	0.5		🟥	🟥			🟥									
	Regression	0.75		🟥			🟥			🟥							
	Random forest	0.75		🟥					🟦			🟦					
	Neural network	0.75		🟥	🟥		🟥			🟥							
	Random forest	fit	🟦	🟥		🟦			🟦	🟦		🟦		🟦			
	Neural network	fit			🟥	🟦			🟦		🟦						
	Random forest	similarity		🟥		🟦			🟦			🟦					
	Neural network	similarity		🟥	🟥		🟥								🟥	🟥	
s_t, $d_{t-12:t}^{(n)}$, $q_{t-12:t}$, $c_{t-12:t}$	Random forest	-															
	Neural network	-															
	Regression	0.25	🟥	🟥													
	Random forest	0.25		🟥													
	Neural network	0.25			🟥												
	Regression	0.5	🟥	🟥	🟥		🟥	🟥									
	Random forest	0.5		🟥					🟦								
	Neural network	0.5			🟥												
	Regression	0.75		🟥	🟥		🟥	🟥		🟥	🟥						
	Random forest	0.75			🟥							🟦					
	Neural network	0.75										🟦					
	Random forest	fit		🟥					🟦	🟦		🟦					
	Neural network	fit							🟦		🟦	🟦		🟦			
	Random forest	similarity							🟦			🟦					
	Neural network	similarity									🟦	🟦		🟦			

Table 27: P-values of 2-sided pairwise tests comparing predictive accuracy of forecasts of 2 year excess returns, using the adjusted Diebold-Mariano statistic as proposed by Harvey et al. (1997). Blue indicates that the column method has a significantly higher R^2_{oos} than the row method, red indicates the reverse.

P-value (%)	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥

Setting	Method	Prior	Reg	Rf	NN	Reg	Rf	NN	Reg	Rf	NN	Reg	Rf	NN	Rf	NN	Rf
			-	-	-	0.25	0.25	0.25	0.5	0.5	0.5	0.75	0.75	0.75	fit	fit	sim
$\Delta f_t^{(n)}$	Random forest	-	🟦														
	Neural network	-		🟥													
	Regression	0.25		🟥													
	Random forest	0.25	🟦	🟥	🟦	🟦											
	Neural network	0.25		🟥			🟥										
	Regression	0.5		🟥			🟥										
	Random forest	0.5	🟦	🟥	🟦	🟦	🟥	🟦	🟦								
	Neural network	0.5		🟥			🟥			🟥							
	Regression	0.75		🟥			🟥		🟦	🟥							
	Random forest	0.75		🟥	🟦	🟦		🟦	🟦		🟦	🟦					
	Neural network	0.75		🟥					🟦		🟦	🟦	🟥				
	Random forest	fit	🟦	🟥	🟦	🟦	🟥	🟦	🟦		🟦	🟦					
	Neural network	fit		🟥			🟥			🟥			🟥		🟥		
	Random forest	similarity		🟥	🟦	🟦		🟦	🟦		🟦	🟦				🟦	
	Neural network	similarity		🟥			🟥			🟥			🟥	🟥	🟥		🟥
$\Delta f_t^{(n)}$ and macro (real time)	Random forest	-															
	Neural network	-															
	Regression	0.25															
	Random forest	0.25				🟦											
	Neural network	0.25		🟥													
	Regression	0.5															
	Random forest	0.5	🟦			🟦			🟦								
	Neural network	0.5		🟥			🟥			🟥							
	Regression	0.75				🟦			🟦								
	Random forest	0.75	🟦			🟦			🟦		🟦	🟦					
	Neural network	0.75				🟦			🟦				🟥				
	Random forest	fit				🟦					🟦						
	Neural network	fit		🟥													
	Random forest	similarity	🟦			🟦			🟦		🟦						
	Neural network	similarity		🟥			🟥			🟥			🟥	🟥	🟥		🟥
s_t, $d_{t-12:t}^{(n)}$, $q_{t-12:t}$, $c_{t-12:t}$	Random forest	-															
	Neural network	-															
	Regression	0.25		🟥	🟥												
	Random forest	0.25				🟦											
	Neural network	0.25				🟦											
	Regression	0.5			🟥		🟥	🟥									
	Random forest	0.5							🟦								
	Neural network	0.5							🟦								
	Regression	0.75			🟥			🟥			🟥						
	Random forest	0.75							🟦			🟦					
	Neural network	0.75							🟦			🟦					
	Random forest	fit				🟦			🟦								
	Neural network	fit				🟦			🟦			🟦					
	Random forest	similarity							🟦								
	Neural network	similarity				🟦			🟦			🟦					

Table 28: P-values of 2-sided pairwise tests comparing predictive accuracy of forecasts of 10 year excess returns, using the adjusted Diebold-Mariano statistic as proposed by Harvey et al. (1997). Blue indicates that the column method has a significantly higher R^2_{oos} than the row method, red indicates the reverse.

