Erasmus University Rotterdam

Erasmus School of Economics

Improving Macroeconomic Forecasts Using New Data Transformations

Bachelor Thesis Quantitative Finance

Student:

Niek Heijmans 535649

Supervisor:

Eoghan O'Neill

Second assessor:

Daan Opschoor

Date final version:

July 2022

Abstract

In the nonlinear machine learning (ML) macroeconomic forecasting environment, linear transformations of variables can alter predictions. In this paper, we investigate whether we could improve macroeconomic forecasts, by examining new data transformations, which are variations on MARX, a method introduced by Goulet Coulombe et al. (2021). Since nonlinear ML algorithms are used more frequently and the group of practitioners in this field is growing, we aim to contribute to this field by finding new data transformations that could be used when forecasting macroeconomic variables. Using the FRED-MD dataset, we perform a pseudo-out-of-sample forecasting experiment. Our new data transformations are used in the Random Forest (RF) algorithm. It is found that EMARX, a transformation using exponential moving averages in MARX, and FMARX, a combination of MARX and PCA dimensionality reduction, produce the best forecasts for more than half of our variables and horizons. We conclude that these new methods should often be considered when forecasting macroeconomic variables.

Table of Contents

1 Introduction

In the macroeconomic forecasting environment, Machine Learning (ML) methods and the use of the widespread availability of big data have recently become increasingly popular. Compared to traditional methods, nonlinear ML methods have shown sizable forecasting gains for macroeconomic variables, for example concluded by Hall (2018), Kim & Swanson (2018) and Medeiros et al. (2021). In these environments, the linear transformations of the regressor matrix X can alter predictions, and as Goulet Coulombe et al. (2021) conclude, the standard of taking first differences is not the optimal transformation. In our research, we have been able to measure the performance of different and new data transformations, in combination with ML algorithms. We have observed which model specifications have better predicting ability, and which do not. As a result, the growing number of researchers in this field are able to use our research when deciding which methods to use. Moreover, the methods that we conclude to have the best forecasting ability, can be used in practical applications. Researchers will be able to make more accurate forecasts in the macroeconomic field, even during periods of high uncertainty (Goulet Colombe et al., 2021).

Goulet Coulombe et al. (2021) have introduced and examined a new data transformation called Moving Average Rotation of X (MARX). They conclude that this non-standard transformation shows very promising results when it comes to macroeconomic forecasting in combination with ML algorithms. However, the use of a Moving Average (MA) approach in MARX may not be optimal. Goulet Coulombe (2020) suggests that laws of motion other than MA can be accommodated. Moreover, variations on MARX using PCA could also possibly improve forecasts. In this paper, we investigate whether we could improve macroeconomic forecasts, by examining new variations on the MARX data transformation in ML algorithms. Eventually, we try to answer our main research question *"How do variations of MARX perform compared to already existing data transformations in macroeconomic forecasting?"*.

To answer our main question, we have analysed the large macroeconomic dataset FRED-MD from McCracken & Ng (2016). We applied our methods on 10 different target variables and 6 different forecasting horizons. As benchmarks for our research, we have used two simple, linear forecasting methods, namely the Autoregressive (AR) and Factor Model (FM). The ML algorithm that we have used is Random Forest (RF), which was fed with 6 different data transformations. The reason why we use RF, is that it is a nonlinear ML algorithm that does not require any optimised hyperparameters. Recent literature, like from Yoon (2021) and Medeiros et al. (2021), have shown that RF performs well in a macroeconomic forecasting environment. For the data transformations, we examined three old and three new approaches. The already existing data transformations are single-period differences and growth rates following McCracken & Ng (2016), their principal components, and MARX that was introduced by Goulet Coulombe et al. (2021). The single-period differences and growth rates serve as a benchmark, since this is straightforward and does not require much transforming of the data. Goulet Coulombe et al. (2021) have concluded that transforming the data into factors is probably the most effective form of feature engineering, which is why we have also included this old method and investigate whether our new methods are

an improvement. MARX has also proven to be a worthy contender when it comes to minimizing prediction errors, according to Goulet Coulombe et al. (2021). Our new data transformations all are variations on this method. First, we have examined EMARX, where Exponential Moving Average (EMA) is used as the law of motion in MARX instead of simple MA. The idea of applying a different law of motion was suggested by Goulet Coulombe (2020), and Holt (2004) shows that EMA has its benefits when forecasting seasonals and trends in the macroeconomic field. Second, we have tried to combine the successes of factorization and MARX, by introducing the data transformations FMARX and MARF. FMARX applies data dimensionality reduction by PCA on the data that is obtained by MARX, and MARF applies MARX on the factors that are obtained by PCA. Since both MARX and PCA have proven their usefulness in macroeconomic forecasting, a combination of the two could possibly improve the accuracy of forecasts even further.

In our research, we have seen and concluded that certain variations on MARX produce the best macroeconomic forecasts, outperforming older methods and the original MARX approach. EMARX and FMARX together are the best model specification in more than half of all cases. The success of EMARX suggests that the use of EMA as the law of motion is a better choice than simple MA. It also brings up the possibility that there are many laws of motion other than MA that could improve MARX. The results of FMARX show that this method produces the best forecasts for a certain group of variables, especially for longer horizons. FMARX outperforming standard factors of the raw data, suggests that forecasters should prefer applying PCA on smoothed data obtained by a method like MARX. Moreover, using reduced dimensionality helps obtaining better forecasts, since FMARX often outperforms MARX. This is probably because MARX is relatively high dimensional. MARF, however, does not produce a set of regressors with interesting forecasting ability. Apparently, smoothed factors do not function well in macroeconomic forecasting, while factors of smoothed data do.

From these results, we learn that MARX and variations on MARX can produce more accurate macroeconomic forecasts. When forecasting macroeconomic variables in the future using nonlinear ML algorithms, researchers should now be aware that methods like EMARX and FMARX should be included in the analysis as data transformations. We expect that there are endless possibilities that could improve these new methods, by for example combining the two methods into FEMARX, where PCA is applied on the data obtained by EMARX. Eventually, practitioners in the field of macroeconomic forecasting should be able to find the best model specification per macroeconomic variable and forecasting horizon.

The rest of this paper is organized as follows. In Section 2 we discuss some relevant literature relating to our research and the methods that we have used. In Section 3 we give insight in the data that we have gathered, and in the next section we give a description and interpretation of the methods that we have used to obtain an answer to our research question from this data. In Section 5 we present the resulting outcome of these methods and data. We provide a general discussion regarding our research in Section 6 and we conclude in Section 7. References and additional information can be found at the end of the paper in the appendix.

