A hybrid approach of extreme gradient boosting and random forest to nowcast US GDP

by

Rogier van de Kamp

Student ID: 426029

Thesis supervisor: Dr. A. Pick

Second assessor: Dr. A.A. Naghi

Master Thesis of the

Master Econometrics and Management Science

at the

Erasmus School of Economics

Erasmus Universiteit Rotterdam

The views stated in this thesis are those of the author and not necessarily those of Erasmus School of Economics or Erasmus University Rotterdam.

Date: May 18, 2022

Abstract

This paper will introduce a Hybrid Approach for nowcasting. The Hybrid Approach is a combination of Random Forest and Extreme Gradient Boosting. The nowcast of the Hybrid Approach will be compared to the Dynamic Factor Model and the sole use of both Random Forest and Extreme Gradient Boosting. For this comparison, four different variations of a combined data set from the FRED-MD and FRED-QD are used. The investigation of which method works best will be done by looking at the MSE followed by the usage of variables. This paper will show that the proposed Hybrid Approach will result in a lower MSE for nowcasting than the benchmark of Dynamic Factor Model and both individual methods of Random Forest and Extreme Gradient Boosting. The lower MSE will result from a more accurate variable selection in the technique. The Hybrid Approach seems like a good technique to be explored in future research.

Contents

1 Introduction

In this research machine learning techniques will be used to investigate if nowcasts can be improved. Nowcasting is done by most central banks with the use of Dynamic Factor Models. Bok et al. (2017) showed us how the New York FED uses the Dynamic Factor Model for their nowcasts. Bok et al. (2017) create multiple nowcasts each month since there is a delay in the release of certain variables. Since the computing power is now increasing more powerful nowcasts techniques can be considered using machine learning.

In this research, three machine learning methods will be compared to the Dynamic Factor Model. Richardson et al. (2021) already showed us that machine learning techniques do outperform simple AR models for nowcasting. The machine learning techniques that will be considered are Random Forest, Extreme Gradient Boosting and a Hybrid Approach consisting of these two methods. Yoon (2021) already discovered that Random Forest and Extreme Gradient Boosting can have more accurate nowcasts than a Dynamic Factor Model. However, Yoon (2021) only uses a small set of variables which limits the possibilities of the different models. Jansen et al. (2016) showed that a Dynamic Factor Model with a lot of variables does outperform other traditional econometric methods for nowcasting. Therefore it should be interesting to do this evaluation of the Dynamic Factor Model with large vintages of the data.

The vintage of the data that will be used in this paper is a combination of the FRED-MD and FRED-QD from McCracken and Ng (2016) and McCracken and Ng (2020) respectively. To get a more thorough look in the machine learning methods four different variations of the vintage of the data will be used. First of all, these vintages of the data will be downloaded with a starting date of 1980 but since not all the variables were available in the 1980s and 1990s this paper has also chosen to train the models with a starting date of 2000. Both these alterations will have a version were the variables of the previous quarter, so-called lag variables, will be included and one where only the variables of the quarter itself will be used. A Dynamic Factor Model automatically uses lag variables since it uses previous observation to predict the new observation, so for this technique there will not be any difference in the performance. Using both with and without lag variables will be done because the vintage of the data already contains over 400 variables without lag variables included using too many variables could be harmful. All these vintages of the data will be downloaded at the 1st of January 2017 until the 1st of January 2020 with intervals for the 1st and 15th day of each month.

Having too many variables can have a negative effect for the prediction of a Random Forest, that is why Speiser et al. (2019) explain methods to reduce variables for a Random Forest by using the Random Forest itself. In which the methods of Jiang et al. (2004) and Genuer et al. (2015) seemed to work best. However, this has been compared in a discrete setting. While this paper works in a continuous setting. Soybilgen and Yazgan (2021) did nowcasting with a random forest on a reduced data set. For this it used the factors of their Dynamic Factor Model. Extreme Gradient Boosting does not use a lot of variables because of its penalization term in the objective function. This penalization term can also be used to reduce the variables. Therefore, this paper introduces a Hybrid Approach. This Approach uses only the variables that Extreme Gradient Boosting selects and put these in a Random Forest. To get the best input parameters for each model a cross validation will be done for each vintage of the data.