	$xr_t^{(24)}$	$xr_t^{(120)}$	$f_t^{(24)}$	$f_t^{(120)}$	$\Delta f_t^{(24)}$	$\Delta f_t^{(120)}$	s_t	$c_{t-12:t}$	$q_{t-12:t}$	$d_{t-12:t}^{(EW)}$
Mean	0.59	1.56	5.21	6.98	-0.07	-0.07	1.60	0.00	0.07	0.19
Minimum	-5.97	-17.35	0.10	2.00	-5.21	-8.40	-4.27	-0.04	-0.59	-1.00
Maximum	5.65	17.83	16.03	14.48	6.65	7.38	4.27	0.06	0.42	1.00
St. dev.	1.75	5.40	3.60	2.37	1.68	1.61	1.43	0.02	0.16	0.98
Skewness	-0.14	-0.14	0.40	0.35	0.02	-0.24	-0.69	0.13	-1.01	-0.40
Kurtosis	1.10	0.77	-0.24	0.07	1.33	8.03	0.51	-0.41	1.46	-1.84

Table 29: Descriptive statistics of the excess returns $xr_{t-12:t}^{(n)}$, forward rates $f_t^{(n)}$, first differences of the forward rates $\Delta f_t^{(n)}$, the yield spread s_t, 12 month bond momentum $d_{t-12:t}^{(n)}$, 12 month equity returns $q_{t-12:t}$ and 12 month commodity returns $c_{t-12:t}^{(n)}$. The period is 1971:08-2018:12, with n being the time to maturity.

Setting	Method	n=24	n=36	n=48	n=60	n=84	n=120	EW
$\Delta f_t^{(n)}$	Random forest	0.01	0.03	0.04	0.04	0.06	0.08	0.04
		(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
	Neural network	0.01	0.02	0.02	0.03	0.04	0.04	0.02
		(0.01)	(0.03)	(0.07)	(0.09)	(0.08)	(0.10)	(0.08)
$\Delta f_t^{(n)}$ and macro	Random forest	0.01	0.02	0.03	0.04	0.05	0.08	0.04
		(0.01)	(0.01)	(0.02)	(0.01)	(0.01)	(0.01)	(0.01)
	Neural network	0.01	0.02	0.02	0.03	0.04	0.06	0.03
		(0.07)	(0.05)	(0.06)	(0.06)	(0.06)	(0.02)	(0.06)
s_t, $d_{t-12:t}^{(n)}$, $q_{t-12:t}$, $c_{t-12:t}^{(n)}$	Random forest	0.01	0.02	0.02	0.03	0.05	0.07	0.03
		(0.01)	(0.01)	(0.02)	(0.04)	(0.04)	(0.04)	(0.03)
	Neural network	0.01	0.03	0.04	0.05	0.08	0.11	0.05
		(0.01)	(0.01)	(0.02)	(0.02)	(0.02)	(0.02)	(0.02)

Table 30: Estimated alphas in a regression of linear regression investment returns on machine learning investment returns. P-values for a two-sided test of $\alpha = 0$ are given between brackets.

	PC 1	PC 2	PC 3	PC 4	PC 5	PC 6	PC 7	PC 8	PC 9	PC 10
n=12	0.35	0.69	0.10	0.27	0.34	0.16	0.25	0.29	0.10	0.15
n=24	0.36	0.36	-0.04	-0.02	-0.13	-0.12	-0.23	-0.43	-0.33	-0.60
n=36	0.34	0.14	-0.09	-0.17	-0.22	-0.19	-0.22	-0.37	-0.04	0.75
n=48	0.33	-0.01	-0.10	-0.23	-0.24	-0.19	-0.08	0.17	0.80	-0.23
n=60	0.29	-0.09	0.08	-0.40	-0.09	-0.18	-0.13	0.69	-0.46	0.00
n=72	0.31	-0.21	-0.43	0.02	-0.22	0.01	0.77	-0.08	-0.15	-0.02
n=84	0.35	-0.24	0.17	0.02	-0.20	0.85	-0.18	-0.02	0.01	0.01
n=96	0.21	-0.22	0.02	-0.51	0.76	0.05	0.08	-0.25	0.06	-0.04
n=108	0.25	-0.30	-0.54	0.52	0.31	-0.10	-0.40	0.15	-0.02	0.01
n=120	0.34	-0.36	0.67	0.39	0.02	-0.35	0.14	-0.10	0.01	0.00
λ	12.70	2.26	1.59	0.74	0.41	0.29	0.21	0.11	0.07	0.04