2 Literature Review

The possibility to obtain an extensive amount of new and accessible economic data provides the opportunity to significantly improve macroeconomic forecasts. However, this calls for more complex models that can capture more of this wide range of economic data. Therefore, Hall (2018) has examined the performance of new ML methods compared to older methods when forecasting macroeconomic variables. These older methods included consensus predictions and statistical modeling techniques. With his analysis, he has shown that a more complex model can significantly outperform those older methods. He states that the key to this result is using regularization to control model complexity. Goulet Coulombe et al. (2019) use the insight that ML methods perform well in large macroeconomic data, and study important features driving this performance. They conclude that nonlinear models are the true game changer for data-rich environments, particularly during periods of macroeconomic uncertainty and financial frictions. The usefulness of nonlinear models is also concluded by Maehashi & Shintani (2020). They found that factor models and ML methods perform better than the linear AR model when forecasting macroeconomic data for Japan. Goulet Coulombe et al. (2021) try to find the best ML method, in combination with data transformations. One of the methods that seems to perform best is Random Forest (RF). This algorithm, which is explained by Lee et al. (2020), uses multiple decision trees to reduce the number of regressors, without overfitting. It is convenient to use, because it is nonlinear and does not require any optimised hyperparamaters. The valuable performance of the RF algorithm is also shown by Yoon (2021) and Medeiros et al. (2021).

Not only the choice for the ML method matters, but also the data transformations that are applied before using ML. This is shown by Goulet Coulombe et al. (2021), who demonstrate that including different data transformations can improve the macroeconomic forecasts. They conclude that in most cases, transforming the data into factors is probably the most effective form of feature engineering. This is also in line with the results of Goulet Coulombe et al. (2019), but in contrast with Medeiros et al. (2021). One of the differences is that Goulet Coulombe et al. (2019) explore the possibility that factors alone could be better than the original data, rather than always both together, which turns out to be the case with RF. Both the factors and their lags are included in the analysis, à la Stock & Watson (2002), so obtained by Principal Component Analysis (PCA).

In the paper of Goulet Coulombe (2020), he shows that time-varying parameter models are in fact ridge regressions. By considering a generic linear model with Random Walk (RW) time-varying parameters, he obtains a ridge regression by reparametrization. This discovery makes computations, tuning, and implementation much easier. Goulet Coulombe et al. (2021) then use these insights to translate the time-varying parameter model to regularized lag polynomials, following Shiller (1973). Eventually, they obtain their new proposed data transformation Moving Average Rotations of X (MARX). They have shown that non-standard choices of macroeconomic data transformations, particularly MARX, minimize the prediction error in macroeconomic forecasting, especially at shorter horizons. The great performance of these non-standard transformations is often paired with nonlinear ML algorithms, particularly RF.

With these promising results, one could argue that transforming data with MARX should be analysed more. For now, since it is a very recently found method, MARX has only been experimented with by using RW or simple Moving Average (MA) as the law of motion for the time-varying parameters. As is suggested by Goulet Coulombe (2020), there is a wide range of laws of motion that are possible to implement instead of RW or MA in MARX. Wolff (1987) found that the introduction of time-varying parameters enhances the forecasting performance of structural exchange rate models, with the RW forecasting rule not always as the best performing one. One of the alternative law of motions that could be worth considering is Exponentially weighted Moving Average (EMA). The details and benefits of using this method when forecasting seasonals and trends are clearly explained by Holt (2004). The method was also already used by Winters (1960), who showed that it is able to obtain better sales forecasts than more traditional methods. Hence, using EMA rather than MA in MARX, could help to obtain better macroeconomic forecasts.

Following the results of previous literature, perhaps a combination of PCA and MARX could improve macroeconomic forecasting. Goulet Coulombe et al. (2021) already considered both PCA and MARX separately as the data transformation, which sometimes showed to be a good collaboration. We would like to take it a step further, by (1) applying MARX on the factors of the data, and (2) applying PCA on MARX. Both have not been done before, but by applying PCA also on the lags of MARX, has some things in common with Dynamic PCA (DPCA). DPCA was proposed by Ku et al. (1995), a method where PCA is performed on a data matrix including time-lagged values of the variables. As Breitung & Eickmeier (2006) demonstrate, dynamic factor models turn out to be useful in investigating macroeconomic problems. The model still had some issues, for example the number of factors or lags. By using an ML method like RF, this issue could be resolved. These previous results regarding DPCA, make us expect that a combination of PCA and MARX could provide some interesting results.

3 Data

For our research, we have used the same dataset as is used by Goulet Coulombe et al. (2021), namely the FRED-MD dataset from McCracken & Ng (2016). This is a large, monthly frequency, macroeconomic database, including 128 macroeconomic variables. The dataset starts in January 1959 and is updated and shared online every month. Just like Goulet Coulombe et al. (2021), we use data for estimation from January 1960, and our forecasting period ends in December 2017. We have used the online version that was released in January 2019. Some variables in the dataset contained missing values for the period that we have analysed, so we decided to omit all those variables, resulting in 123 macroeconomic variables remaining. We will refer to this dataset as $H_t = [H_{1t}, ..., H_{Kt}]$ for $t = 1, ..., T$, with K as the number of covariates.

We have analysed 10 target variables, the same as Goulet Coulombe et al. (2021). These are the industrial production index (INDPRO), total nonfarm employment (EMP), unemployment rate (UNRATE), real personal income excluding current transfers (INCOME), real personal consumption

expenditures (CONS), retail and food services sales (RETAIL), housing starts (HOUST), M2 money stock (M2), consumer price index (CPI), and the production price index (PPI). We are particularly interested in target variables where RF in combination with MARX performed well following the results of Goulet Coulombe et al. (2021), since our research is focusing on these methods.

3.1 Target Variable Transformation

The target variables are transformed in a different way than the rest of the data. Goulet Coulombe et al. (2021) use the same approach as Kim & Swanson (2018), where transformations are used to induce stationarity. For the predictions made at horizons of 1, 3, 6, 9, 12 and 24 months (denoted as h), they are effectively targeting the average growth rate over these periods, except for UNRATE for which they target the average difference. Let Y_t be the target level variable, which is not transformed and coming from H_t. Then, the model is fitted on the target variable denoted by y_{t+h} directly, and $\hat{y}_{t+h}$ is used as prediction (corresponding to $\hat{y}_{t+h}^{\text{direct}}$ used by Goulet Coulombe et al. (2021)). For the average difference we have used the same formula, where the target variable is computed by $y_t = (Y_t - Y_{t-h})/h$. However, for the average growth rate we have used a different approach than Goulet Coulombe et al. (2021) and Kim & Swanson (2018). They transform their variable by $y_t = \ln(Y_t/Y_{t-h})$, where they make use of the approximation that this log difference is close to average growth rates when the changes are small. This may be the case for small horizons, but we think this assumption of small changes is not always feasible for bigger horizons. That is why we have not used an approximation, but transformed our variables by computing the average growth rate. This corresponds to $y_t = \frac{1}{h}\sum_{h'=1}^{h}\left(\frac{Y_{t+h'-h}-Y_{t+h'-1-h}}{Y_{t+h'-1-h}}\right)$.

