Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrics and Operations Research

Combining machine learning and dimension reduction techniques in forecasting inflation

Nienke Ybema (583094)

Supervisor:	E.P. O'Neill
Second assessor:	A. Pick
Date final version:	1st July 2024

Abstract

The prices of products and goods change constantly and have a direct impact on consumption and investment. Hence, the trend of prices, inflation, is a great determinant in the decision-making models of almost all economic agents and the predictions of inflation are essential in our economy. While early literature relied on linear models, most of the driving factors of inflation exhibit non-linear patterns. Therefore, recently more advanced methods like dimension reduction techniques or machine learning techniques were introduced in the forecasting environment. This paper aims to combine the strengths of these methods and trains several machine learning methods on the factors extracted from the data with dimension reduction techniques. To evaluate the contribution of the machine learning methods, we compare the forecasting performance with the predictive ability of a dynamic regression with shrinkage priors, which is employed with the obtained factors as covariates. The forecasts, made one-quarter ahead, were evaluated with different forecasting performance tests. For the estimation of these models we use the FRED-MD database. This is a recent and relatively complete dataset of monthly macro-economic variables measured in the US from 1959 until the present. The results of our research suggest that the inclusion of machine learning methods in factor-augmented models was not useful in the predicting of inflation. The machine learning methods provided less accurate results than the simpler AR(12) benchmark model. However, certain dimension reduction techniques, especially squared and quadratic principal component analysis enhanced the accuracy of the simpler benchmark model.

1 Introduction

The fluctuation of prices, known as inflation, is a concept that affects everyone and is a crucial indicator of the economic stability of a country or region. Inflation is a very volatile variable, and it is complex to accurately predict inflation. However, apart from the direct impact current inflation has on our daily lives, the expectations of inflation are also crucial in many economic forecasts used by institutions as the European Central Bank (ECB). Likewise, other economic agents, such as consumers and investors base their decision-making about investing or saving on their predictions of this variable. Therefore it is of great importance to form accurate predictions of inflation to implement appropriate monetary policies.

While simple univariate models already predict inflation quite accurately and are difficult to beat (Stock and Watson, 2006), complex machine learning methods and dimension reduction techniques can possibly assess the non-linearity in the macro-economic environment even better. Dimension reduction techniques, which transform variables into a smaller number of factors, can be linear and relatively simple or highly non-linear and complex. Machine learning methods are complex computer algorithms which can improve the predictions relative to ordinary least squares in complex or non-linear forecasting frameworks.

The forecasting of inflation using machine learning methods or dimension reduction techniques is widely researched. Both methods seem to improve the simple benchmark AR(p) model (Medeiros et al., 2018), (Hauzenberger et al., 2023). Therefore this paper aims to combine the strong features of both techniques and thus apply machine learning methods on the factors obtained with dimension reduction techniques. In this way we investigate whether adding another non-linear model instead of a simple regression improves the results of these factor-augmented

forecasts. Hence the main research question of this paper is: *How can machine learning methods improve the accuracy of real-time quarterly ahead forecasts of inflation in combination with dimension reduction techniques?*

We try to answer this question by first analyzing the performance of models which implement the different dimension reduction techniques with a dynamic regression using shrinkage priors. Thereafter three different machine learning methods are employed on the factors obtained with the dimension reduction techniques.

This paper is built on the paper of Hauzenberger et al. (2023), which researched the predictive performance of dimension reduction techniques applied to the forecasting of inflation. This paper extends the work of Hauzenberger et al. (2023) by forecasting inflation using machine learning methods (elastic net, random forest and kernel ridge regression) trained on the reduced dimension data instead of applying linear models to these factors and thus studying the added value of combining these two methods. The first part of the analysis is limited to the performance of the dimension reduction techniques. In this part some of the models that were best performing in the paper of Hauzenberger et al. (2023) are applied and compared with simpler benchmark models, which include an AR(p) and ARX model. The covariates of the ARX model are set to be the 5 variables which are most correlated with the factors obtained through the best performing dimension reduction technique. The factors obtained with the factor-augmented methods are then linked to inflation in a second stage regression and the results of the paper of Hauzenberger et al. (2023) are verified.

The coefficients estimated with the dynamic regression are assumed to be constant over time. To shrink the numbers of parameters, two different shrinkage methods are employed, particularly the Minnesota prior (Chan, 2019) and the Horsehoe prior (Carvalho et al., 2010). The specification of the model allows for stochastic volatility. In the second part of the analysis the performance of the different machine learning methods in the framework of inflation forecasting is analysed. Selection of the models employed is based on the performance of these models in previous literature. Different sets of combinations including machine learning methods and factor models are made to examine their contribution to each other.

To assess the performance of the models, different evaluation metrics and tests are employed. The root mean square error (RMSE) and mean absolute error (MAE) are calculated and to determine whether the results of a model are significantly different from the AR(p) benchmark model a Diebold-Mariano test is performed. Additionally, the model confidence set is constructed to establish the set of best performing models based on a certain confidence level.

The FRED-MD database is a large dataset including many macro-economic variables (McCracken and Ng, 2015), this database is used for the estimation of our models. A rolling window of 20 years is estimated monthly, while the estimation sample remains constant at 240 observations. The machine learning methods and dimension reduction techniques are valuable in forecasting with such large datasets. Therefore the combination of these two different kinds of methods applied to a big dataset can increase the predictive ability of the model even more.

We evaluate three machine learning methods and four dimension reduction techniques on this data and conclude that the more complex models including machine learning methods and dimension reduction techniques were not a significant improvement of the AR(p) benchmark

model. The forecasting performance of most of these combinations was significantly worse than the performance of the dimension reduction techniques combined with a dynamic regression and a Bayesian prior. Therefore including machine learning methods when predicting inflation with factor augmented estimation is not beneficial. Squared an quadratic principal component analysis provide smaller values for RMSE and MAE than the benchmark model. However, these differences are not significant according to the Diebold-Mariano test or the model confidence set procedure.

In conclusion, the inclusion of these particular machine learning methods in the framework of dimension reduction techniques in forecasting inflation results in unnecessary complexity of the models. Thus, it would be preferred for economic agents to avoid combining these models in the forecasting of inflation one-quarter ahead. However, the use of only dimension reduction techniques could be valuable by incorporating the non-linearity's of the data in the model.

In this paper, we first provide a general overview of the existing literature and relevance in Section 2. Thereafter we describe the FRED-MD dataset and the variables used in our models in Section 3. In section 4 we explain the methods employed. Then the results are presented in section 5. Finally, we state our conclusions in section 6.

2 Existing literature

2.1 Dimension reduction techniques

Various dimension reduction techniques have been tested in a macro econometric forecasting environment. Most of the literature came to the conclusion that models which apply dimension reduction techniques generally outperform simpler models in forecasting inflation. Hauzenberger et al. (2023) researched the autoencoder, different forms of principal component analysis (PCA), diffusion maps, local linear embeddings and isometric feature mapping. The autoencoder was the best overall predictor in monthly ahead forecasts, but squared and quadratic PCA provided the most accurate predictions in quarterly ahead forecasts. Additionally, this paper constructed a dynamic combination of the best performing forecasting methods. They found that the dynamic combination performed slightly worse than the best performing model. However, this model reduced much uncertainty in turbulent times.

2.2 Machine learning methods

Another interesting way of forecasting inflation is employed by utilizing machine learning methods. Machine learning methods impose a more flexible structure than traditional econometric methods and do not need a lot of assumptions. Therefore, these methods are capable of capturing complex macroeconomic relationships. Goulet Coulombe et al. (2022) employed machine learning methods in different forecasting environments and evaluated that these methods are particularly useful when there is a high level of uncertainty. Machine learning methods are able to capture the non-linearity's arising from this uncertainty. The literature is quite consistent in the belief that most machine learning methods are an improvement on simpler regression methods. Nonetheless, there is no clear superior method.

The random forest algorithm, introduced by Breiman (2001), uses multiple decision trees to form a model and make predictions. This method was found to be the best performing method in predicting the inflation in the USA by Medeiros et al. (2018) and Goulet et al. (2021) for a time horizon of 3, 6 and 12 months. The random forest model also provided accurate predictions for inflation over longer time horizons of one year in Brazil (Araujo and Gaglianone, 2023), while for shorter horizons a combination of the best performing models worked the best. Other Tree-based models, like the extreme gradient boosting model (XGBoost) showed good performance as well (Li et al., 2022). Because of the excellent performance in previous literature, this paper uses the random forest algorithm as one of the machine learning methods employed to estimate our model. This model can handle non-linearity and is not restricted to assumptions, which allows the model to be quite flexible.