Table 31: This Table shows the loadings of the 10 principal components of the 10 changes in forward rates $\Delta f_t^{(m)} = f_t^{(n)} - f_{t-12}^{(n)}$ and their corresponding eigenvalues λ. Times to maturity are n and the period is 1971:08-2018:12.

Prior weight	n=24	n=36	n=48	n=60	n=84	n=120	EW
-	19.5***	18.1***	15.9***	13.2***	12.1***	8.4***	13.3***
0.25	17.5***	16.2***	14.5***	12.2***	11.3***	8.3***	12.4***
0.5	13.6***	12.5***	11.3***	9.7***	9.1***	6.9***	9.9***
0.75	7.8***	7.1***	6.5***	5.6***	5.3***	4.1***	5.7***

Table 32: The out-of-sample R^2 (in %) of linear regressions with the first, second, and average of third, fourth and fifth principal component of the first differences of the forward rates $\Delta f_t^{(n)}$ as explanatory variables. *** indicates significance at 1% level.

	n=24	n=36	n=48	n=60	n=84	n=120	EW
Const.	0.62 (8.72)	1.08 (8.38)	1.52 (8.37)	1.74 (7.69)	2.22 (7.10)	2.62 (6.05)	1.63 (7.40)
PC 1	0.02 (0.78)	0.03 (0.55)	0.03 (0.37)	0.01 (0.11)	-0.05 (-0.45)	-0.13 (-0.75)	-0.02 (-0.19)
PC 2	0.38 (7.84)	0.70 (8.22)	0.97 (8.16)	1.17 (8.02)	1.57 (8.13)	2.05 (7.89)	1.14 (8.22)
PC 3, 4 & 5	-0.13 (-2.30)	-0.30 (-2.94)	-0.45 (-3.00)	-0.49 (-2.63)	-0.93 (-3.26)	-1.31 (-3.23)	-0.60 (-3.10)

Table 33: This Table reports the coefficients of a regression of excess returns on the first, second, and average of third, fourth and fifth principal component of the first differences of the forward rates $\Delta f_t^{(n)}$ and an intercept for the period 1971:08:2018:12. T statistics are between brackets, all significant at a 5% level except for PC 1.

B Robustness checks

Input	Method	Prior weight	n=24	n=36	n=48	n=60	n=84	n=120	EW
$\Delta f_t^{(n)}$ and macro (real time)	Regression	-	6.0**	12.5**	14.9***	17.2***	13.3***	22.2***	15.3***
	Random forest	-	-31.4°	-16.3°	-8.3°	0.9**	6.5**	17.3***	3.7**
	Neural network	-	-60.4°	-26.9°	-17.8°	-6.2°	-1.8°	12.6***	-6.7°
	Regression	0.25	12.1**	16.2**	18.5***	20.1***	18.5***	25.9***	20.1***
	Random forest	0.25	-16.0°	-5.2°	0.9**	7.1**	11.3**	18.9***	9.8**
	Neural network	0.25	-30.9°	-8.8°	-2.4°	5.0**	8.8**	19.2***	6.2**
	Regression	0.5	13.2**	15.3**	17.2***	18.1***	18.0***	23.4***	19.2***
	Random forest	0.5	-5.6°	1.2*	5.3**	9.0**	11.9**	16.5***	11.2**
	Neural network	0.5	-11.0°	1.7**	5.7**	9.7**	12.7**	19.3***	11.7**
	Regression	0.75	9.1**	9.9**	11.0***	11.4***	11.9***	14.8***	12.5***
	Random forest	0.75	-0.3°	2.9*	5.0**	6.6**	8.1**	10.2***	8.0**
	Neural network	0.75	-0.7°	4.6**	6.5**	8.0**	9.7**	12.9***	9.6**
	Random forest	fit	-14.8°	-4.4°	1.5*	7.5**	11.6**	18.8***	10.1**
	Neural network	fit	-28.9°	-7.7°	-1.5°	5.5**	9.3**	19.3***	6.9**
	Random forest	similarity	-8.3°	-1.6°	2.9*	6.8**	9.4**	14.4**	8.7**
	Neural network	similarity	-9.4°	2.8*	7.6**	11.5**	13.8**	20.0**	13.0**

Table 34: The out-of-sample R^2 (in %) with the out-of-sample period starting in 1999:08 instead of 1990:01. *, ** and *** indicate that the R^2 is significantly larger than 0 at 10, 5 or 1% significance level respectively, and ∘ indicates no significance.