4 Methodology

Using the data, target variables and horizons explained in the previous section, we describe our used methods in this section. It contains a detailed explanation of the forecasting algorithms and data transformations that we have used. All computations are performed in Python.

To examine and compare the forecasting performance of the different methods, we have done a pseudo-out-of-sample (POOS) forecasting experiment. We use an expanding window for estimation starting from January 1960. The POOS period starts in January 1980 and ends in December 2017. We evaluate the performance of point forecasts using the Root Mean Squared Error (RMSE). The RMSE for variable v and horizon h is computed by

$$RMSE_{v,h} = \sqrt{\frac{1}{\#\text{OOS}}\sum_{t\in\text{OOS}}(y_{t+h}^{v} - \hat{y}_{t+h}^{v})^2}, \tag{1}$$

where OOS is the set of all POOS periods and $\hat{y}_{t+h}^{v}$ is the computed value that is predicted at time t for h months later.

4.1 Forecasting Methods

We have considered the Factor Model (FM) from Stock & Watson (2002). The model is given by

$$y_{t+h} = \beta Z_t + \varepsilon_{t+h}, \tag{2}$$

where β is the coefficient vector and ε_{t+h} is the error term at $t+h$. Z_t is our feature matrix, defined by $Z_t := [y_t, ..., L^{P_y} y_t, F_t, ..., L^{P_f} F_t]$, where L is the lag operator, P_y is the autoregressive lag order, P_f is the factor lag order and F_t are the factors obtained by PCA of X_t (explained in Section 4.2.1). PCA is performed by using `PCA` from the Python package `sklearn.decomposition` from Pedregosa et al. (2011). The AR model is obtained by narrowing down Z_t to a matrix only containing the target variable and its lagged values. The hyperparameters P_y, P_f, and the number of principal components R are selected by using BIC. The model is estimated monthly, while hyperparameter selection is performed every two years. AR and FM are considered as forecasting methods, since they are relatively easy to implement and can be seen as a benchmark.

For Random Forest, we have used the same algorithm as described in the appendix from Goulet Coulombe et al. (2021), with only a small adjustment. Parameters are chosen to reduce running time, but without losing accuracy. A detailed explanation of the algorithm and chosen parameters can be found in Appendix A. The regressors that are considered by the RF algorithm depend on matrix Z_t, which contains transformations of the data, further explained in the next subsection. The RF algorithm is performed by using `RandomForestRegressor` from the Python package `sklearn.ensemble` from Pedregosa et al. (2011). We use RF, since it is a well-performing nonlinear ML algorithm, particularly in combination with MARX, as explained in Section 2.

4.2 Data Transformations

To use our forecasting methods, we first have to obtain a set of regressors that should be considered. A certain data transformation, denoted by f_Z, is applied to the original data H_t, to obtain our transformed data including regressors to be considered by the forecasting method. The transformed data is denoted by $f_Z(H_t) = Z_t$. In this subsection, we present the selection of Z_t's that we have considered. First, the transformations also appearing in the paper by Goulet Coulombe et al. (2021), and after that we present our new proposed choices for f_Z.

4.2.1 X and F

McCracken & Ng (2016) propose a specific transformation method for every variable in their dataset to obtain a stationary X_t out of H_t. We have used the transformed matrix $X = [X_t]_{t=1}^T$ as a regressor matrix, as well as to obtain a matrix F containing the principal components of X. When X is considered as the transformed data, all variables and their lagged values are included as regressors. On the other hand, when using F, only a small number of principal components and their lags are considered as regressors. The full principal components decomposition of X, first found by Pearson

(1901), can be given as

$$F = XW,$$

where W is a $K \times K$ matrix whose columns are the eigenvectors of $X'X$. When we only want to consider the first R principal components, we use W_R which has only the first R columns of W, and obtain F_R, which now has T rows but only R columns. F and X are considered as data transformations, to see if there is any improvement when using our new transformations compared to these relatively simple ones.

4.2.2 MARX

Moving Average Rotations of X (MARX) is first used by Goulet Coulombe et al. (2021). Since some of the matrices and vectors used in their explanation are not always clearly defined, we try to clarify where needed. We start our explanation by using a certain equation they came up with. They start with a derivation which is a simple translation of Goulet Coulombe (2020)'s insights for time-varying parameters model to regularized lag polynomials, following Shiller (1973). After rewriting a generic regularized ARDL model as a ridge regression and some reparametrization, they obtain the Ridge regression problem given by

$$\min_{\boldsymbol{\theta}} (\boldsymbol{y} - \boldsymbol{Z\theta})'(\boldsymbol{y} - \boldsymbol{Z\theta}) + \lambda \boldsymbol{\theta'\theta}. \tag{3}$$

On the right hand side of the plus sign is the regularization, including λ as the degree of rigidity. In the equation, $\boldsymbol{Z} \equiv \boldsymbol{XC}$ and $\boldsymbol{\theta} = \boldsymbol{D\beta}$, with $\boldsymbol{D} = \boldsymbol{C}^{-1}$. $\boldsymbol{X}$ is the regressor matrix, same as the original X matrix, but including lagged values. It is a $T \times KP_m$ matrix, where P_m is the number of lags used when applying MARX. The matrix starts with all lagged values of the first variable, with the earliest lag in the front. Hence, our matrix is defined as $\boldsymbol{X} = [X_{t-P_m+1,1}, ..., X_{t,1}, X_{t-P_m+1,2}, ..., X_{t,2}, ..., X_{t-P_m+1,K}, ..., X_{t,K}]$. Every column has its own coefficient, denoted by $\beta_{p,k}$, where p corresponds to the lag and k to the variable. So, we define the vector $\boldsymbol{\beta} = [\beta_{P_m-1,1}, ..., \beta_{0,1}, \beta_{P_m-1,2}, ..., \beta_{0,2}, ..., \beta_{P_m-1,K}, ..., \beta_{0,K}]'$.

Now, we are particularly interested in the remaining matrix $\boldsymbol{C}$, since $\boldsymbol{C}$ transforms the matrix $\boldsymbol{X}$ to $\boldsymbol{Z}$, which is the matrix that is eventually used in the ML method. Additionally, we are interested in its inverse $\boldsymbol{D}$, since it will give us an intuition of how the regularization is defined. Depending on the choice for $\boldsymbol{C}$, a large number of law of motions for the time-varying parameters can be used. Goulet Coulombe et al. (2021) use a Random Walk (RW) as an example of the law of motion of their time-varying parameters. We define the matrix C for a single variable, and note that $\boldsymbol{C} = I_K \otimes C$, with C a $P_m \times P_m$ matrix. In the case of RW, C is a lower triangular matrix of

ones. For the simple case of one parameter and $P_m = 4$, we have

$$\beta = \begin{bmatrix} \beta_3 \\ \beta_2 \\ \beta_1 \\ \beta_0 \end{bmatrix} = C^{\text{RW}}\theta^{\text{RW}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \beta_3 \\ \beta_2 - \beta_3 \\ \beta_1 - \beta_2 \\ \beta_0 - \beta_1 \end{bmatrix}.$$

The inverse D^{RW} of C^{RW} is the first difference operator. The regularization in this case, for general K and P_m, is given by

$$\lambda \boldsymbol{\theta'\theta} = \lambda \sum_{k=1}^{K} \sum_{p=1}^{P_m} (\beta_{p-1,k} - \beta_{p,k})^2, \tag{4}$$

where $\beta_{P_m,k}$ should be interpreted as zero for all k, since it does not exist. The regularization equation shows that for every variable k and every lag order p, the difference between a lag coefficient and its neighbours should be as low as possible to minimize the Ridge regression in equation (3). This can be interpreted as that β_p shrinks to β_{p-1} and β_{p+1} rather than to zero, which is in line with the idea of Goulet Coulombe et al. (2021) that a simple linear combination of lags impacts the target variable rather than a single one of them. This new prior for the lags, coming from the ideas of Shiller (1973), causes the lag curve to be more smooth.