During the COVID pandemic, the forecasting of inflation became even more complex, because of the unpredictability of the macroeconomic environment. However, the kernel ridge regression still produced accurate predictions. Goulet Coulombe et al. (2022) tested the performance of the kernel ridge regression. This method is quite accurate in predictions in turbulent times. The kernel ridge regression handle high-dimensionality and non-linearity in the data. It prevents over-fitting by penalizing the parameters. These properties demonstrate that the kernel ridge regression provides an attractive framework for estimating nonlinearity's in the data. Thus, we choose the kernel ridge regression and the random forest algorithm as methods to estimate our factor-augmented model. In addition, we implement the elastic net method. This method is close to linearity and thus we can compare the highly non-linear machine learning methods with a more linear machine learning method.

2.3 Feature selection for machine learning methods

Goulet et al. (2021) researched the effect of different data transformations before applying machine learning methods in inflation forecasting. Such transformations can alter the results of the models drastically. Therefore one should be careful with using these transformation methods. Most machine learning methods generate accurate predictions using high-dimensional data. However, in forecasting we normally do not have many data points available and the issue of over-fitting arises. Therefore the simple principle component analysis is often used to prevent these over-fitting issues (Goulet Coulombe et al., 2022). Uddin et al. (2018) identifies the selection of the features used in the machine learning methods as the key to better prediction accuracy of machine learning methods and shows that the selection of specific feature sets can optimize the machine learning process. Goulet et al. (2021) extract factors from the data using MARX and MAF methods and the simple principal component analysis. They found that the inclusion of these dimension reduction techniques yield lower RMSEs and is beneficial for the forecasting accuracy. Even the simple principle component analysis provided more accurate results than the benchmark model. Because of these results we are interested in further researching different dimension reduction techniques in combination with machine learning methods. Hence, this paper applies other dimension reduction techniques (squared and quadratic PCA and ISOMAP) which are frequently used in previous literature to the machine learning forecasting framework.

In conclusion, most of the existing literature considers the performance of both machine

learning methods and dimension reduction techniques as a good addition in the forecasting of inflation. However, the combination of the two methods is rarely researched. This paper contributes to the literature by using a big and recent dataset and applying the best performing machine learning methods to a set of factors created by dimension reduction techniques in the forecasting procedure. Therefore it combines the strengths of already existing models in creating a new forecasting model. In addition we compare the predictive ability of these models directly to forecasting performance of dimension reduction techniques combined with a simpler dynamic regression.

3 Data

3.1 Data source

The data source used for the estimation of the models is the FRED-MD database. This data is publicly available and covers macroeconomic variables collected at a monthly frequency in the U.S. A detailed description of the data can be found in McCracken and Ng (2015). In our paper data from 1963 until 2020 is used. The dataset contains 134 indicator variables of the U.S. economy and is updated monthly. The dataset is designed for big data analyses and includes 8 different groups of variables: Output and income, labor market, housing, consumption, money and credit, interest and exchange rates, prices and stock market. Because not all variables are consistent over the years we use a subset of 105 monthly variables. A list of all the included variables can be found in Appendix Section 3.4.

3.2 Data transformations

Stationary data is more convenient for estimation. Therefore we transform some of the variables in the dataset to cause them to be stationary. The transformations include taking first differences, logarithms, or first and second differences of logarithms and calculating the percentage change over one period. The exact data transformations per variable are described in Appendix Section 3.4. Additionally, the covariates are standardized to have a mean of zero, and a variance of one. Some variables have missing values for several months. These missing observations are handled by setting the value of these observations as the previous value of this variable.

3.3 Rolling window

For our estimation, we use data from January 1963 until December 2019. A rolling window is adopted, the length of the estimation sample stays fixed at 240 observations. The estimation sample is used for the model estimation and the forecasting of inflation in the hold-out sample, which ranges until December 2019. We use the first month of our hold-out sample to evaluate the predictions. One-quarter-ahead predictions are made for the periods ranging from January 2000 until December 2019. We estimate the quarterly-ahead inflation for all months within this period, while using the latest dataset available at time t. Thus, the model is re-estimated each month to generate forecasts. The predictions are evaluated and compared with the actual inflation numbers. The actual inflation numbers of time $t + h$ are gathered 3 months after time

$t + h$. Hence, we make sure that the realized inflation is not subject to revision. Most data revisions take place in the first months after t (Croushore, 2011), so this leads to the most realistic outcome.

3.4 Dependent variable

Our dependent variable, inflation, is specified using the Consumer Price Index (CPI) variable in the dataset. We estimate the h-period inflation rate, which is calculated by taking the logarithm of the change in CPI in the period ranging from t until $t + h$. This variable indicates the percentage change in price over the h months after t. We subtract the inflation rate between the period of $t - 1$ until t from the h-period inflation rate. This method, based on Stock and Watson (1999), causes the dependent variable to have an I(1) process. Thus, the dependent variable y_{t+h} is specified as

$$y_{t+h} = log\left(\frac{CPI_{t+h}}{CPI_t}\right) - log\left(\frac{CPI_t}{CPI_{t-1}}\right) \tag{1}$$

In Table 1, the descriptive statistics of the estimated variable y_{t+h}, the h-period ahead inflation, and CPI are shown for the period from 1980 until 2019. y_{t+h}, the variable we predict, fluctuates around its mean, which is close to zero. This indicates that y_{t+h} is relatively stationary, without a constant trend. The inflation rate itself ranges from 0 to 0.10. This is to be expected, because the inflation rate is normally between 0 and 10 percent. Notably, inflation tends to be lower in later years. The standard deviation is large, implying that inflation is a volatile variable. The values of the price level increase over time due to positive inflation. The standard deviation of CPI over the whole period is larger than the standard deviation in specific periods. This is because of the larger sample size and the largely significant differences in prices between 1980 and 2019.

Table 1: Descriptive statistics of the dependent variable, the inflation rate and the Consumer Price Index (CPI)

	y_{t+h}		Inflation rate		CPI	
	Mean	**S.D.**	**Mean**	**S.D.**	**Mean**	**S.D.**
1980-1984	-3.59 e-03	3.56 e-02	0.061	0.038	95.77	7.46
1985-1989	1.69 e-03	2.72 e-02	0.037	0.020	115.57	6.50
1990-1994	-1.12 e-03	2.16 e-02	0.033	0.015	140.90	6.34
1995-1999	5.91 e-04	1.67 e-02	0.024	0.010	160.68	5.21
2000-2004	-4.43 e-04	3.19 e-02	0.025	0.015	181.34	6.09
2005-2009	-1.80 e-04	6.25 e-02	0.025	0.041	207.80	7.79
2010-2014	-8.74 e-04	2.65 e-02	0.016	0.019	229.15	6.43
2015-2019	1.40 e-03	2.49 e-02	0.018	0.013	246.76	7.28
Overall	-6.20 e-04	3.34 e -02	0.030	0.027	172.25	50.98

Note: $y_{t+h} = log(CPI_{t+h}/CPI_t) - log(CPI_t/CPI_{t-1})$; Inflation rate $= log(CPI_{t+h}/CPI_t)$

4 Methodology

The forecasting of inflation is split in two parts. First, we apply dimension reduction techniques to extract the most important factors in the data. These obtained factors are used as independent variables in the next step. In this step a dynamic regression model with Bayesian priors is applied to estimate the model. Additionally, three different machine learning methods (elastic net, random forest and kernel ridge regression) are employed with the obtained factors as covariates. The forecasting accuracy of these methods is compared with the forecasting performance of the dynamic regression methods. We forecast inflation one quarter ahead and use four different dimension reduction techniques, particularly linear, squared and quadratic principle components analysis and isometric feature mapping. Additionally, two benchmark models are estimated. In particular, an AR(12) model and an ARX model, with 5 exogenous variables. These variables are selected based on the outcomes of the best performing dimension reduction technique. The variables which are most correlated with the obtained factors belonging to this technique are chosen as regressors.

Our inflation variable is constructed from the consumer price index (CPI), as specified in Section 3. We are interested in quarterly ahead forecasting, thus $h = 3$.

There are 104 different covariates from the FRED-MD database included in each model. Additionally, 12 lags of each of these variables are used as additional covariates in the construction of the factors. Furthermore, the 12 lags of CPI itself are included as additional regressors in the dynamic regressions and machine learning methods. As mentioned in the Section 3 a rolling window is applied and the model is re-estimated for each t, to predict inflation at time $t + h$ using the latest available information.