The results of the algorithms show great promise for the Hybrid Approach since it performs best for each of the four different variations of the data. It really shows a lot of promise with a longer data set since the variance of the squared error is low as well in that case. The percentage of variables used shows that Random Forest uses more different kind of variables than Extreme Gradient Boosting and that the Hybrid Approach is in between these two techniques.

2 Data

For this paper a combination of the FRED-MD and FRED-QD data sets are being used. These data sets are explained in McCracken and Ng (2016) and McCracken and Ng (2020) respectively. When variables are in both the FRED-QD and FRED-MD only the variable in the FRED-MD will be selected. In the FRED-QD McCracken and Ng (2020) decided to use the average of the three months when monthly variables are available. In this paper the value of the first month in the quarter will be used as input for the transformation. Since the first months' data is soonest available, it will be more accurate for the nowcast. The same holds true for daily variables where the first day of the month will be selected to use as input for the transformation. All the variables that have been used and their transformation can be found in Appendix A. All the vintages of the data are downloaded with a starting date of 1980-01-01. Since this is real time data some values will be added during the month and therefore data is downloaded twice a month, on the first and 15th of every month. Since not every variable was defined during the 1980s and 1990s it is chosen to train the models with a starting date of 1980-01-01 as well as a starting date of 2000-01-01. The vintages of the data are downloaded between 2017-01-01 until 2020-01-01 which results in two vintages of the data for each month (first and 15th of the month) for 3 years (2017, 2018, 2019) plus one vintage of the data on 2020-01-01. Besides the different starting dates, another variation of the data set is created where the values of the previous quarter will be included as lag variables. The aforementioned choices result in four different data sets for each downloaded vintage of the data. Where the two data sets without lag variables have 467 variables and the two data sets with lag variables have 934 variables.

3 Methodology

The FRED-MD consists mostly of monthly data and the FRED-QD of quarterly data. Combining these data sets results in a mixed frequency data set. Most machine learning techniques are not able to handle a mixed frequency data set. Therefore the monthly variables will be used to create three new variables for the quarter. For example if you have the monthly variable IPMAT the quarterly variables will be IPMATM1 for the first month, IPMATM2 for the second month and IPMATM3 for the third month in the quarter.

In this research there are 73 vintages of the data. Each vintage of the data will nowcast the GDP of the quarter in which the vintage of the data is downloaded. Since the vintages of the data are downloaded twice a month, each quarter has six nowcasts.

3.1 Dynamic Factor Model

For the Dynamic Factor Model, the same notation and program as Bok et al. (2017) will be used. A Dynamic Factor Model can be seen as a regression as shown in equation 1. the equation is a regression of the data $y_{i,t}$, on the common factors $f_{1,t}, \ldots, f_{r,t}$ and $e_{i,t}$. With $e_{i,t}$ being the idiosyncratic movement of the individual variables. In this paper only one common factor will be used.

$$y_{i,t} = \lambda_{i,1} f_{1,t} + \cdots + \lambda_{i,r} f_{r,t} + e_{i,t}, \quad \text{for } i = 1, \ldots, n \tag{1}$$

To use the Kalman smoother, the common factors and the idiosyncratic component are considered to be following a Gaussian autoregressive processes as shown in equations 2 and 3.

$$f_{j,t} = a_j f_{j,t-1} + u_{j,t}, \quad u_{j,t} \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2_{u_j}) \quad \text{for } j = 1, \ldots, r \tag{2}$$

$$e_{i,t} = p_i e_{i,t-1} + \epsilon_{i,t}, \quad \epsilon_{i,t} \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2_{\epsilon_i}) \quad \text{for } i = 1, \ldots, n \tag{3}$$

Together the equations 1, 2 and 3 form a state space model. With equation 1 as the measurement equation and equations 2 and 3 as the transition equations.