For the RW case, the interpretation for the matrix $\boldsymbol{Z}$ is as follows. For a single variable k, the matrix is defined as

$$\begin{aligned} \boldsymbol{Z}_k^{\text{RW}} &= \boldsymbol{X}_k C^{\text{RW}} = [X_{t-P_m+1,k}, ..., X_{t,k}] C^{\text{RW}} \\ &= \left[\sum_{p=0}^{P_m-1} X_{t-p,k}, \sum_{p=0}^{P_m-2} X_{t-p,k}, ..., \sum_{p=0}^{1} X_{t-p,k}, X_{t,k} \right], \end{aligned} \tag{5}$$

so C^{RW} is summing up the lags. Now, to obtain a matrix of moving averages for MARX, we should divide each column by the number of lags included. In the case of one parameter, this corresponds to using

$$C^{\text{MA}} = \begin{bmatrix} \frac{1}{P_m} & 0 & 0 & \dots & 0 \\ \frac{1}{P_m} & \frac{1}{P_m-1} & 0 & & 0 \\ \frac{1}{P_m} & \frac{1}{P_m-1} & \frac{1}{P_m-2} & & 0 \\ \vdots & & & \ddots & \vdots \\ \frac{1}{P_m} & \frac{1}{P_m-1} & \frac{1}{P_m-2} & \dots & 1 \end{bmatrix} \qquad \theta^{\text{MA}} = \begin{bmatrix} P_m \beta_{P_m-1} \\ (P_m-1)(\beta_{P_m-2} - \beta_{P_m-1}) \\ (P_m-2)(\beta_{P_m-3} - \beta_{P_m-2}) \\ \vdots \\ \beta_0 - \beta_1 \end{bmatrix}, \tag{6}$$

which changes the regularization for equation (3) to

$$\lambda \boldsymbol{\theta'\theta} = \lambda \sum_{k=1}^{K} \sum_{p=1}^{P_m} (p(\beta_{p-1,k} - \beta_{p,k}))^2, \tag{7}$$

for general K and P_m. Just like in the case of RW, the lag coefficients are forced to be close to their

neighbours and make a smooth lag curve. But now, the penalty for a big difference is multiplied by the square of the lag. Hence, for lags further back in time, lag coefficients are even closer to their neighbours and the lag curve is more smooth.

Now that we have explained and interpreted MARX, the obtained matrix can be used for Z_t in the ML methods. In the following subsections, we will explain and interpret some variations on MARX that we have used.

4.2.3 EMARX

The first variation on MARX that we have used and would like to discuss is EMARX. The EMARX data transformation works the same as MARX, but uses an Exponential Moving Average (EMA) as the law of motion for the lag coefficients instead of MA. We have used the EMA given by Winters (1960), but only compute the average using the last P_m periods. This means that we obtain for a single variable the matrix

$$\boldsymbol{Z}_k^{\text{EMA}} = \left[\alpha \sum_{p=0}^{P_m-1} (1-\alpha)^p X_{t-p,k}, \alpha \sum_{p=0}^{P_m-2} (1-\alpha)^p X_{t-p,k}, ..., \alpha \sum_{p=0}^{1} (1-\alpha)^p X_{t-p,k}, \alpha X_{t,k} \right], \tag{8}$$

with $0 < \alpha < 1$ as the weighting parameter. For interpretation: the weighted average of the lagged values is computed, where the weights decline exponentially for further lags. This matrix $\boldsymbol{Z}$ corresponds to

$$C^{\text{EMA}} = \begin{bmatrix} \alpha(1-\alpha)^{P_m-1} & 0 & 0 & \dots & 0 \\ \alpha(1-\alpha)^{P_m-2} & \alpha(1-\alpha)^{P_m-2} & 0 & & 0 \\ \alpha(1-\alpha)^{P_m-3} & \alpha(1-\alpha)^{P_m-3} & \alpha(1-\alpha)^{P_m-3} & & 0 \\ \vdots & & & \ddots & \vdots \\ \alpha & \alpha & \alpha & \dots & \alpha \end{bmatrix} \tag{9}$$

$$\theta^{\text{EMA}} = \begin{bmatrix} \frac{\beta_{P_m-1}}{\alpha(1-\alpha)^{P_m-1}} \\ \frac{\beta_{P_m-2}}{\alpha(1-\alpha)^{P_m-2}} - \frac{\beta_{P_m-1}}{\alpha(1-\alpha)^{P_m-1}} \\ \frac{\beta_{P_m-3}}{\alpha(1-\alpha)^{P_m-3}} - \frac{\beta_{P_m-2}}{\alpha(1-\alpha)^{P_m-2}} \\ \vdots \\ \frac{\beta_0}{\alpha} - \frac{\beta_1}{\alpha(1-\alpha)} \end{bmatrix}, \tag{10}$$

resulting in a regularization of

$$\lambda \boldsymbol{\theta'\theta} = \lambda \sum_{k=1}^{K} \sum_{p=1}^{P_m} \left(\frac{\beta_{p-1,k}}{\alpha(1-\alpha)^{p-1}} - \frac{\beta_{p,k}}{\alpha(1-\alpha)^p} \right)^2, \tag{11}$$

for general P_m and K. This regularization can be interpreted as that the lag coefficient $\beta_{p,k}$ is shrinking towards a smaller factor of the preceding lag coefficient, namely $(1-\alpha)\beta_{p-1,k}$. Moreover,

the further the lag, the bigger the penalty for the difference between $\beta_{p,k}$ and $(1-\alpha)\beta_{p-1,k}$. For lag p, the penalty for the difference is multiplied by $\left(\frac{1}{\alpha(1-\alpha)^p}\right)^2$, which is increasing exponentially in p. For the mathematical proofs, see Appendix B. The choice for α is arbitrary, and there is no literature that claim to have found an optimal value for it. Therefore, we have chosen this ourselves, and decided to use $\alpha = 0.2$[1]. We tried to find an α that still gives a visible weight to the P_m'th lag, but a negligible weight to further lags, since we initially set these weights to zero. Since we use $P_m = 12$, we found that $\alpha = 0.2$ was suitable for this requirement.