4.1 Dimension reduction techniques

The dataset of FRED-MD is relatively big, therefore including all variables could cause the number of regressors (K) to be bigger than the number of observations (T). This could lead to overfitting issues. That is why dimension reduction techniques are introduced. The basic idea of this techniques is to use a function f that transforms regressor matrix $X = (x_1, ..., x_T)'$ to a lower-dimensional $T \times q$ matrix representation $Z = f(X) = (z_1, ..., z_T)'$. f reaches from linear to highly non-linear depending on the specific technique. q has to be smaller than K and represents the number of factors included in the model. The value of q is crucial for the predictive ability of a model. Therefore we consider different values for q, and retrieve 5, 15 and 30 factors for each model.

4.1.1 Linear Principal Component Analysis

Linear principal component analysis (PCA) is the simplest dimension reduction technique presented in this paper. PCA alters the correlated variables into a set of uncorrelated variables, called factors. These factors explain most of the variance. PCA alters the magnitude of the covariates with function $g : W = g(X)$ and it alters the sample covariance with function $h : k = h(W'W)$. The relation between the PCs and X is as follows:

$$Z = f(X) = g(X)\Lambda(k) = W\Lambda(k) \tag{2}$$

Where $\Lambda(k)$ the truncated Kxq eigenvector matrix of k. For the linear PC model $W = X$ and $k = X'X$. Hence, Linear PCA takes the eigenvectors and orders them based on their eigenvalue. The first principal component has the largest eigenvalue and explains the most variance in the data. We choose the first q eigenvectors as factors in our model. These factors are consistent when K and T go to infinity. The downside of this model is that it assumes linearity.

4.1.2 Squared and Quadratic Principle Component Analysis

To overcome the linearity restriction an quadratic link could be applied to function f. This results in a more flexible structure. Squared and Quadratic PCA are similar to linear PCA. The difference is in the specification of the functions f and h. Squared PCs are represented by $W = X^2$ and $k = (X^2)'(X^2)$. Where X^2 denotes the element-wise multiplication of X. Quadratic PCs are represented by $W = (X, X^2)$ and $k = W'W$. We construct the factors in a similar manner as for the linear PCA. Here, the factors are equal to the first q eigenvectors of the corresponding matrix W. The quadratic transformations in the establishment of W cause strong movements in the covariates to dominate the effects on the forecasts, while smaller movements have little effect. This implies that crisis periods with irregular observations have great impact on the predictions. Squared and quadratic PCAs were the best performing dimension reduction techniques for one-quarter-ahead forecasts in the paper of Hauzenberger et al. (2023).

4.1.3 Isometric Feature mapping

Isometric Feature mapping (ISOMAP) was introduced by Tenenbaum et al. (2000). ISOMAP calculates the geodesic distances between data points and uses them to retrieve a small number of underlying factors. In the first step the geodesic distance between all data points is measured. Thereafter we identify the nearest neighbours of each datapoint. Then the shortest path distances between all data points are stored in a dissimilarity matrix D where $D_{ij} = \sum_k |x_{ki} - x_{kj}|$ is the distance between points i and j. Multidimensional Scaling is applied to the matrix. This method transforms the distances to a lower dimensional space, such that the data points are preserved as much as possible. We determine the eigenvectors of the centered distance matrix to get the low dimensional coordinates. The centered distance matrix B is computed as follows:

$$B = -\frac{1}{2} H D^2 H \tag{3}$$

$$H = I - \frac{1}{n} \iota\iota^T \tag{4}$$

Where I represents the identity matrix and ι is a vector of ones. Because the transformation used is non-linear, it is possible to identify a non-linear structure with this dimension reduction technique.

4.2 Linear model

After applying the dimension reduction techniques the following regression model is estimated:

$$y_{t+h} = d_t'\beta + \epsilon_{t+h}, \qquad \epsilon_{t+h} \sim \mathcal{N}(0, \sigma^2_{t+h}) \tag{5}$$

Here, β is a $M(= q + p + 1) \times 1$ vector of parameters, associated with d_t, which includes an intercept, q factors from one of the dimension reduction techniques and p lagged values of y_t. To further optimize and shrink the parameters two different Bayesian priors are adapted, namely the Horsehoe prior (Carvalho et al., 2010) and the Adaptive Minnesota prior (Chan, 2019). The parameters are assumed to be time-invariant. The process of the Bayesian prior estimation is discussed in detail in Appendix Section A. σ^2_{t+h} denotes the time varying variance, which follows the stochastic volatility model developed by Kastner and Frühwirth-Schnatter (2014). This means $log(\sigma^2_{t+h})$ follows a normal distribution:

$$\log(\sigma^2_{t+h}) \sim \mathcal{N}(\mu + \phi \log \sigma^2_{t+h-1}, \nu^2) \tag{6}$$

where μ, ϕ and ν are the hyperparameters drawn from the stochastic volatility priors. These are drawn in each MCMC iteration using the estimated beta from the Minnesota or Horseshoe prior and the latent values of the parameters and $log(\sigma^2_{t+h})$. The posterior distribution of the parameters is explained in more detail in Appendix Section A.2.1. Forecasts are made based on the estimated β and σ_{t+h} from the previous iteration and saved for the last 8000 MCMC iterations. The estimation and forecasting of these regression models is performed for each rolling window.

4.3 Machine learning methods

Additional to the dynamic regression methods, three machine learning methods are evaluated to examine whether they could improve the predictive ability of the forecasting model even more. The factors obtained from the dimension reduction techniques are used as independent variables in estimating these models. In accordance with the other methods, these models are estimated for each rolling window and each month forecasts are made.

4.3.1 Elastic Net

Elastic Net is a regularization technique introduced by Zou and Hastie (2005). This method takes the basic form of a linear regression and adds a penalty to the magnitude of the parameters in the loss function to shrink the parameters towards zero. The elastic net loss function combines the penalties of the Ridge and Lasso methods. The penalty of Lasso is equal to the absolute value of the magnitude of coefficients: $\lambda \sum_{j=1}^{p} |\beta_j|$. This leads to models for which some coefficient estimates are exactly zero. The penalty of Ridge is equal to $\lambda \sum_{j=1}^{p} \beta_j^2$. Coefficients of Ridge are generally not shrunk to zero. Elastic net minimizes the following loss function

$$Loss = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \tag{7}$$

Where λ_1 and λ_2 are tuning parameters. We tune these parameters in each rolling window using 5-fold cross validation. The elastic net estimator of β is equal to vector of parameters that minimizes this loss function. The elastic net model can overcome over-fitting due to the penalization of the parameters and is a flexible model due to the tuning parameters. However, this machine learning method is not optimal for capturing non-linear relationships.

4.3.2 Random Forest

The random forest algorithm, introduced by Breiman (2001), constructs multiple decision trees during training and delivers the mean prediction of the individual trees as forecasts. It handles classification or regression models. The decision trees splits the data into subsets at each decision node based on the values of some of the exogenous variables. The specific variables at each decision node are determined randomly. The training dataset (X, Y) is used to construct a function $m(x) = E[Y|X = x]$. The random forest algorithm uses bagging: It creates multiple subsets of the training data, which are used to train different decision trees. At each decision node in the tree a random subset of features is adopted as potential splitting variables. Therefore the different trees have reduced correlation. Each branch in the tree represents a specific outcome of the decision at the corresponding note. The forecasts are calculated as the region-specific mean of y_{t+h} at the node corresponding to the specific data point used to predict y_{t+h}. The forecast is averaged over all the constructed trees in a specific rolling window. In this paper 500 trees are constructed for each model estimation, which should be enough to stabilise predictions. The random forest model is adopted because of its predictive accuracy and flexible structure, where no assumptions are needed. Moreover, it can capture non-linearities in the dataset.