To estimate the Dynamic Factor Model the EM algorithm with a Kalman smoother will be used. For the initialization of the algorithm, principal components are calculated. These principal components usually give a reliable estimation of the unobserved common factors and therefore should work as a good starting point. In the first step of the algorithm the model parameters are estimated via an OLS regression using the common factors. In the second step the common factors are updated via the Kalman smoother. To get a maximum likelihood estimation these two steps are repeated until convergence.

3.2 Decision Tree

A Decision Tree is a commonly used Machine Learning method and is used for Random Forest and Extreme Gradient Boosting. The most common implementation of Decision Tree is the one from Pedregosa et al. (2011). A Decision Tree consists of various nodes and leafs. The beginning of a tree is just one single node. In this node a variable is used to split the data creating two new nodes. The split in the data is based on different criteria. In this paper the mean squared error (MSE) criteria will determine the split. The MSE is calculated for each variable in the dataset, from which the best variable for a split is chosen. The splitting of the data continues until a stopping criteria is met. When this criteria is met the node that reached the criteria will be considered a leaf. The average of all the observations in the leaf will be the value associated with the leaf.

A Decision Tree will not work when there are missing values because the splitting criterion for each variable has to be calculated. However, this paper uses vintages of the data, which are real time data sets. Not all the observations for each variable are known at a certain time. To deal with this issue data can be imputed. There are different ways to impute the data. In this paper the missing data will be imputed by using the smoothed series from the Dynamic Factor Model as explained in section 3.1.

3.2.1 Hyperparameters Decision Tree

When optimizing a Decision Tree there are several input values which can be optimized. The input values are also known as hyperparameters. Since a Decision Tree tend to over-fit the data, the hyperparameters should be chosen in such a way that it counters over-fitting. To counter over-fitting two hyperparameters are chosen in this paper. The chosen hyperparameters are the maximum depth of the tree and the minimum samples per leaf. When a maximum depth is set for the tree, it means that the amount of nodes and leaves in the tree are limited. For example when having a maximum depth of 2 the tree can at most have 3 nodes and 4 leaves, one node in the first layer and two nodes in the second, which will result in at most 4 leaves. By restricting the amount of nodes in the tree the tree cannot continue growing until each observation has its own leaf. The second chosen hyperparameter tells the tree how many samples are needed in each node to be allowed to split. The effect of this hyperparameter could be that the outliers in the data set will be mixed with non-outlier observations and will therefore have a significant influence on the result of the respective leaf.

3.3 Random Forest

A Random Forest is a combination of multiple Decision Trees. The Decision Trees in a Random Forest are created by using only a few variables from the entire data set. The variables for each tree are randomly selected which results in different trees in the forest. The prediction of the Random Forest will be the average outcome of all Decision Trees. The algorithm that will be used in this paper is the algorithm of Pedregosa et al. (2011).

Because this paper is using vintages of the data which are real time data sets, not all variables are available. To deal with this ragged edge issue, the smoothed series from the Dynamic Factor Model, as explained in section 3.1, will be used. Since the Dynamic Factor Model has a prediction for all the variables this could be a good solution.

As mentioned in section 3 there are 73 vintages of the data in this research. Each vintage of the data will nowcast the GDP of the quarter in which the vintage of the data is downloaded. Since the vintages of the data are downloaded twice a month, the first and 15th of each month, each quarter has six nowcasts.

3.3.1 Hyperparameters Random Forest

Besides the hyperparameters explained in section 3.2.1 there are other hyperparameters a Random Forest can use to determine how much it should over-fit. Three hyperparameters will be highlighted. First of all, there is the number of estimators. The number of estimators regulates the number of trees that will be created in the forest. More trees in the forest will result in more possible combinations of variables.

Secondly, there is a maximum number of variables parameter. This parameter tells the algorithm how many variables it can include in each creation of a tree. Having a small amount of variables in each tree reduces the chance of having used all the variables in the forest. Another effect of using a small amount of variables in each tree is that there is a higher variance between the trees. This could cause the trees to over-fit but the forest will not. This approach can be seen as column sub-sampling. The most common value to pick for the maximum number of variables parameter is the square root of the total amount of variables. Therefore, in this paper the square root of the total amount of variables will be used.