4.2.4 FMARX and MARF

For the variations FMARX and MARF, we have tried to implement a combination of the two data transformations that have shown great results in past literature, namely MARX and PCA.

With FMARX, we first apply MARX to the matrix $\boldsymbol{X}$ as explained in Section 4.2.2, and obtain $\boldsymbol{Z}^{\text{MA}} = \boldsymbol{X}\boldsymbol{C}^{\text{MA}}$. The columns of this matrix now contain the moving averages of all variables, computed by using 1 to P_m lags. Afterwards, we apply PCA to this matrix by using the weighting matrix $W^{\mathbf{XC}}$ that contains the eigenvectors of $(\boldsymbol{X}\boldsymbol{C})'(\boldsymbol{X}\boldsymbol{C})$ as its columns. Just like any other PCA, we are able to reduce the dimensionality of $\boldsymbol{X}\boldsymbol{C}$ without losing too much information, by only keeping the first R columns of $W^{\mathbf{XC}}$. We obtain the principal component decomposition

$$FMARX = (\boldsymbol{X}\boldsymbol{C})W_R^{\mathbf{XC}}. \tag{12}$$

This matrix is applied in the ML algorithm, together with P_f lagged values.

For MARF, it is the other way around. First, we apply PCA and retain only R factors, to obtain the matrix F_R just as explained in Section 4.2.1. Then, the matrix containing F_R and its lagged values gets multiplied by $\boldsymbol{C}^{\text{MARF}} = I_R \otimes C^{\text{MA}}$. We now obtain a matrix of Moving Average Rotation of Factors (MARF), which for a single factor r can be written out as

$$MARF_r = \left[\frac{1}{P_f}\sum_{p=0}^{P_f-1} F_{t-p,r}, \frac{1}{P_f-1}\sum_{p=0}^{P_f-2} F_{t-p,r}, ..., \frac{1}{2}\sum_{p=0}^{1} F_{t-p,r}, F_{t,r}\right], \tag{13}$$

which is then used in the ML algorithm.

For both FMARX and MARF, there is no clear interpretation for the regularization resulting from these data transformations. The regularization is given as $\lambda\boldsymbol{\theta}'\boldsymbol{\theta} = \lambda\boldsymbol{\beta}'\boldsymbol{D}'\boldsymbol{D}\boldsymbol{\beta}$, where $\boldsymbol{D}$ is the inverse of the matrix that transforms $\boldsymbol{X}$. This transformation matrix is $\boldsymbol{C}W_R^{\mathbf{XC}}$ for FMARX and $W_R^{\text{X}}\boldsymbol{C}^{\text{MARF}}$ for MARF. But because of dimensionality reduction, these transformation matrices have more rows than columns, hence they are not invertible. So, there exists no matrix $\boldsymbol{D}$ for $R < K$, and the regularization cannot be interpreted. Even if $R = K$, the regularization would be a dot product of eigenvectors of $X'X$ and first difference lag coefficients, which cannot be interpreted nicely as well.

[1] Ideally, α would be set by cross-validation.

4.3 Recap

For the forecasting methods, we have used the Autoregressive (AR), Factor Model (FM) and the nonlinear ML algorithm Random Forest (RF). For RF, a number of data transformations is included in the analysis. We have used six transformations to obtain the matrix Z_t: (1) single-period differences and growth rates following McCracken & Ng (2016) (X_t and their lags), (2) principal components of X_t (F_t and their lags), (3) sets of simple moving averages of X_t ($MARX_t$), (4) sets of exponential moving averages of X_t ($EMARX_t$), (5) principal components of $MARX_t$ ($FMARX_t$ and their lags) and (6) sets of simple moving averages of F_t ($MARF_t$). For all these data transformations, lags of month-to-month log change of the original series to forecast are always included in Z_t.

5 Results

Table 1 shows the best RMSE data transformation for every target variable and horizon. It summarizes the main findings and shows important recommendations for future work in macroeconomic forecasting. The detailed RMSE results can be found in Appendix C.

Table 1: Best model specifications for every variable and horizon

	INDPRO	**EMP**	**UNRATE**	**INCOME**	**CONS**	**RETAIL**	**HOUST**	**M2**	**CPI**	**PPI**
H=1	EMARX	MARX	EMARX	FMARX	EMARX	FM	AR	EMARX	EMARX	MARX
H=3	EMARX	EMARX	EMARX	MARX	MARX	EMARX	MARF	F	FMARX	MARX
H=6	FMARX	EMARX	FMARX	FMARX	F	EMARX	MARF	AR	FMARX	X
H=9	FMARX	FMARX	FMARX	EMARX	F	EMARX	F	AR	FMARX	EMARX
H=12	FMARX	FMARX	FMARX	EMARX	F	X	F	AR	FM	AR
H=24	FMARX	X	F	X	FMARX	MARX	X	F	FMARX	FM

Note: The data transformations X, F, MARX, EMARX, FMARX and MARF have all been applied in the RF algorithm.

First thing to notice, is that for all targets and horizons, no less than 52 out of 60 times the nonlinear ML algorithm Random Forest outperforms the standard linear models FM and AR. This is in line with recent literature, concluding that nonlinear models are more useful in macroeconomic forecasting, like for example concluded by Maehashi & Shintani (2020). Second, using the non-standard macroeconomic data transformation MARX, found by Goulet Coulombe et al. (2021), has an advantage compared to the more standard transformations including X and F. This holds particularly for the target variables INDPRO, EMP, UNRATE, INCOME, CPI and PPI, which are the same variables where MARX produced good results in the paper of Goulet Coulombe et al. (2021).

5.1 New data transformations

What we were especially interested in, is the performance of the variations on MARX. MARX and variations on MARX were the best data transformation in combination with RF, in 39 out of 60 cases. Only 6 out of these 39 best RMSE results were obtained with the original MARX

that was found by Goulet Coulombe et al. (2021). This provides recommendations for practitioners in the field of macroeconomic forecasting, since we have not only shown that MARX is a data transformation with promising results, but also that there is enough room for improvement.

5.1.1 EMARX

First, as suggested by Goulet Coulombe (2020), choices for law of motions other than Moving Average could be a better fit when forecasting macroeconomic data. We have considered Exponential MA, following the promising results of Winters (1960) and Holt (2004). This new data transformation, EMARX, occurs to produce the best predictions in at least one forecasting horizon for every variable except HOUST. Even when it is not the best model specification, it often belongs to the top three. EMARX is not the most useful for a specific group of target variables or horizons, but seems to be spread out over Table 1. The success of EMARX suggests that weighted moving averages of macroeconomic variables, where the weights decline exponentially, should be included as regressors when forecasting macroeconomic data, rather than the original data. This is in line with Holt (2004), who suggests the use of EMA over simple MA.