4.3.3 Kernel Ridge regression

The kernel ridge regression (KRR) extends the regular ridge regression with a non-linear Kernel function and thus it can capture non-linear relationships in the data. KRR uses the Kernel trick which was introduced by Boser et al. (1992). The Kernel trick maps the inputs in a higher dimensional space without computing all the exact outcomes. Instead, the Kernel trick directly computes the inner product. This makes it possible to map our data in a high dimensional space, while keeping the computation time small. The kernel trick is represented by the fact that there exist a reproducing kernel $K()$ such that

$$\hat{E}(y_{t+h}|Z_t) = \sum_{i=1}^{t} \hat{\beta}_i \langle \phi(Z_i), \phi(Z_t) \rangle = \sum_{i=1}^{t} \hat{\alpha}_i K(Z_i, Z_t) \tag{8}$$

$$K_\gamma(x, x') = exp\left(-\frac{\|x - x'\|^2}{2\gamma^2}\right) \tag{9}$$

We use the standard radial basis function (RBF) kernel with tuning parameter γ as function $K()$. The loss function of the kernel ridge regression resembles the loss function of the normal

ridge regression. This function is minimized to obtain the optimal estimates of β

$$Loss = \sum_{i=1}^{n}(y_i - \sum_{j=1}^{n}\beta_j K(x_i, x_j))^2 + \lambda \sum_{i=1}^{n}\alpha_i^2 \tag{10}$$

γ and λ are tuning parameters. These two parameters are tuned by minimizing the Akaike information criterion (AIC). For the evaluation of the tuning parameters the forecasting data set is split into a training and evaluating data set of respectively 80 and 20 percent of the data. The AIC of the evaluation dataset is calculated for 7 values of γ and λ ranging from 10^{-3} until 10^3. We choose γ and λ based on the lowest value of AIC. Then the loss function is minimized for these values of γ and λ. The forecast obtained for the KRR is as follows:

$$\hat{E}(y_{t+h}|Z_t) = K_\gamma(Z_t, Z)(K_\gamma(Z_t, Z) + \lambda I_T)^{-1} y_t \tag{11}$$

Because of the non-linear kernel function, this method is useful in capturing non-linear relationships in the data and is flexible in the tuning of the parameters.

4.4 Forecasting and testing

4.4.1 Evaluation metrics

To generate forecasts, y_{t+h} is predicted in each rolling window utilizing X_t as a matrix of regressors. X_t includes the factors extracted from all regressors and their lags obtained at time t. For the evaluation of point forecasts root mean squared errors (RMSEs) and mean absolute errors (MAEs) are obtained for each model. Both metrics use the forecast errors of all rolling windows and measure their magnitude. The RMSEs are more affected by outliers than the MAEs, because of the squared errors in the function of RMSE. The RMSE and MAE are calculated as follows over all rolling windows:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_{t+h} - \hat{y}_{t+h})^2} \tag{12}$$

$$MAE = \frac{1}{T}\sum_{t=1}^{T}|y_{t+h} - \hat{y}_{t+h}| \tag{13}$$

Where y_{t+h} is the actual value of our dependent variable and $\hat{y}_{t+h}$ is the predicted value of y_{t+h} using the information available at time t. For the dynamic regressions using Bayesian optimization the $\hat{y}_{t+h}$ is the expected value of y_{t+h} averaged over all MCMC iterations.

4.4.2 Forecasting performance tests

Additionally, two performance tests are applied: The Diebold-Mariano (DM) test (Diebold and Mariano, 1994) and the model confidence set (MCS) procedure (Hansen et al., 2011). The Diebold-Mariano test computes a loss function using the errors of the tested model and the errors of the benchmark AR(12) model. The test consists of a simple t-test to test the null hypothesis which is equal to no difference in the forecast errors of the competing models. In this

paper two different loss functions are employed to test the significance of the particular metrics (RMSE, MAE). The loss functions and null hypotheses are equal to:

$$L(e_{it}) = e_{it}^2 \tag{14}$$

$$L(e_{it}) = |e_{it}| \tag{15}$$

$$H_0 : E(g(e_{it})) = E(g(e_{0t})) \tag{16}$$

Where e_{it} is the forecast error of model i computed at time t and model 0 represents the AR(12) benchmark model. Equation 14 tests the significance of the difference in the RMSEs of the tested model and AR(12) benchmark model and Equation 15 determines the significance of difference in the MAEs.

The model confidence set procedure is a sequence of tests that determines a set of superior models. Within this set of models the null hypothesis of equal predictive ability (EPA) is not rejected at a specified confidence level α. This procedure is performed for $\alpha = 0.05$ and $\alpha = 0.1$. $l_{i,t}$ is the loss function for model i at time t, this function is equal to the square of the errors as in Equation 14. $d_{ij,t} = l_{i,t} - l_{j,t}$ is the loss differential between two models. Thus the EPA null hypothesis is equal to:

$$E(d_{ij,t}) = 0, \quad \forall i, j \tag{17}$$

$$t_{ij} = \frac{\overline{d}_{ij}}{\sqrt{\widehat{Var}\left(\overline{d}_{ij}\right)}} \tag{18}$$

Where t_{ij} represent the t-statistics and $\overline{d}_{ij} = \frac{1}{m}\sum_{t=1}^{T} d_{ij,t}$ is equal to the relative sample loss between the ith and jth model. We perform a block bootstrap procedure with 5000 bootstrap samples. $\widehat{Var}\left(\overline{d}_{ij}\right)$ denotes the bootstrap estimate of $Var\left(\overline{d}_{ij}\right)$. The test statistic for each model is equal to:

$$T_{R,M} = \max_{i,j \in M} |t_{ij}| \tag{19}$$

The model with the highest test statistic is eliminated from the model confidence set if the test statistic is significant according to confidence level α. The elimination process is repeated until all models satisfy the null hypothesis of equal predictive ability at confidence level α.

5 Results

In this section we elaborate on the outcomes of our forecasting models. We used two different evaluation metrics to compare the forecasting performance of the different models: The mean absolute error (MAE) and the root mean squared error (RMSE). Lower values of these metrics relate to better forecasting performances of the corresponding model. Because of the square in the formula of RMSE, large errors have more influence on this metric and it is more sensitive to large outliers than the MAE. The values of these metrics are calculated relative to the RMSE

and MAE of the benchmark AR(12) Minnesota model.

5.1 Root Mean Squared Errors

The results of the relative RMSEs are presented in Table 2. The asterisks represent the significance of the difference in RMSE between the tested model and the benchmark model according to the Diebold-Mariano test. The T-statistics and p-values of the Diebold-Mariano test for all RMSE values are presented in Appendix Section C

5.1.1 The performance of the machine learning methods

None of the machine learning models yield significantly better results than the benchmark model. On the contrary, the relative RMSEs of these methods are all below zero, which indicates these methods performed worse than the benchmark model. Especially the random forest method shows poor forecasting performance. The RMSEs of the random forest method are all significantly higher than the RMSE of the benchmark model. While most of the RMSEs of elastic net and the kernel ridge regression are not significantly worse than the benchmark AR(12) model, these models yield undesirable results in terms of predictive accuracy. Therefore, these models do not improve the linear AR(12) model combined with the Minnesota prior, and none of the machine learning methods can outperform the benchmark model according to the RMSE.

5.1.2 The performance of dimension reduction techniques

Thereafter we compare the factor-augmented methods estimated with the Bayesian priors. There is not a lot of significance in the differences of the specific models with the benchmark model. The RMSE values are the lowest for the squared and quadratic principle component analysis with five factors for both the Minnesota and Horseshoe prior. Hence, these dimension reduction techniques generate the most accurate forecasts. This is in line with the results of Hauzenberger et al. (2023). All other models provide a relative RMSE above one which indicates these models do not improve the benchmark model. Additionally, the two best performing factor models (the squared and quadratic PCA with $q = 5$) are used to create an ARX model with independent variables, which are most correlated with the obtained factors from squared or quadratic PCA. These models do not improve the benchmark model according to their higher values of RMSE. Thus, when we examine the RMSE, the squared and quadratic PCA are considered the best-performing dimension reduction techniques. However, there is no statistical evidence that these models have a better predictive accuracy than the AR(12) benchmark model. This lack of significance could be due to the small sample size or noisy data. Because of this insignificance we are not able to reject the null hypothesis that the factor-augmented models have equal predictive ability as the benchmark AR(12) model.

5.2 Mean Absolute Error

The MAEs of the different specifications are presented in Table 5.2.2, where the asterisks represent their significance. The T-statistics and p-values of the Diebold-Mariano test are presented in

Table 2: Values of the relative Root Mean Squared Error (RMSE) for all specifications calculated over an estimation sample from 2000 until 2019

Specification	**Minnesota**	**Horseshoe**	**Elastic Net**	**Random Forest**	**Kernel Ridge**
AR (p)	1.053	0.995'	1.016	1.196*	1.077
PCA linear (q = 5)	1.044	1.049	1.153	1.212*	1.114
PCA linear (q = 15)	1.031	1.025	1.208	1.214*	1.170*
PCA linear (q = 30)	1.026	1.023	1.179	1.221*	1.227*
PCA squared (q = 5)	0.966	0.925	1.578	1.204*	1.130
PCA squared (q = 15)	1.012	0.990	1.140	1.221*	1.175'
PCA squared (q = 30)	1.033*	1.036	1.264'	1.231**	1.182'
PCA quadratic (q = 5)	0.980	0.951	1.484	1.209*	1.128
PCA quadratic (q = 15)	1.036	1.036	1.109	1.206*	1.169'
PCA quadratic (q = 30)	1.023	1.066	1.242*	1.225*	1.200'
ISOMAP (q = 5)	1.006	1.007	1.055	1.238*	1.109
ISOMAP (q = 15)	1.008	1.014	1.053	1.246*	1.155'
ISOMAP (q = 30)	1.001	0.994	1.093*	1.219*	1.185*
ARX (PCAs (q = 5))	1.033	1.046	1.216	1.210*	1.119
ARX (PCAq (q = 5))	1.028	1.030	1.213	1.213*	1.141

Note: The RMSE is calculated over the whole sample from 2000 until 2019; the first model (AR(p), Minnesota) is the benchmark model, the rest of the values of RMSE are relative to this model; q is the number of factors used for estimation; ': $p < 0.01$ *: $p < 0.05$, **: $p < 0.001$ indicate the statistical significance for each model relative to the benchmark.