Strategy	Setting	Method	n=24	n=36	n=48	n=60	n=84	n=120	EW
		Benchmark	0.84	0.75	0.70	0.64	0.58	0.53	0.63
1+0.5z	$\Delta f_t^{(n)}$	Regression	0.89	0.79	0.74	0.67	0.61	0.56	0.67
		Random forest	0.85	0.76	0.71	0.64	0.57	0.51	0.62
		Neural network	0.85	0.76	0.72	0.65	0.58	0.53	0.64
1+0.5z	$\Delta f_t^{(n)}$ and macro	Regression	0.89	0.80	0.75	0.68	0.62	0.58	0.68
		Random forest	0.84	0.75	0.71	0.64	0.58	0.54	0.64
		Neural network	0.86	0.76	0.72	0.65	0.59	0.54	0.64
1+0.5z	s_t, $d_{t-12:t}^{(n)}$, $q_{t-12:t}$, $c_{t-12:t}$	Regression	0.84	0.74	0.69	0.63	0.59	0.56	0.64
		Random forest	0.81	0.72	0.67	0.61	0.56	0.53	0.61
		Neural network	0.82	0.73	0.69	0.63	0.57	0.53	0.62
1+2z	$\Delta f_t^{(n)}$	Regression	0.93	0.83	0.78	0.70	0.63	0.58	0.69
		Random forest	0.83	0.74	0.68	0.59	0.50	0.43	0.56
		Neural network	0.88	0.79	0.74	0.66	0.57	0.51	0.64
1+2z	$\Delta f_t^{(n)}$ and macro	Regression	0.95	0.86	0.81	0.75	0.68	0.64	0.74
		Random forest	0.81	0.73	0.69	0.63	0.57	0.54	0.62
		Neural network	0.83	0.75	0.70	0.64	0.58	0.55	0.63
1+2z	s_t, $d_{t-12:t}^{(n)}$, $q_{t-12:t}$, $c_{t-12:t}$	Regression	0.78	0.68	0.63	0.58	0.56	0.57	0.60
		Random forest	0.71	0.63	0.59	0.55	0.52	0.51	0.56
		Neural network	0.77	0.68	0.65	0.59	0.55	0.52	0.59

Table 35: The Sharpe Ratios for timing strategies with average positive duration exposure. The benchmark strategy is to buy a $\frac{n}{12}$ year bond and to sell a 1 year bond every month. After a year, the former is sold and the latter has expired, such that the portfolio consists of 12 bonds at each point in time. For the active strategies, model forecasts are interpreted as signal and a z-score is computed. Investment equals $1 + 0.5z$ or $1 + 2z$.

Setting	Prior weight	n=24	n=36	n=48	n=60	n=84	n=120	EW
$\Delta f_t^{(n)}$	-	-6.6°	-1.4°	-3.6°	-2.3°	-9.4°	-9.3°	-8.1°
	0.25	0.2**	3.2**	1.5**	2.2**	-3.4°	-4.0°	-2.1°
	0.5	3.6**	5.0**	3.8**	4.1**	0.1°	-0.7°	1.2°
	0.75	3.5**	3.9**	3.3**	3.4**	1.3°	0.7°	1.9°
	fit	1.1**	3.9**	2.2**	2.7**	-2.4°	-2.9°	-1.0°
	similarity	4.3**	5.6**	4.1**	4.6**	0.8°	0.5°	2.3°
$\Delta f_t^{(n)}$ and macro	-	-13.6°	-9.7°	-6.8°	-2.6°	-0.9°	2.2**	1.2**
	0.25	-0.3°	1.4*	3.9**	5.7**	7.9**	9.4**	10.1**
	0.5	6.4*	6.8*	8.6**	8.9**	11.0**	11.4**	12.8**
	0.75	6.5*	6.3*	7.3**	7.0**	8.4**	8.2**	9.5**
	fit	-0.1°	1.8*	4.1**	5.8**	8.4**	10.2***	10.1**
	similarity	6.1*	6.1*	9.5**	8.6**	11.2**	10.5***	11.7**
s_t, $d_{t-12:t}^{(n)}$, $q_{t-12:t}$, $c_{t-12:t}$	-	-69.2°	-49.0°	-48.0°	-34.7°	-37.1°	-24.9°	-35.8°
	0.25	-40.5°	-26.4°	-24.5°	-15.6°	-14.7°	-4.3°	-14.5°
	0.5	-19.4°	-10.8°	-8.6°	-3.4°	-1.0°	6.7**	-1.4°
	0.75	-5.9°	-1.9°	-0.5°	1.8°	3.9°	8.1**	3.4°
	fit	-34.7°	-22.2°	-20.1°	-12.0°	-10.3°	-0.4°	-10.4°
	similarity	-20.2°	-10.3°	-7.0°	-1.8°	0.3*	7.7**	0.3°