5.1.2 FMARX and MARF

Other variations on MARX that we have analysed are FMARX and MARF, which both combine MARX with PCA. According to our results, FMARX is the best performing data transformation in combination with RF, when it comes to macroeconomic forecasting. With being the best model specification in 16 cases, it appears one time more often in Table 1 than EMARX. It seems to perform particularly well for the targets INDPRO, EMP, UNRATE, INCOME and CPI, which were also concluded by Goulet Coulombe et al. (2021) to be variables that are predicted well by using MARX. Moreover, FMARX shows to accomplish great forecasts for longer horizons in particular. One of the possible reasons why FMARX performs well, is that it is a combination of two methods that have shown great performance in previous literature, namely PCA and MARX. Intuitively, FMARX applies PCA on high dimensional data, in this case smoothed by MA. From this principal component decomposition, only a small part is retained, without losing much information. It seems like the combination of applying PCA on smoothed instead of raw data, and only keeping a small number of regressors that contain most information, produces a set of features with great forecasting performance. However, the order in which the two methods are applied can alter the results. For MARF, we applied MARX on factors obtained by PCA, which does not accomplish the same interesting results as FMARX. MARF is only the best data transformation in two cases, namely for HOUST at forecasting horizons 3 and 6. Apparently, smoothing factors using a moving average approach, does not provide any additional value when it comes to macroeconomic forecasting. Intuitively, PCA of smoothed data (FMARX) has a more clear interpretation than smoothed factors (MARF), which is a possible explanation why FMARX has a better performance. Nevertheless, MARF is not useless, since it does seem to perform well for the target HOUST, which is one of the targets in the paper by Goulet Coulombe et al. (2021) that does not have a consistently best model

specification.

6 Discussion

We understand that our research has a number of shortcomings, but also a lot of methods with promising results that could be analysed further. That is why would like to discuss some suggestions for further research in this section.

First, as one may have noticed, we do not provide any statistical tests when we compare our forecasting methods. We claim to have found great results and new methods that perform better than old methods in macroeconomic forecasting, but we are not able substantiate any of these claims by statistical tests. The reason for this, is that we only had limited time. In general, ML algorithms like RF are extremely time consuming, especially for our large amount of forecasting windows and regressors. A statistical test to compare forecasting methods would require a pairwise comparison for every predicted value, for every used method, horizon and variable. This would both be time and space consuming. Nevertheless, we understand that without any statistical tests, we cannot claim to have found significantly improved methods. Hence, for further research, we would still like to suggest the use of these tests. For example, the Diebold & Mariano (2002) test procedure could be used, which is also used by Goulet Coulombe et al. (2021). Model confidence set (MCS) by Hansen et al. (2011) could also be used, a set of models that contains the best model with a given level of confidence, which is easy to display and read.

Second, Goulet Coulombe et al. (2021) examine the use of another, unusual approach to construct the forecast for average growth/difference of the target variable. The usual approach, that we have also used in this paper, is that of fitting the model on the target variable h periods ahead directly, and using $\hat{y}_{t+h}^{\text{direct}}$ as prediction. The new approach forecasts every step separately, and uses the path average as its prediction, denoted by $\hat{y}_{t+h}^{\text{path-avg}}$. They conclude that for all variables which strongly co-move with the business cycle (INDPRO, EMP, UNRATE, INCOME, CONS), the path average approach produces better forecasts than the usual direct approach. However, we decided to not include this in our research, since this path average approach requires the estimation of h different functions for every forecast. We figured that this would take too much time, but we are still interested in whether this new approach would improve our forecasting methods. Especially for the before mentioned target variables, since our new data transformations EMARX and FMARX seem to be performing well for these variables. Furthermore, Goulet Coulombe et al. (2021) use a lot more data transformations and forecasting methods which can all be analysed further. We decided to limit our research to model specifications that we thought were the most promising. Just like in their paper, one could examine the marginal contribution of the different data pre-processing methods and apply them to a specific case.

Finally, we think that the new methods that have been analysed in this paper have shown some interesting results, and promise to be great performers in macroeconomic forecasting for research in the future. Therefore, we would like to suggest to investigate these methods in further research. For

example, following the suggestion from Goulet Coulombe (2020), we have used EMA as a new law of motion in MARX, but there are many more possibilities. For example, AR models of arbitrary order and RW with drifts could be implemented easily by changing matrix $\boldsymbol{C}$ in equation (3). A more complicated law of motion, often requires the estimation of parameters. This is also the case for our EMARX transformation, where we chose a value of $\alpha = 0.2$ for our weighting parameter. This choice was arbitrary, but could be optimised by using cross-validation. Optimising parameters can be time consuming, but may change the model completely and improve the forecasting results. Additionally, one could try a combination of the two data transformation that produced the best results for half of our variables and horizons, FMARX and EMARX. We think that the successes of these methods promise to make great forecasts when combined. We would suggests a transformation like FEMARX, where first EMARX is applied on the data, and then the dimensionality is reduced using PCA.

In summary, there are a lot of possibilities to investigate and improve the results of our new methods, and eventually find the best model specification for forecasting every macroeconomic variable.

7 Conclusion

In this paper, we have examined the macroeconomic forecasting performance of different data transformations. In particular, we have analysed the data transformation MARX, which was found by Goulet Coulombe et al. (2021), and tried to contribute to the field of macroeconomic forecasting by coming up with variations on this method. The main research question that we have attempted to answer is *"How do variations on MARX perform compared to already existing data transformations in macroeconomic forecasting?"*.

To answer our main question, we have analysed the large macroeconomic dataset FRED-MD from McCracken & Ng (2016). We applied our methods on 10 different target variables and 6 different forecasting horizons. We have used two simple, linear forecasting methods, namely the Autoregressive (AR) and Factor Model (FM), as benchmarks for our research. Additionally, we have used the nonlinear Machine Learning algorithm Random Forest (RF), which was fed with 6 different data transformations. The already existing data transformations that we have examined are single-period differences and growth rates following McCracken & Ng (2016), their principal components, and MARX that was found by Goulet Coulombe et al. (2021). Our first variation on MARX included an exponential moving average (EMA) as the law of motion, called EMARX. The last two variations were a combination of PCA and MARX, called FMARX and MARF.

In general, MARX or a variation of MARX shows the best forecasting performance for no less than 39 out of 60 cases. This remarkable success of MARX is in line with the results of Goulet Coulombe et al. (2021). What is even more noteworthy, is the fact that our new variations of MARX mostly outperform the original method. EMARX and FMARX together are the best model specification in more than half of all cases. The success of EMARX suggests that the

use of EMA as the law of motion is a better choice than simple moving average (MA). This can be explained by the fact that EMA is preferred over MA when smoothing macroeconomic data, following Holt (2004). FMARX applies dimensionality reduction using PCA on smoothed data that is obtained by MARX, and the results show that this method produces the best forecast for a certain group of variables, especially for longer horizons. FMARX outperforming factors of the raw data, suggests that forecasters should prefer applying PCA on smoothed data. Moreover, using reduced dimensionality helps obtaining better forecasts, since FMARX also often outperforms MARX. However, the data transformation MARF, where MARX is applied on the factors of the data, does not produce a set of regressors with interesting forecasting ability. Apparently, obtaining factors of smoothed data works well in macroeconomic forecasting, while smoothed factors do not.