Appendix Section C as well. The collected MAEs confirm most of the conclusions of the RMSE evaluation.

5.2.1 The performance of the machine learning methods

The machine learning methods, random forest and kernel ridge regression, again provide significantly higher values for MAE relative to the benchmark model. Thus, there is statistical evidence that the two most complex machine learning methods, the kernel ridge regression and the random forest algorithm can not improve the simple AR(12) model. The MAE values of elastic net are all above one. Thus, according to the RMSE and MAE there is no additional value in including the machine learning methods in the factor-augmented model for the forecasting of inflation. This result contradicts the remarkable performance of machine learning methods in the literature on forecasting. This could be the result of the way the data is transformed. The manner in which the data is transformed is crucial for the predictive accuracy of the machine learning methods. The unexpected results could also be due to the great performance of the dynamic regression methods in combination with dimension reduction techniques. Most machine learning methods handle high-dimensionality in the data and prevent over-fitting by themselves. Hence, imposing an additional restriction may cause the methods to underperform and reduces their flexibility.

The differences in MAEs of the machine learning methods and the benchmark model are more significant than the differences in RMSE, especially for the kernel ridge regression. RMSE

is more sensitive to outliers. Thus, this confirms that machine learning methods have relatively better predictive ability during periods of high volatility and uncertainty and are strong in coping with outliers in the data. This is especially true for the kernel ridge regression. The RMSEs of this method are only significantly different for some of the dimension reduction techniques, while the MAEs of this method are significantly different from the benchmark model for all of the dimension reduction techniques employed.

5.2.2 The performance of dimension reduction techniques

The differences in the MAEs of the dimension reduction techniques are similar to the differences in RMSE. The only models with smaller MAEs than the benchmark model are the PCA squared and AR(12) model estimated with the Horseshoe prior. Most of the more complex factor models have lower relative RMSEs compared to their relative MAEs. This implies that these methods are an improvement of the simple AR(12) model when there are large outliers or the data is more volatile. This corresponds with the nature of squared and quadratic PCA to respond heavily to large outliers. Most of the differences in MAE between the dimension reduction methods and the benchmark model are not significant. The ARX model is estimated for the same models (PCA squared and PCA quadratic). These models again do not show statistical significance. From the values of RMSE and MAE we can not draw a secure conclusion on a superior model. However, both metrics demonstrate good forecasting performance for the squared and quadratic PCA with five factors. Thus, the use of these dimension reduction techniques can be beneficial, certainly in turbulent times.

Table 3: Values of the relative Mean Absolute Error for all specifications calculated over an estimation sample from 2000 until 2019

Specification	**Minnesota**	**Horseshoe**	**Elastic Net**	**Random Forest**	**Kernel Ridge**
AR (p)	0.710	0.992*	1.020	1.283**	1.176*
PCA linear (q = 5)	1.017	1.019	1.112	1.299**	1.203**
PCA linear (q = 15)	1.017	1.019	1.164'	1.303**	1.255**
PCA linear (q = 30)	1.009	1.035	1.198*	1.331**	1.334**
PCA squared (q = 5)	1.011	0.992	1.176	1.273**	1.197**
PCA squared (q = 15)	1.018	1.015	1.133	1.306**	1.234**
PCA squared (q = 30)	1.044*	1.046	1.223**	1.324**	1.218**
PCA quadratic (q = 5)	1.016	1.000	1.063	1.302**	1.179*
PCA quadratic (q = 15)	1.069	1.072	1.102	1.297**	1.253**
PCA quadratic (q = 30)	1.033	1.080	1.194*	1.306**	1.254**
ISOMAP (q = 5)	1.008	1.014	1.066*	1.326**	1.194**
ISOMAP (q = 15)	1.004	1.024'	1.074*	1.309**	1.237**
ISOMAP (q = 30)	1.002	1.013	1.146**	1.305**	1.260**
ARX (PCAs (q = 5))	1.024	1.028	1.105	1.283**	1.173*
ARX (PCAq (q = 5))	1.021	1.018	1.110	1.287**	1.198**

Note: The relative MAE is calculated over the estimation sample from 2000 until 2019; the first model (AR(p), Minnesota) is the benchmark model, the rest of the values of the MAE are calculated relative to this model; q is the number of factors used for estimation; ': $p < 0.01$ *: $p < 0.05$, **: $p < 0.001$ indicate the statistical significance for each model relative to the benchmark.

5.3 Model Confidence Set

In addition to the Diebold-Mariano test the model confidence set (MCS) procedure is performed. This method selects the set of the best performing methods. These models conform to the null hypothesis of equal predictive ability at confidence level α. We calculated the MCS for $\alpha = 0.05$ and $\alpha = 0.1$. The MCS for those values is displayed in Table 4. Apart from six kernel ridge models and the PCA linear model with 30 factors and the AR(12) model with an horseshoe prior all models we used in our estimation are in both of these model confidence sets. Therefore we can not choose superior models based on these results. This is consistent with the insignificant differences in MAE and RMSE between models. However, this gives us more evidence that the more complex machine learning methods, especially the kernel ridge regression, are underperforming in this environment. It also does not reject our hypothesis that the PCA squared and quadratic model are the best performing dimension reduction techniques.

Table 4: Model confidence set for different α

Specification	**Minnesota**	**Horseshoe**	**Elastic Net**	**Random Forest**	**Kernel Ridge**
AR (p)	x x	x x	x x	x x	x x
PCA linear (q = 5)	x x	x x	x x	x x	x x
PCA linear (q = 15)	x x	x x	x x	x x	x x
PCA linear (q = 30)	x x	- -	x x	x x	x -
PCA squared (q = 5)	x x	x x	x x	x x	x -
PCA squared (q = 15)	x x	x x	x x	x x	x x
PCA squared (q = 30)	x x	x x	x x	x x	x x
PCA quadratic (q = 5)	x x	x x	x x	x x	x x
PCA quadratic (q = 15)	x x	x x	x x	x x	- -
PCA quadratic (q = 30)	x x	x x	x x	x x	x -
ISOMAP (q = 5)	x x	x x	x x	x x	- -
ISOMAP (q = 15)	x x	x x	x x	x x	x x
ISOMAP (q = 30)	x x	x x	x x	x x	x x
ARX (PCAs (q = 5))	x x	x x	x x	x x	x x
ARX (PCAq (q = 5))	x x	x x	x x	x x	- -

Note: The left x's display the model confidence set at a confidence level $\alpha = 0.05$ and the right x's display the model confidence set at confidence level $\alpha = 0.1$

6 Conclusion

In this paper, different combinations of dimension reduction techniques and machine learning methods were tested in forecasting inflation 3 months ahead using a rolling window framework from 2000 until 2019 with data from the FRED-MD database. Our main research question was, "How can machine learning algorithms improve the results of real-time quarterly-ahead inflation forecasts in combination with dimension reduction techniques?". We used mean absolute errors (MAE), root mean squared errors (RMSE), the Diebold-Mariano test and the model confidence set test (MCS) to compare and evaluate the forecasts and find an answer to this question.

The three machine learning methods (elastic net, random forest and kernel ridge regression) do not express very accurate forecasting performance. The random forest method and kernel ridge regression have significantly higher values for the evaluation metrics than the AR(12) benchmark model. Elastic net does not have many statistically significant differences in performance compared to the benchmark model. However, this method still showed high values

for RMSE and MAE. Therefore it is preferred to use a dynamic regression with Bayesian priors after applying dimension reduction techniques. This conclusion contradicts the sentiment of the literature regarding machine learning methods in forecasting environments. This difference is probably due to the specific data transformations made. These transformations can alter the predictive accuracy of the machine learning methods. The specific data transformation imposes an additional restriction to the model, while most machine learning methods can handle high dimensional data without dimension reduction. The differences in squared errors were greater than those in absolute errors. Thus, machine learning methods were more appropriate in turbulent times with larger outliers.