Table 36: The out-of-sample R^2 (in %) of gradient-boosted trees. *, ** and *** indicate that the R^2 is significantly larger than 0 at 10, 5 or 1% significance level respectively, and $\circ$ indicates no significance.

Setting	Prior weight	n=24	n=36	n=48	n=60	n=84	n=120	EW
$\Delta f_t^{(n)}$	-	-0.2°	0.4*	-0.3°	-0.5°	-3.3°	-5.1°	-2.8°
	0.25	2.6*	2.8*	2.3°	2.0°	-0.3°	-1.8°	0.3°
	0.5	3.6*	3.6*	3.2°	2.9°	1.3°	0.1°	1.8°
	0.75	2.7*	2.7*	2.4°	2.2°	1.4°	0.7°	1.7°
	fit	3.2*	3.4*	2.8*	2.4°	0.3°	-1.1°	0.9°
	similarity	4.1*	4.4*	4.2*	4.1*	2.3°	1.2°	2.9°
$\Delta f_t^{(n)}$ and macro	-	-13.2°	-6.7°	-3.1°	1.8**	5.5**	12.3***	3.8**
	0.25	-5.1°	-0.6°	2.1*	5.4**	8.2**	13.0***	7.1**
	0.5	-0.1°	2.5°	4.3*	6.3**	8.1**	11.2***	7.6**
	0.75	1.6°	2.7°	3.6*	4.5**	5.4**	6.8***	5.2**
	fit	-5.4°	-0.9°	1.8*	5.0**	7.7**	12.4***	6.7**
	similarity	-0.3°	2.2°	4.3*	6.3**	7.8**	10.9***	7.4**
s_t, $d_{t-12:t}^{(n)}$, $q_{t-12:t}$, $c_{t-12:t}$	-	-26.4°	-18.7°	-14.1°	-8.7°	-3.2°	5.8**	-6.2°
	0.25	-14.5°	-9.1°	-5.5°	-1.9°	2.7*	9.7**	0.7°
	0.5	-6.1°	-2.8°	-0.4°	1.8°	5.3*	10.1**	4.0°
	0.75	-1.3°	0.2°	1.5°	2.4°	4.4*	6.8**	3.8°
	fit	-12.9°	-8.0°	-4.7°	-1.3°	3.3*	9.9**	1.3°
	similarity	-7.5°	-3.7°	-0.7°	1.7°	5.0*	9.7**	3.6*

Table 37: The out-of-sample R^2 (in %) of extreme trees. *, ** and *** indicate that the R^2 is significantly larger than 0 at 10, 5 or 1% significance level respectively, and $\circ$ indicates no significance.