To answer our research question, we conclude that certain variations on MARX outperform already existing data transformations in macroeconomic forecasting. Like we discussed in Section 6, our research has its shortcomings and there is a lot of room for improvement. Nevertheless, our paper contributes to the field by showing the remarkable performance of new data transformations, and we expect it to help in the search for the best model specification when forecasting macroeconomic variables.

References

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

Breitung, J., & Eickmeier, S. (2006). Dynamic factor models. *Allgemeines Statistisches Archiv*, *90*(1), 27–42.

Diebold, F. X., & Mariano, R. S. (2002). Comparing predictive accuracy. *Journal of Business & economic statistics*, *20*(1), 134–144.

Goulet Colombe, P., Marcellino, M., & Stevanović, D. (2021). Can machine learning catch the covid-19 recession? *National Institute Economic Review*, *256*, 71–109.

Goulet Coulombe, P. (2020). Time-varying parameters as ridge regressions. *arXiv preprint arXiv:2009.00401*.

Goulet Coulombe, P., Leroux, M., Stevanovic, D., & Surprenant, S. (2019). How is machine learning useful for macroeconomic forecasting? *Journal of Applied Econometrics*.

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

Hall, A. S. (2018). Machine learning approaches to macroeconomic forecasting. *The Federal Reserve Bank of Kansas City Economic Review*, *103*(63), 2.

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

Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, *20*(1), 5–10.

Kim, H. H., & Swanson, N. R. (2018). Mining big data using parsimonious factor, machine learning, variable selection and shrinkage methods. *International Journal of Forecasting*, *34*(2), 339–354.

Ku, W., Storer, R. H., & Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and intelligent laboratory systems*, *30*(1), 179–196.

Lee, T.-H., Ullah, A., & Wang, R. (2020). Bootstrap aggregating and random forest. In *Macroeconomic forecasting in the era of big data* (pp. 389–429). Springer.

Maehashi, K., & Shintani, M. (2020). Macroeconomic forecasting using factor models and machine learning: an application to japan. *Journal of the Japanese and International Economies*, *58*, 101104.

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

Medeiros, M. C., Vasconcelos, G. F., Veiga, Á., & Zilberman, E. (2021). Forecasting inflation in a data-rich environment: the benefits of machine learning methods. *Journal of Business & Economic Statistics*, *39*(1), 98–119.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, *2*(11), 559–572.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Shiller, R. J. (1973). A distributed lag estimator derived from smoothness priors. *Econometrica: journal of the Econometric Society*, 775–788.

Stock, J. H., & Watson, M. W. (2002). Forecasting using principal components from a large number of predictors. *Journal of the American statistical association*, *97*(460), 1167–1179.

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management science*, *6*(3), 324–342.

Wolff, C. C. (1987). Time-varying parameters and the out-of-sample forecasting performance of structural exchange rate models. *Journal of Business & Economic Statistics*, *5*(1), 87–97.

Yoon, J. (2021). Forecasting of real gdp growth using machine learning models: Gradient boosting and random forest approach. *Computational Economics*, *57*(1), 247–265.

A Appendix - Random Forest in Detail

The Random Forest algorithm provides a method to approximating nonlinear functions by combining regression trees. The algorithm was introduced by Breiman (2001). RF grows many trees on random subsamples of observations, and are splitting for a random subset of features. The final forecast obtained by the mean of the forecasts of all trees. For a large number of regressors, RF can have a high computation time. Some parameters are chosen to reduce this running time. The full algorithm is shown below.

Algorithm 1 Random Forest

for b **in** 1:200 **do**
    Randomly draw a subsample $(Z^{(b)}, y^{(b)})$ of size $N/2$ from training sample (Z, y)
    Grow regression tree T_b on $(Z^{(b)}, y^{(b)})$:
    while terminal node size $n > 5$ **do**
        Randomly draw $\#Z/3$ regressors from $Z^{(b)}$
        Find the variable-threshold pair minimizing MSE in daughter regions
        (Prediction in daughter regions is mean of $y^{(b)}$ in said region)
        Repeat for each terminal node and split accordingly
    end while
    Compute $c_m^{(b)} = avg\{y^{(b)} | Z_t^{(b)} \in R_m^{(b)}\}$ where $R_m^{(b)}$ is region m of tree T_b
end for
Define prediction of Tree T_b with M nodes: $f(Z_t, T_b) = \sum_{m=1}^{M} c_m^{(b)} I(Z_t \in R_m)$
Return prediction $\frac{1}{200} \sum_{b=1}^{200} f(Z_t, T_b)$

We have not optimised any hyperparameters for RF. The only difference with the parameters used by Goulet Coulombe et al. (2021), is that our randomly drawn subsample is of size $N/2$ instead of N, to reduce running time. For this algorithm we use a standard value for the lag orders and number of factors, namely $(P_y, P_f, r) = (12, 12, 8)$. The large number of lags and factors do not mean our method is overfitting, since RF averages over "randomized trees".

B Appendix - Mathematical Proofs

B.1 Proof EMARX properties

We rewrite the regularization in the Ridge regression in the case of EMARX, given in equation 11, as

$$\begin{aligned}\lambda\boldsymbol{\theta}'\boldsymbol{\theta} &= \lambda\sum_{k=1}^{K}\sum_{p=1}^{P_m}\left(\frac{\beta_{p-1,k}}{\alpha(1-\alpha)^{p-1}} - \frac{\beta_{p,k}}{\alpha(1-\alpha)^{p}}\right)^2 \\ &= \lambda\sum_{k=1}^{K}\sum_{p=1}^{P_m}\left(\frac{(1-\alpha)\beta_{p-1,k} - \beta_{p,k}}{\alpha(1-\alpha)^{p}}\right)^2 \\ &= \lambda\sum_{k=1}^{K}\sum_{p=1}^{P_m}\left(\frac{1}{\alpha(1-\alpha)^{p}}\right)^2 ((1-\alpha)\beta_{p-1,k} - \beta_{p,k})^2\end{aligned}$$

which is minimized. That means that the term after the summation is shrinking towards zero for every p and k. To find what the lag $\beta_{p,k}$ is shrinking towards, we solve

$$\begin{aligned}\left(\frac{1}{\alpha(1-\alpha)^{p}}\right)^2 ((1-\alpha)\beta_{p-1,k} - \beta_{p,k})^2 &= 0 \\ (1-\alpha)\beta_{p-1,k} - \beta_{p,k} &= 0 \\ (1-\alpha)\beta_{p-1,k} &= \beta_{p,k},\end{aligned}$$

and find that $\beta_{p,k}$ is shrinking towards a smaller factor of the preceding lag, namely $(1-\alpha)\beta_{p-1,k}$. The factor with which the penalty of the difference between these is multiplied, can be seen directly in the rewritten regularization, namely $\left(\frac{1}{\alpha(1-\alpha)^{p}}\right)^2$ for every lag p.