Another hyperparameter that could be used is selecting a maximum number of observations for each tree to train on, also known as row sub-sampling. Since there are not that many observations in total, this option will not be further explored in this paper.

3.4 Extreme Gradient Boosting

The boosting algorithm that will be used in this paper is the algorithm from Chen and Guestrin (2016). Boosting is a method which combines models, who normally do not have an high performance when there is a bias or variance in the data, by training new models more heavenly on the wrongly predicted outcome of the previous model. In the algorithm of Chen and Guestrin (2016) the used models are Decision Trees. Equation 4 shows how the combination of different models generally is used to calculate the outcome of the regression, with f_k being the k^{th} tree.

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), \qquad f_k \in \mathcal{F} \tag{4}$$

In each boosting round the amount of trees will grow with the initial amount of trees. To reduce the complexity of the algorithm there has been chosen to start with 1 tree. Each subsequent boosting round tries to explain more of the variance or the bias of the data. The objective of the algorithm is to minimize the regularized object shown in equation 5. Where $l(\hat{y}_i, y_i)$ needs to be a differential convex loss function and Ω penalizes the complexity of the model. Where T is the amount of leaves in a tree and $||w||^2$ is the square root of the sum of the squared weights also known as the L2-norm. γ is a parameter which can be set to different values to increase the penalization of more leaves in a tree. λ is a parameter which can be set to different values to get a more smooth prediction to reduce over-fitting.

$$\begin{gathered}\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \\ \text{where } \Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2\end{gathered} \tag{5}$$

In this paper the convex loss function is the squared error between the predicted $\hat{y}_i$ and the true value y_i.

Equation 5 has functions as parameters which cannot be optimized in Euclidean space via traditional optimization methods. Therefore the model will be trained in an additive manner. Which will finally result in the optimal value shown in equation 6. Where I_j is the set of leaf j and g_i and h_i are explained in equation 7 & 8 respectively. With the optimal weight of leaf j shown in equation 9. For the complete derivation please look at Chen and Guestrin (2016).

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{6}$$

$$g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}) \tag{7}$$

$$h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}) \tag{8}$$

$$w_j^* = -\frac{(\sum_{i \in I_j} g_i)}{\sum_{i \in I_j} h_i + \lambda} \tag{9}$$

In tree based methods it is important to find the best possible split. The Extreme Gradient Boosting algorithm uses a different formula for finding the best split than the Decision Tree as explained in section 3.2. The Extreme Gradient Boosting algorithm uses the formula in equation 10 to calculate the values for the split. To find the best possible split two different methods can be considered, an exact greedy algorithm and an approximate algorithm. Building multiple trees in each boosting round is very computational demanding. Therefore, it can be very helpful to use the approximate algorithm. Since only one tree is used in each boosting round the exact greedy algorithm is chosen.

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda}\right] - \gamma \tag{10}$$

Because this paper is using vintages of the data which are real time datasets, not all variables are available. To deal with this ragged edge issue, the smoothed series from the Dynamic Factor Model, as explained in section 3.1, will be used. Since the Dynamic Factor Model has a prediction for all the variables this could be a good solution.

As mentioned in section 3 there are 73 vintages of the data in this research. Each vintage of the data will nowcast the GDP of the quarter in which the vintage of the data is downloaded. Since the vintages of the data are downloaded twice a month, the first and 15th of each month, each quarter has six nowcasts.

3.4.1 Hyperparameters Extreme Gradient Boosting

The Extreme Gradient Boosting algorithm also has some extra hyperparameters which can be implemented to reduce over-fitting. From the hyperparameters discussed in section 3.2.1, one will be used in this algorithm, this is the maximum depth of the tree. Besides the maximum depth of the tree, the number of estimators can be defined. This is number K from equation 4. The number of estimators tells how many boosting rounds will be used, which consequentially shows how many trees will be constructed. A higher number of boosting rounds means a more precise prediction in the training data, which means more over-fitting. The other hyperparameter which can be changed to reduce over-fitting is the learning rate. The learning rate can also be seen as a shrinkage factor. The learning rate shrinks the newly added weights after each boosting round. That means that a lower learning rate will increase the amount of boosting rounds necessary to fit the training data perfectly. Therefore, it reduces the chance of over-fitting.