Setting	Layers	Nodes	Prior weight	n=24	n=36	n=48	n=60	n=84	n=120	EW
$\Delta f_t^{(n)}$	1	5	-	2.7**	6.1**	7.7**	8.1**	7.7**	7.9**	8.2**
	2	3	-	1.1*	4.6**	6.9**	8.2***	8.7***	10.1***	8.9***
	3	3	-	2.7*	4.4**	4.4**	5.2***	5.4***	5.1***	4.5**
	1	5	0.25	3.7**	5.9**	7.1**	7.2**	7.1**	7.2**	7.5**
	2	3	0.25	2.3*	4.6**	6.3**	7.1***	7.5***	8.5***	7.8***
	3	3	0.25	2.5*	3.6**	3.5**	4.1***	4.2***	4.0***	3.6**
	1	5	0.5	3.5**	4.9**	5.7**	5.6**	5.7**	5.7**	6.0**
	2	3	0.5	2.5*	3.9**	5.0**	5.4***	5.7***	6.3***	5.9***
	3	3	0.5	2.0*	2.6**	2.5**	2.9***	2.9***	2.8***	2.6**
	1	5	0.75	2.3**	2.9**	3.3**	3.2**	3.3**	3.3**	3.4**
	2	3	0.75	1.8*	2.3**	2.9**	3.0***	3.2***	3.5***	3.3***
	3	3	0.75	1.1*	1.4**	1.3**	1.5***	1.5***	1.5***	1.4**
	1	5	fit	3.5**	5.6**	6.7**	6.7**	6.6**	6.7**	7.0**
	2	3	fit	2.1*	4.5**	6.0**	6.7**	7.2***	8.0***	7.4***
	3	3	fit	2.0*	3.1**	3.1**	3.6***	3.7***	3.6***	3.2**
	1	5	similarity	2.6*	4.3**	5.2**	5.3**	5.2**	5.1**	5.4**
	2	3	similarity	1.7°	3.3*	4.6**	4.9**	5.1***	5.6***	5.3**
	3	3	similarity	1.6°	2.2*	2.4**	2.7**	2.8**	2.7***	2.5**
$\Delta f_t^{(n)}$ and macro	1	64	-	-38.4°	-15.1°	-9.2°	-3.0**	0.7***	10.1***	-1.7°
	2	32	-	-30.1°	-8.5°	-2.3°	3.9***	5.8***	13.1***	4.1***
	3	32	-	-33.1°	-13.3°	-6.9°	1.0***	2.6***	10.6***	0.4***
	1	64	0.25	-14.9°	-0.5°	3.4**	7.4**	10.4***	17.1***	9.2***
	2	32	0.25	-9.9°	3.9**	8.1**	12.1***	14.2***	19.5***	13.4***
	3	32	0.25	-11.2°	1.4**	5.6**	10.9***	12.7***	18.6***	11.7***
	1	64	0.5	-0.7°	6.9**	9.2**	11.4**	13.5***	17.7***	13.2***
	2	32	0.5	1.9*	9.4**	12.0**	14.2***	16.0***	19.4***	15.8***
	3	32	0.5	1.6*	8.5**	10.9**	14.1***	15.6***	19.5***	15.4***
	1	64	0.75	4.2*	7.1**	8.0**	8.9**	10.0***	12.0***	10.1***
	2	32	0.75	5.2*	8.1**	9.3**	10.2***	11.2***	12.9***	11.3***
	3	32	0.75	5.4*	8.1**	9.1**	10.4***	11.4***	13.3***	11.5***
	1	64	fit	-13.9°	-0.4°	3.4**	7.4**	10.4***	16.8***	9.2***
	2	32	fit	-9.3°	3.8**	7.9**	11.8***	14.1***	19.3***	13.2***
	3	32	fit	-10.4°	1.6**	5.7***	10.9***	12.9***	18.7***	11.8***
	1	64	similarity	3.9*	10.7**	13.0**	14.9***	15.9***	19.0***	16.0***
	2	32	similarity	6.3*	12.7**	15.4**	17.4***	18.0***	20.2***	18.2***
	3	32	similarity	5.4**	11.5**	14.2***	16.9***	17.4***	20.0***	17.3***
s_t, $d_{t-12:t}^{(n)}$, $q_{t-12:t}$, $c_{t-12:t}$	1	5	-	-15.0°	-11.3°	-9.5°	-7.4°	-5.6°	-1.1°	-6.5°
	2	3	-	-9.9°	-5.1°	-2.9°	-0.6°	2.0*	6.9**	0.9°
	3	3	-	-4.1°	-0.7°	0.3°	1.6°	2.5°	4.0**	3.0°
	1	5	0.25	-9.4°	-6.8°	-5.2°	-3.6°	-2.0°	1.6°	-2.6°
	2	3	0.25	-6.4°	-2.9°	-1.1°	0.6°	2.7 *	6.5**	2.0°
	3	3	0.25	-2.5°	-0.2°	0.6°	1.5°	2.2°	3.4**	2.6°
	1	5	0.5	-5.0°	-3.4°	-2.2°	-1.1°	0.2°	2.7°	-0.3°
	2	3	0.5	-3.6°	-1.3°	0.0°	1.1°	2.6*	5.2**	2.1°
	3	3	0.5	-1.4°	0.1°	0.6°	1.2°	1.7°	2.5**	2.0°
	1	5	0.75	-1.8°	-1.1°	-0.4°	0.1°	0.8°	2.1°	0.6°
	2	3	0.75	-1.4°	-0.3°	0.3°	0.9°	1.7*	3.1**	1.5°
	3	3	0.75	-0.5°	0.2°	0.4°	0.7°	1.0°	1.4**	1.1°
	1	5	fit	-8.7°	-6.3°	-4.9°	-3.4°	-1.8°	1.6°	-2.5°
	2	3	fit	-6.3°	-3.0°	-1.3°	0.4°	2.4*	6.1**	1.6°
	3	3	fit	-2.6°	-0.5°	0.2°	1.1°	1.7°	2.8**	2.0°
	1	5	similarity	-5.2°	-3.8°	-2.4°	-1.5°	-0.2°	2.4°	-0.6°
	2	3	similarity	-5.0°	-2.6°	-1.3°	0.1°	1.6°	4.4**	1.0°
	3	3	similarity	-1.4°	0.1°	0.6°	1.2°	1.6°	2.4**	2.0°