The dimension reduction techniques combined with a dynamic regression do not significantly improve the AR(12) benchmark model, which is in contrast to most of the literature. However, the squared and quadratic PCA demonstrate the most desirable values for all evaluation metrics. The insignificance could be due to the choice of the dataset, the particular dimension reduction methods chosen or the dynamic regression used to estimate the model.

Based on our results, we conclude that machine learning methods can not contribute to the accuracy of the forecasting of inflation using factors obtained through dimension reduction techniques. The dimension reduction techniques combined with a dynamic regression create a model which produces more accurate results. Because these methods are capable of capturing non-linearity's in the data and perform well in case of outliers we recommend applying these methods when forecasting inflation.

For further research, it would be interesting to test these specific dimension reduction techniques on other macro-economic datasets to search for scientific proof that methods as the PCA squared and PCA quadratic, which performed well in our paper, significantly outperform simpler benchmark models. We concluded that applying machine learning methods after dimension reduction techniques does not yield better results. However, both methods perform well in predicting inflation. Therefore dynamically combining both machine learning and factor-augmented models, as in Hauzenberger et al. (2023), could be interesting to construct a model that is more robust and does not depend on specific methods and test whether this way of combining models does improve the predictions. Moreover, the performance of machine learning methods applied to the higher-dimensional datasets could be examined and compared with the performance of factor-augmented methods.

References

Araujo, G. S. and Gaglianone, W. P. (2023). Machine learning methods for inflation forecasting in brazil: New contenders versus classical models. *Latin American Journal of Central Banking*, 4(2):100087.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory*.

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

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010). The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480.

Chan, J. C. (2019). Minnesota-type adaptive hierarchical priors for large bayesian vars. *SSRN Electronic Journal*.

Croushore, D. (2011). Frontiers of real-time data analysis. *Journal of Economic Literature*, 49(1):72–100.

Diebold, F. and Mariano, R. (1994). *Comparing predictive accuracy*.

Goulet, P., Leroux, M., Stevanovic, D., and Surprenant, S. (2021). Macroeconomic data transformations matter. *International Journal of Forecasting*, 37(4):1338–1354.

Goulet Coulombe, P., Leroux, M., Stevanovic, D., and Surprenant, S. (2022). How is machine learning useful for macroeconomic forecasting? *Journal of Applied Econometrics*, 37(5):920–964.

Hansen, P. R., Lunde, A., and Nason, J. M. (2011). The model confidence set. *Econometrica*, 79(2):453–497.

Hauzenberger, N., Huber, F., and Klieber, K. (2023). Real-time inflation forecasting using non-linear dimension reduction techniques. *International Journal of Forecasting*, 39(2):901–921.

Kastner, G. and Frühwirth-Schnatter, S. (2014). Ancillarity-sufficiency interweaving strategy (asis) for boosting mcmc estimation of stochastic volatility models. *Computational Statistics amp; Data Analysis*, 76:408–423.

Li, Y.-S., Pai, P.-F., and Lin, Y.-L. (2022). Forecasting inflation rates be extreme gradient boosting with the genetic algorithm. *Journal of Ambient Intelligence and Humanized Computing*, 14(3):2211–2220.

McCracken, M. W. and Ng, S. (2015). *Fred-MD: A monthly database for macroeconomic research*.

Medeiros, M., Vasconcelos, G., Veiga, A., and Zilberman, E. (2018). Forecasting inflation in a data-rich environment: The benefits of machine learning methods. *SSRN Electronic Journal*.

Stock, J. and Watson, M. (1999). *Forecasting inflation*.

Stock, J. and Watson, M. (2006). *Why has U.S. inflation become harder to forecast?*

Tenenbaum, J. B., Silva, V. d., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Uddin, M. F., Lee, J., Rizvi, S., and Hamada, S. (2018). Proposing enhanced feature engineering and a selection model for machine learning processes. *Applied Sciences*, 8(4):646.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320.

A Technical Appendix

A.1 Bayesian priors structure

For the estimation of the dynamic regression Bayesian priors are implemented. In this paper we use time-invariant parameters, this means we want to estimate the following regression:

$$y_{t+h} = d_t'\beta + \epsilon_{t+h}, \qquad \epsilon_{t+h} \sim N(0, \sigma^2_{t+h}) \tag{20}$$

where d_t is an $(M \times 1)$ vector, which consist of the lagged values of y_{t+h}, an intercept and q different factors. $\log(\sigma^2_{t+h})$ are stochastic volatility's. We use two different multivariate Gaussian priors on β:

$$\beta \mid \underline{V} \sim \mathcal{N}(0, \underline{V}) \tag{21}$$

The prior mean is set to zero. This is because our dependent variable y_{t+h} is stationairy and has supposedly a zero mean. $\underline{V}$ is a M-dimensional diagonal variance-covariance matrix ($\underline{V} = diag(\theta^2_1, ...\theta^2_M)$). $\theta^2_1, ...\theta^2_M$ are shrinkage parameters collected from the priors. The two priors considered are the Horseshoe prior (Carvalho et al., 2010) and the adaptive Minnesota prior (Chan, 2019).

1. For the Horseshoe prior $\theta^2_{j,H}$ is calculated as:

$$\theta^2_{j,H} = \tau^2_H \zeta^2_{j,H} \tag{22}$$

   Where τ_H is a global parameter which pushes the coefficients to 0 (the mean of β) and $\zeta_{j,H}$ is a variable-specific (local) scaling, which allows for deviations from zero for specific variables. This leads to heavy tails in the marginal prior, which is useful for forecasting. Both shrinkage parameters have a prior structure and are standard half Cauchy distributed

2. We consider the Adaptive Minnesota as in Carvalho et al. (2010). This prior treats the different type of variables (the intercept, the lags of inflation and the factors) in a different way. $\theta^2_{j,M}$ is calculated in a similar manner as for the Horseshoe prior:

$$\theta^2_{j,M} = \tau^2_{i,M} \zeta^2_{j,M} \tag{23}$$

   where i is the kind of independent variable j and

$$\zeta^2_{j,M} = \begin{cases} \frac{1}{l^2} & \text{for parameters associated with own lags of inflation (l = 1, ..., p)} \\ \frac{\hat{\sigma}^2_\pi}{\hat{\sigma}^2_k} & \text{for parameters associated with q factors (k = 1, ..., q)} \\ \hat{\sigma}^2_\pi & \text{for the intercept} \end{cases} \tag{24}$$

   $\hat{\sigma}^2_k$ is the estimated OLS variance of an AR(1) model of the particular variable. Thus, $\hat{\sigma}^2_\pi$ is the OLS variance of an AR(1) model on inflation. These local scaling parameters are constant, while $\tau_{i,M}$ features a hierarchical prior structure and is standard half-Cauchy

distributed for $i = 1, 2$, where $i = 1$ if j belongs to a lag of inflation, $i = 2$ if j belongs to a factor. For the intercept $\tau_{3,M} = 1$.

A.2 Bayesian updating

We use a Markov Chain Monte Carlo (MCMC) algorithm to get the posterior distribution of the parameters and log-volatility's and perform forecasts with the coefficients. We sample 10,000 times, updating the parameters within each iteration and saving the forecast error for each iteration. We set $\tau_H = \zeta_{j,H} = \nu_j = \phi_H = 1$ as starting values for the Horseshoe prior and and $\tau_{i,M} = \phi_{i,M} = 1$ as starting values for the Minnesota prior. With the starting values for θ we construct $\underline{V}$. We draw β from $\mathcal{N}(\overline{\beta}, \overline{V})$ with $\overline{V} = (\tilde{d}'\tilde{d} + \underline{V}^{-}1)^{-}1$ and $\overline{\beta} = \overline{V}(\tilde{d}\tilde{y})$. $\tilde{y}$ is a T-dimensional vector with elements y_t/σ_t and $\tilde{d}$ is a T x M matrix with rows $\tilde{d}_t/\sigma_t$. In each iteration σ_t and the elements of $\underline{V}$ are calculated based on the estimate of β and the prior parameters. This depends on the specific prior.