C Appendix - Detailed Relative RMSE Results

Table 2: Relative RMSE - H=1,3,6

H = 1

	INDPRO	**EMP**	**UNRATE**	**INCOME**	**CONS**	**RETAIL**	**HOUST**	**M2**	**CPI**	**PPI**
Benchmarks										
FM (RMSE)	0.007	0.001	0.162	0.007	0.005	0.011	0.077	0.003	0.002	0.006
AR	0.98	0.95	1.00	1.02	1.00	1.01	0.99	0.99	1.08	1.02
Random Forest										
F	0.95	0.95	0.94	0.93	1.01	1.01	1.02	0.95	1.06	1.05
X	0.96	0.96	0.92	0.94	1.01	1.01	1.03	0.94	1.04	1.02
MARX	0.89	0.87	0.91	0.95	1.00	1.01	1.03	0.95	1.01	0.99
EMARX	0.88	0.88	0.90	0.93	0.99	1.00	1.02	0.94	0.99	0.99
FMARX	0.95	0.91	0.94	0.92	1.02	1.02	1.03	0.98	1.06	1.03
MARF	0.92	0.92	0.92	0.95	1.01	1.04	1.01	0.95	1.04	1.02

H = 3

	INDPRO	**EMP**	**UNRATE**	**INCOME**	**CONS**	**RETAIL**	**HOUST**	**M2**	**CPI**	**PPI**
Benchmarks										
FM (RMSE)	0.005	0.001	0.103	0.004	0.003	0.006	0.037	0.003	0.002	0.004
AR	0.95	0.92	0.95	1.01	0.91	0.99	0.94	0.98	0.99	1.03
Random Forest										
F	0.99	0.98	0.93	0.96	0.84	1.02	0.92	0.93	0.96	1.08
X	1.03	1.03	0.91	0.93	0.85	0.98	0.92	0.97	0.95	1.01
MARX	0.95	0.85	0.82	0.90	0.83	0.99	0.92	0.96	0.96	1.00
EMARX	0.90	0.84	0.80	0.91	0.84	0.98	0.92	0.96	0.96	1.00
FMARX	0.94	0.89	0.88	0.91	0.89	0.99	0.94	0.99	0.92	1.02
MARF	0.96	0.91	0.86	0.94	0.85	1.04	0.90	0.98	0.95	1.07

H = 6

	INDPRO	**EMP**	**UNRATE**	**INCOME**	**CONS**	**RETAIL**	**HOUST**	**M2**	**CPI**	**PPI**
Benchmarks										
FM (RMSE)	0.004	0.001	0.086	0.002	0.002	0.004	0.026	0.002	0.002	0.003
AR	0.95	0.91	1.00	1.00	0.91	0.97	1.00	0.96	0.99	1.01
Random Forest										
F	1.01	0.99	1.04	1.04	0.89	1.11	0.95	0.96	0.93	1.09
X	0.99	1.01	0.99	0.98	0.90	0.98	0.96	1.02	0.93	0.99
MARX	0.98	0.87	0.94	0.93	0.90	1.01	0.95	1.03	1.00	1.01
EMARX	0.97	0.81	0.94	0.93	0.91	0.95	0.95	1.02	1.00	1.00
FMARX	0.89	0.86	0.91	0.91	0.93	0.96	0.98	1.03	0.89	0.99
MARF	1.06	0.92	0.95	0.99	0.90	1.14	0.92	1.08	0.96	1.11

Table 3: Relative RMSE - H=9,12,24

H = 9

	INDPRO	EMP	UNRATE	INCOME	CONS	RETAIL	HOUST	M2	CPI	PPI
Benchmarks										
FM (RMSE)	0.004	0.001	0.083	0.002	0.002	0.004	0.022	0.002	0.002	0.003
AR	0.91	0.89	1.05	1.03	0.87	0.96	0.96	0.96	0.99	0.99
Random Forest										
F	0.94	0.92	1.01	1.06	0.86	1.08	0.88	1.00	0.97	1.05
X	0.90	0.92	0.96	0.96	0.88	0.93	0.93	1.08	0.99	0.99
MARX	0.90	0.83	0.95	0.91	0.90	0.93	0.92	1.09	1.08	1.00
EMARX	0.91	0.82	0.98	0.91	0.90	0.89	0.92	1.10	1.07	0.97
FMARX	0.82	0.80	0.93	0.91	0.91	0.93	0.95	1.07	0.96	0.98
MARF	1.03	0.88	0.95	1.00	0.87	1.12	0.88	1.17	1.07	1.13

H = 12

	INDPRO	EMP	UNRATE	INCOME	CONS	RETAIL	HOUST	M2	CPI	PPI
Benchmarks										
FM (RMSE)	0.004	0.001	0.087	0.002	0.002	0.003	0.020	0.002	0.001	0.003
AR	0.91	0.91	1.00	1.00	0.89	0.94	0.94	0.94	1.00	0.98
Random Forest										
F	0.90	0.88	0.92	1.04	0.86	1.00	0.83	1.00	1.06	1.05
X	0.87	0.88	0.86	0.92	0.88	0.89	0.90	1.09	1.07	1.01
MARX	0.86	0.83	0.86	0.88	0.90	0.91	0.92	1.09	1.12	1.01
EMARX	0.87	0.83	0.88	0.88	0.89	0.89	0.92	1.12	1.13	0.99
FMARX	0.80	0.79	0.86	0.88	0.89	0.91	0.95	1.07	1.01	0.99
MARF	1.02	0.87	0.92	1.02	0.86	1.05	0.88	1.18	1.23	1.13

H = 24

	INDPRO	EMP	UNRATE	INCOME	CONS	RETAIL	HOUST	M2	CPI	PPI
Benchmarks										
FM (RMSE)	0.003	0.001	0.067	0.002	0.002	0.003	0.015	0.002	0.002	0.002
AR	0.93	1.04	1.11	0.97	0.93	0.92	1.00	0.95	1.07	1.03
Random Forest										
F	0.78	0.84	0.88	0.96	0.88	0.85	0.78	0.94	1.05	1.09
X	0.77	0.83	0.90	0.90	0.89	0.83	0.73	1.00	1.05	1.05
MARX	0.84	0.86	0.90	0.94	0.92	0.81	0.84	1.05	1.10	1.17
EMARX	0.80	0.87	0.94	0.93	0.93	0.84	0.87	1.07	1.07	1.17
FMARX	0.76	0.86	0.97	0.91	0.85	0.93	0.91	1.03	0.93	1.05
MARF	0.81	0.88	0.89	1.01	0.92	0.88	0.83	1.09	1.28	1.17