Table 38: The out-of-sample R^2 (in %) of neural networks with varying numbers of nodes and layers. *, ** and *** indicate that the R^2 is significantly larger than 0 at 10, 5 or 1% significance level respectively, and ∘ indicates no significance.

Setting	Method	n=24	n=36	n=48	n=60	n=84	n=120	EW
$\Delta f_t^{(n)}$	Neural network (1 L - 5 N)	0.17	0.15	0.14	0.12	0.09	0.08	0.11
	Neural network (2 L - 3 N)	0.17	0.16	0.15	0.13	0.10	0.09	0.12
	Neural network (3 L - 3 N)	0.21	0.20	0.18	0.17	0.15	0.13	0.15
	Extreme tree	0.28	0.27	0.24	0.21	0.14	0.10	0.17
	Gradient-boosted tree	0.28	0.32	0.23	0.25	0.12	0.09	0.16
$\Delta f_t^{(n)}$ and macro	Neural network (1 L - 5 N)	0.19	0.18	0.17	0.15	0.14	0.14	0.16
	Neural network (2 L - 3 N)	0.21	0.19	0.19	0.17	0.15	0.14	0.17
	Neural network (3 L - 3 N)	0.20	0.19	0.18	0.17	0.15	0.14	0.16
	Extreme tree	0.60	0.54	0.52	0.49	0.45	0.45	0.49
	Gradient-boosted tree	0.57	0.51	0.51	0.48	0.47	0.47	0.54
s_t, $d_{t-12:t}^{(n)}$, $q_{t-12:t}$, $c_{t-12:t}$	Neural network (1 L - 5 N)	0.03	0.03	0.04	0.04	0.05	0.07	0.05
	Neural network (2 L - 3 N)	0.07	0.07	0.08	0.08	0.10	0.12	0.10
	Neural network (3 L - 3 N)	0.09	0.08	0.08	0.08	0.08	0.09	0.09
	Extreme tree	0.28	0.28	0.30	0.31	0.36	0.42	0.35
	Gradient-boosted tree	0.21	0.27	0.28	0.31	0.34	0.41	0.35

Table 39: The Information Ratios for timing strategies with no duration exposure on average, for extreme trees, gradient-boosted trees and alternative configurations of neural networks.

Setting	Method	n=24	n=36	n=48	n=60	n=84	n=120	EW
Benchmark	-	0.84	0.75	0.70	0.64	0.58	0.53	0.63
$\Delta f_t^{(n)}$	Neural network (1 L - 5 N)	0.87	0.78	0.73	0.66	0.58	0.53	0.64
	Neural network (2 L - 3 N)	0.86	0.77	0.72	0.65	0.58	0.53	0.64
	Neural network (3 L - 3 N)	0.86	0.76	0.71	0.65	0.59	0.54	0.64
	Extreme tree	0.84	0.75	0.70	0.62	0.55	0.50	0.61
	Gradient-boosted tree	0.87	0.78	0.73	0.66	0.57	0.51	0.63
$\Delta f_t^{(n)}$ and macro	Neural network (1 L - 5 N)	0.85	0.77	0.72	0.65	0.59	0.54	0.64
	Neural network (2 L - 3 N)	0.86	0.78	0.73	0.66	0.60	0.55	0.65
	Neural network (3 L - 3 N)	0.86	0.77	0.72	0.66	0.59	0.54	0.65
	Extreme tree	0.85	0.75	0.70	0.64	0.58	0.54	0.63
	Gradient-boosted tree	0.86	0.76	0.72	0.65	0.60	0.55	0.66
s_t, $d_{t-12:t}^{(n)}$, $q_{t-12:t}$, $c_{t-12:t}$	Neural network (1 L - 5 N)	0.72	0.64	0.61	0.56	0.52	0.51	0.57
	Neural network (2 L - 3 N)	0.78	0.71	0.67	0.61	0.57	0.55	0.62
	Neural network (3 L - 3 N)	0.80	0.72	0.67	0.62	0.57	0.53	0.62
	Extreme tree	0.70	0.62	0.59	0.56	0.54	0.54	0.57
	Gradient-boosted tree	0.62	0.58	0.54	0.52	0.50	0.52	0.54