The horseshoe prior updates the parameters $\tau_H^2, \zeta_{j,H}^2, \nu_j, \phi_H$. These parameters follow an inverse Gamma distribution:

$$\zeta_{j,H}^2|\beta_j, \tau_H, \nu \sim InvGamma\left(1, \nu_j^{-1} + \frac{{\beta_j}^2}{2\tau_H^2}\right), \tag{25}$$

$$\tau_H^2|\beta_j, \zeta_{j,H}, \phi_H \sim InvGamma\left(\frac{2M+1}{2}, \phi^{-1} + \sum_{j=1}^{2M} \frac{\beta_j^2}{2\zeta_{j,H}^2}\right) \tag{26}$$

$$\nu_j|\zeta_{j,H} \sim InvGamma\left(1, 1 + \zeta_{j,H}^{-2})\right), \tag{27}$$

$$\phi|\tau_H \sim InvGamma\left(1, 1 + \tau_H^{-2}\right) \tag{28}$$

The minnesota prior updates the variabels $\tau_{1,M}, \tau_{2,M}, \tau_{3,M}, \phi_{i,M}$ in a similar way.

$$\tau_{1,M}^2|\beta_j, \zeta_{j,M}, \phi_1 \sim InvGamma\left(\frac{p+1}{2}, \phi_1^{-1} + \sum_{j=1}^{p} \frac{\beta_j^2}{2\zeta_{j,M}^2}\right) \tag{29}$$

$$\tau_{2,M}^2|\beta_j, \zeta_{j,M}, \phi_2 \sim InvGamma\left(\frac{M-p+1}{2}, \phi_2^{-1} + \sum_{j=p+1}^{M} \frac{\beta_j^2}{2\zeta_{j,M}^2}\right) \tag{30}$$

$$\phi_i|\tau_{i,M} \sim InvGamma\left(1, 1 + \tau_{i,M}^{-2}\right), \quad \text{for } i = 1, 2, 3. \tag{31}$$

A.2.1 Stochastic volatility

The variance of β is also updated every MCMC and follows a stochastic volatility model. The stochastic volatilities log (σ_{t+h}^2) are drawn from the algorithm of Kastner and Frühwirth-Schnatter (2014).

$$\log(\sigma_{t+h}^2) \sim \mathcal{N}(\mu + \varphi \log(\sigma_{t+h-1}^2), \eta) \tag{32}$$

where

$$\mu \sim \mathcal{N}(0, 1) \tag{33}$$

$$\frac{\varphi + 1}{2} \sim B(25, 5) \tag{34}$$

$$\eta \sim \Gamma(0.5, 10) \tag{35}$$

B Programming code

We used R to implement our methods. Parts of the code of Hauzenberger et al. (2023) were used. The code first gathers the data from a folder and stores it in a list. Thereafter all functions and characteristics for each model are specified. For the machine learning functions we used the **randomForest**, **glmnet** and **kernlab** package from R. Additionally we used the **caret** package for cross validation of the parameters of elastic net. In the machine learning functions we construct the model for an X and y and calculate forecast errors based on the predictions made with these models.

The dynamic regression estimation is performed through a function which uses X and y. It first sets up initial values and calculates the hyperparameters which stays constant over all MCMC iterations. Thereafter the Bayesian priors and stochastic volatility's are updated 10,000 times in a for loop according to the joint posterior distribution of the parameters. We use the package **stochvol** for sampling the stochastic volatilities. The distribution of the Bayesian priors is implemented by the code itself.

For each specified model a for loop is executed over all the months ranging from January 2000 until December 2019. For all rolling windows the X and y are constructed based on the data available in the estimation sample and the specific dimension reduction technique, to construct the factors for ISOMAP, the **vegan** package is employed. The X and y for the first month of the hold-out sample are also stored to evaluate the predictions. This data is used to carry out the dynamic regressions and machine learning functions for each window and retrieve the forecast errors.

The RMSE, MAE, Diebold-Mariano test and Model Confidence Set are calculated with these errors using the R packages **MCS** and **forecast**.

C Test statistics and p-values

Table 5: Diebold-Mariano test statistic for all specifications calculated relative to the benchmark AR(p) model over the complete sample using an absolute loss function

Specification	**Minnesota**	**Horseshoe**	**Elastic Net**	**Random Forest**	**Kernel Ridge**
AR (p)	0.000 (0.000)	2.447* (0.015)	-0.783 (0.435)	-3.501** (0.001)	-2.538* (0.012)
PCA linear (q = 5)	-0.549 (0.583)	-0.548 (0.584)	-1.539 (0.125)	-3.460** (0.001)	-3.090** (0.002)
PCA linear (q = 15)	-0.614 (0.540)	-0.687 (0.493)	-1.871' (0.063)	-3.553** (0.000)	-3.552** (0.000)
PCA linear (q = 30)	-0.424 (0.672)	-1.524 (0.129)	-2.546* (0.012)	-4.148** (0.000)	-3.932** (0.000)
PCA squared (q = 5)	-0.206 (0.837)	0.107 (0.915)	-1.002 (0.318)	-3.283** (0.001)	-2.732** (0.007)
PCA squared (q = 15)	-0.359 (0.720)	-0.296 (0.767)	-1.350 (0.178)	-3.901** (0.000)	-3.075** (0.002)
PCA squared (q = 30)	-2.093' (0.037)	-1.360 (0.175)	-2.688** (0.008)	-4.084** (0.000)	-2.699** (0.007)
PCA quadratic (q = 5)	-0.387 (0.699)	0.007 (0.995)	-0.325 (0.746)	-3.616** (0.000)	-2.528* (0.012)
PCA quadratic (q = 15)	-1.047 (0.296)	-1.013 (0.312)	-0.958 (0.339)	-3.596** (0.000)	-3.472** (0.001)
PCA quadratic (q = 30)	-1.050 (0.295)	-1.540 (0.125)	-2.182* (0.030)	-3.791** (0.000)	-3.107** (0.002)
ISOMAP (q = 5)	-1.038 (0.300)	-1.322 (0.187)	-2.051* (0.041)	-3.754** (0.000)	-2.769** (0.006)
ISOMAP (q = 15)	-0.405 (0.686)	-1.683' (0.094)	-2.565* (0.011)	-3.481** (0.001)	-3.252** (0.001)
ISOMAP (q = 30)	-0.139 (0.890)	-0.646 (0.519)	-2.957** (0.003)	-3.579** (0.000)	-3.339** (0.001)
ARX (PCAs (q = 5))	-0.866 (0.387)	-0.776 (0.438)	-1.241 (0.216)	-3.330** (0.001)	-2.352* (0.019)
ARX (PCAq (q = 5))	-0.705 (0.481)	-0.555 (0.579)	-1.292 (0.198)	-3.465** (0.001)	-2.779** (0.006)

Note: The first number in each column is the test statistic and in between brackets is the corresponding p-value; The test statistics are calculated over the period 2000-2019; the first model (AR(p), Minnesota) is the benchmark model; q is the number of factors used for estimation; ': p ¡ 0.01 *: p ¡ 0.05, **: p ¡ 0.001 indicate the statistical significance for each model relative to the benchmark.

Table 6: Diebold-Mariano test statistic for all specifications calculated relative to the benchmark AR(p) model over the complete sample using a squared loss function

Specification	**Minnesota**	**Horseshoe**	**Elastic Net**	**Random Forest**	**Kernel Ridge**
AR (p)	0.000 (0.000)	1.688' (0.093)	-0.877 (0.381)	-2.234* (0.026)	-0.914 (0.361)
PCA linear (q = 5)	-1.389 (0.166)	-1.297 (0.196)	-1.323 (0.187)	-2.164* (0.031)	-1.476 (0.141)
PCA linear (q = 15)	-1.428 (0.154)	-1.296 (0.196)	-1.522 (0.129)	-2.229* (0.027)	-2.031* (0.043)
PCA linear (q = 30)	-1.371 (0.172)	-1.578 (0.116)	-1.642' (0.102)	-2.427** (0.016)	-2.283* (0.023)
PCA squared (q = 5)	0.391 (0.696)	0.498 (0.619)	-1.021 (0.308)	-2.266* (0.024)	-1.411 (0.160)
PCA squared (q = 15)	-0.282 (0.778)	0.211 (0.833)	-0.810 (0.419)	-2.548** (0.011)	-1.720' (0.087)
PCA squared (q = 30)	-1.980' (0.049)	-0.998 (0.319)	-1.918' (0.056)	-2.754** (0.006)	-1.701' (0.090)
PCA quadratic (q = 5)	0.365 (0.715)	0.538 (0.591)	-0.754 (0.451)	-2.330* (0.021)	-1.392 (0.165)
PCA quadratic (q = 15)	-0.575 (0.566)	-0.553 (0.581)	-0.573 (0.567)	-2.163* (0.032)	-1.747' (0.082)
PCA quadratic (q = 30)	-0.894 (0.372)	-1.340 (0.182)	-2.001' (0.047)	-2.466** (0.014)	-1.891' (0.060)
ISOMAP (q = 5)	-0.893 (0.373)	-0.746 (0.457)	-1.405 (0.161)	-2.491** (0.013)	-1.374 (0.171)
ISOMAP (q = 15)	-1.150 (0.251)	-1.301 (0.194)	-2.302* (0.022)	-2.416** (0.016)	-1.931' (0.055)
ISOMAP (q = 30)	-0.157 (0.875)	0.324 (0.746)	-2.245* (0.026)	-2.448** (0.015)	-2.037* (0.043)
ARX (PCAs (q = 5))	-0.892 (0.373)	-0.829 (0.408)	-1.056 (0.292)	-2.170* (0.031)	-1.223 (0.223)
ARX (PCAq (q = 5))	-0.708 (0.480)	-0.626 (0.532)	-1.045 (0.297)	-2.259* (0.025)	-1.643' (0.102)

Note: The first number in each column is the test statistic and in between brackets is the corresponding p-value; The test statistics are calculated over the period 2000-2019; the first model (AR(p), Minnesota) is the benchmark model; q is the number of factors used for estimation; ': p ¡ 0.01 *: p ¡ 0.05, **: p ¡ 0.001 indicate the statistical significance for each model relative to the benchmark.

D Data transformations

Table 7: Variables used for estimation

Classification	Variable Name	Variable Description	I(0)
Real activity	RPI	Real personal income	5
	W875RX1	Real personal income ex transfer receipts	5
	INDPRO	IP Index	5
	IPFPNSS	IP: Final Products	5
	IPFINAL	IP: Final Products (Market Group)	5
	IPCONGD	IP: Consumer Goods	5
	IPMAT	IP: Materials	5
	IPMANSICS	IP: Manufacturing (SIC)	5
	CUMFNS	Capacity Utilization: Manufacturing	2
	RETAILx	Retail and Food Services Sales	5
	AMDMNOx	New Orders for Durable goods	5
	ANDENOx	New Orders for Nondefense Capital goods	5
	AMDMUOx	Unfilled Orders for Durable goods	5
	BUSINVx	Total Business Inventories	5
	ISRATIOx	Total Business: Inventories to Sales Ratio	2
	UMCSENTx	Consumer Sentiment Index	2
	CMRMTSPLx	Real Manu. and TradeIndustries Sales	5
Housing	HOUST	Housing Starts: Total New Privately Owned	4
	HOUSTNE	Housing Starts, Northeast	4
	HOUSTMW	Housing Starts, Midwest	4
	HOUSTS	Housing Starts, South	4
	HOUSTW	Housing Starts, West	4
	PERMIT	New Private Housing Permits (SAAR)	4
	PERMITNE	New Private Housing Permits, Northeast (SAAR)	4
	PERMITMW	New Private Housing Permits, Midwest (SAAR)	4
	PERMITS	New Private Housing Permits, South (SAAR)	4
	PERMITW	New Private Housing Permits, West (SAAR)	4
Labor	market	CLF16OV Civilian Labor Force	5
	CE16OV	Civilian Employment	5
	UNRATE	Civilian Unemployment Rate	2
	UEMPMEAN	Average Duration of Unemployment (Weeks)	2
	UEMPLT5	Civilians Unemployed : Less Than 5 Weeks	5
	UEMP5TO14	Civilians Unemployed for 41760 Weeks	5
	UEMP15OV	Civilians Unemployed : 15 Weeks & Over	5
	UEMP15T26	Civilians Unemployed for 15-26 Weeks	5
	UEMP27OV	Civilians Unemployed for 27 Weeks and Over	5
	CLAIMSx	Initial Claims	5

Table 7: Variables used for estimation

Classification	Variable Name	Variable Description	I(0)
	PAYEMS	All Employees: Total nonfarm	5
	USGOOD	All Employees: Goods-Producing Industries	5
	CES1021000001	All Employees: Mining and Logging: Mining	5
	USCONS	All Employees: Construction	5
	MANEMP	All Employees: Manufacturing	5
	DMANEMP	All Employees: Durable goods	5
	NDMANEMP	All Employees: Nondurable goods	5
	SRVPRD	All Employees: Service-Providing Industries	5
	USWTRADE	All Employees: Wholesale Trade	5
	USTRADE	All Employees: Retail Trade	5
	USFIRE	All Employees: Financial Activities	5
	USGOVT	All Employees: Government	5
	CES0600000007	Avg Weekly Hours: Goods-Producing	1
	AWOTMAN	Avg Weekly Overtime Hourse: Manufacturing	2
	AWHMAN	Avg Weekly Hours: Manufacturing	1
	CES0600000008	Avg Hourly Earnings: Goods-Producing	6
	CES2000000008	Avg Hourly Earnings: Construction	6
	CES3000000008	Avg Hourly Earnings: Manufacturing	6
Prices	OILPRICEx	Crude Oil, , spliced WTI and Cushing	6
	PPICMM	PPI: Metals and metal products	6
	CPIAUCSL	CPI : All Items	6
	CPIAPPSL	CPI : Apparel	6
	CPITRNSL	CPI : Transportation	6
	CPIMEDSL	CPI : Medical Care	6
	CUSR0000SAC	CPI : Commodities	6
	CUSR0000SAS	CPI : Services	6
	CPIULFSL	CPI : All Items Less Food	6
	CUSR0000SA0L5	CPI : All Items Less Medical Care	6
Money stocks	M1SL	M1 Money Stock	6
	M2SL	M2 Money Stock	6
	M2REAL	Real M2 Money Stock	5
	AMBSL	St. Louis Adjusted Monetary Base	6
Reserves & loans	TOTRESNS	Total Reserves of Depository Institutions	6
	NONBORRES	Reserves of Depository Institutions	7
	BUSLOANS	Commercial and Industrial Loans	6
	REALLN	Real Estate Loans at All Commerical Banks	6
	NONREVSL	Total Nonrevolving Credit	6
	CONSPI	Nonrevolving consumer credit to Personal Income	2
	MZMSL	MZM Money Stock	6

Table 7: Variables used for estimation

Classification	Variable Name	Variable Description	I(0)
	DTCOLNVHFNM	Consumer Motor Vehicle Loans Outstanding	6
	DTCTHFNM	Total Consumer Loans and Leases Outstanding	6
	INVEST	Securities in Bank Credit at All Commercial Banks	6
	CP3Mx	3-Month AA Financial Commercial Paper Rate	2
Interest rates	FEDFUNDS	Effective Federal Funds Rate	2
	TB3MS	3-Month Treasury Bill	2
	TB6MS	6-Month Treasury Bill	2
	GS1	1-Year Treasury Rate	2
	GS5	5-Year Treasury Rate	2
	GS10	10-Year Treasury Rate	2
	AAA	MoodyÕs Seasoned Aaa Corporate Bond Yield	2
	BAA	MoodyÕs Seasoned Baa Corporate Bond Yield	2
	COMPAPFFx	3-Month Commercial Paper Minus FEDFUNDS	1
	TB3SMFFM	3-Month Treasury C Minus FEDFUNDS	1
	TB6SMFFM	6-Month Treasury C Minus FEDFUNDS	1
	T1YFFM	1-Year Treasury C Minus FEDFUNDS	1
	T5YFFM	5-Year Treasury C Minus FEDFUNDS	1
	T10YFFM	10-Year Treasury C Minus FEDFUNDS	1
	AAAFFM	Moody's Aaa Corporate Bond Minus FEDFUNDS	1
	BAAFFM	Moody's Baa Corporate Bond Minus FEDFUNDS	1
Stock market	TWEXMMTH	Trade Weighted U.S. Dollar Index: Major Currencies	5
	EXSZUSx	Switzerland / U.S. Foreign Exchange Rate	5
	EXJPUSx	Japan / U.S. Foreign Exchange Rate	5
	EXUSUKx	U.S. / UK Foreign Exchange Rate	5
	EXCAUSx	Canada / U.S. Foreign Exchange Rate	5
	S.P.500	S&Ps Common Stock Price Index: Composite	5
	S.P..indust	S&Ps Common Stock Price Index: Industrials	5
	S.P.div.yield	S&Ps Composite Common Stock: Dividend Yield	2
	S.P.PE.ratio	S&Ps Composite Common Stock: Price-Earnings Ratio	5

Note: the column I(0) denotes the transformation of each variable to obtain stationairity where 1. No transformation, 2. Δx_t, 4. $\log(x_t)$, 5. $\Delta \log(x_t)$, 6. $\Delta^2 \log(x_t)$, 7. $\Delta(x_t/X_{t-1} - 1.0)$