Table 40: The Sharpe Ratios for timing strategies with average positive duration exposure for extreme trees, gradient-boosted trees and alternative configurations of neural networks.

C Computational details

C.1 Principal-component regressions

Principal components are computed using the PCA function of Scikit Learn.[37] The first differences in forward rates are not standardized before computing the covariance matrix. The rationale for this is that the forward rates themselves are of similar size, but the forward rates at the long end are more volatile. Thus, by standardizing the changes in the forward rates, some information would be lost. The data is also not standardized for the machine learning techniques. The macro variables on the other hand, are in widely different ranges and require standardization.

C.2 Tree methods

For random forests, extreme trees and gradient-boosted trees, we have made use of the functions by Scikit Learn.[38] Hyperparameters are chosen by a grid search over the maximum depth of the tree (2, 3 ,5 ,7 or 9), the number of trees (50, 100 or 200) and the number of features to consider at each split (2,3,5,7 or 10 in the first setting, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1 in the second setting, as a share of the total number of features, and and 1, 2, 3 or 4 in the third setting). The last 15% of the in sample data is used as validation sample, and the parameter values are chosen that minimize the Mean Squared Error over the validation sample. All other settings are set to their default value. The eleven observations preceding the validation sample are dropped to prevent overlap between the training and validation sample.

C.3 Neural networks

For neural networks, we have made use of the Keras[39] package, which is built on TensorFlow.[40] We have generally followed Bianchi et al. (2020b) in the neural network design, as tuning many settings was beyond our computing power. Vaying the network settings can have substantial impact on the results. However, the results reported in our work appear to be obtained with approximately optimal parameter settings, as any adjustments to hyperparameters that we have tried lead to lower R^2's. In no case have seen R^2's exceeding those obtained with regressions.

For an in-depth discussion of neural network settings, we refer to Bianchi et al. (2020b). We

[37] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[38] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html`, `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html` and `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html`.

[39] `https://keras.io/`

[40] `https://www.tensorflow.org/`

here simply state our approach for replication purposes. The numerical optimization is done using Stochastic Gradient Descent, with the learning rate set to 0.01 and Nesterov Momentum to 0.9. An Early Stopping procedure is used that stops optimization if the validation loss does not improve for 20 consecutive periods. In this case, the optimal weights are restored. Explanatory variables are scaled to be between -1 and 1. Kernel weights are initialized with the normal distribution. To combat overfitting, we make use of four techniques. Firstly, we fit each neural network 20 times, and take the average prediction of the 10 networks with the lowest validation errors.[41] Secondly, each batch is standardized before it is passed on to the final layer. Thirdly, after each layer, a fraction of the nodes is dropped out. Finally, L1L2 Regularization is applied to kernel weights.

Every 60 observations, the values of the dropout fraction (0.1, 0.3 or 0.5) and L1L2 regularization parameter are chosen to minimize the validation loss. Bianchi et al. (2020b) allow the regularization parameter to vary between 0.5 and 1 in the forwards-only setting, and between 0.001 and 0.1 in the forwards + macro setting. However, we do not have knowledge of the optimal regularization parameter ex ante. Therefore, we perform a grid search over [0.001, 0.01, 0.1, 1] in settings 1 and 3. This grid search leads to much lower R^2's in the forwards + macro setting.[42] We too, have therefore used a grid of [0.001, 0.01] for the regularization parameter in the second setting to obtain the results reported in this paper.

[41] Bianchi et al. (2020b) select 10 from 100 networks, but this is beyond our computing power. Regardless, our research indicated that selecting 10 works out of 20 or out of 100 does not appear to have much impact on the results.

[42] The R^2's, ascending in maturity, are [-26.9, -18.3, -15.0, -10.5, -5.8, 1.7, -8.7].