3.5 Hybrid approach

A Random Forest only has a choice of a certain amount of variables for each tree. If all variables that are randomly chosen for a tree do not contain much information, the tree will not have a good prediction. Since this paper makes use of a big data set, this could happen in multiple occasions. That is why, this paper proposes to combine Extreme Gradient Boosting and Random Forest.

The penalization term of the objective function of Extreme Gradient Boosting is shown in 11 where T is the amount of leaves in a tree and $||w||^2$ is the square root of the sum of the squared weights also known as the L2-norm.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2 \tag{11}$$

This penalization therm shows that a simpler model with less variables is preferred. The first term of the penalization of the objective function restrict the amount of variables that are being used. This is because having less leaves means less possibilities for the data to split. Combining this with the second term which is shrinking the individual weights to 0 only a small amount of variables will be chosen. To make sure only a small amount will be chosen the γ and λ parameters can be given a higher value. The increase in the γ value will have the most effect on the decrease of variables.

When the Extreme Gradient Boosting algorithm is trained, it uses a small amount of variables. These variables will than be used to train a Random Forest. The Random Forest will be trained in a similar matter as mentioned earlier in Section 3.3.

Because this paper is using vintages of the data which are real time data sets, not all variables are available. To deal with this ragged edge issue, the smoothed series from the Dynamic Factor Model, as explained in section 3.1, will be used. Since the Dynamic Factor Model has a prediction for all the variables this could be a good solution.

As mentioned in section 3 there are 73 vintages of the data in this research. Each vintage of the data will nowcast the GDP of the quarter in which the vintage of the data is downloaded. Since the vintages of the data are downloaded twice a month, the first and 15th of each month, each quarter has six nowcasts.

3.6 Cross Validation

To get the best results from your machine learning algorithm, the right hyperparameters need to be chosen. Since it is difficult to know which values are the best, cross validation will be used.

A normal cross validation consists of a couple of steps. First of all the data needs to be split in N separate groups. In this paper, $N = 10$. The second step is to train the machine learning algorithm on $N - 1$ groups and get the test value on the last remaining group. The algorithm will be trained with all the different combinations of the hyperparameter input. However, since this paper is working with time sensitive data a normal cross validation could include a look-ahead bias. Therefore an expanding window will be used, the first training data only contains the first $N + 1$ part of the observations with the test data being the second $N + 1$ part of the observations, the second training data contains the first two $N + 1$ parts of the observations and the second test data contains the third $N + 1$ part of the observation, etcetera. Thirdly, the test score values will be calculated for every given hyperparameter combination in each trained window on a given test set. To calculate the test score the Coefficient of Determination will be used. The Coefficient of Determination, well known as the R^2, is shown in equation 12. Where $\hat{Y_i}$ is the predicted value for Y_i and $\bar{Y}$ is the average of Y.

$$R^2 = \left(1 - \frac{\sum_{i=0}^{N-1}(Y_i - \hat{Y_i})^2}{\sum_{i=0}^{N-1}(Y_i - \bar{Y})^2}\right) \tag{12}$$

The Coefficient of Determination can be seen as a comparison between the forecast of the model and the average of the true values. Finally, the average of all these test score will be calculated and the highest one will be chosen as the optimal value.

Table 1 shows the hyperparameter values that are chosen for this cross validation. Since there is not an infinite amount of time available to do all the cross validations, some choices have been made on which values to pick to optimize. The standard values given by Chen and Guestrin (2016) are a maximum depth of 6, number of Boosts of 100 and a learning rate of 0.3. Since there are not that many observations it has been chosen for the maximum depth to be less than 6. Since the maximum depth of the tree is so low it is not necessary to also add the minimal sample per leaf hyperparameter, therefore this one is omitted from the cross validation. The number of Boosts is a difficult hyperparameter to consider. Normally, more boosting rounds with a not so deep tree really improves the algorithm. However, since this is a variable which can over-fit the algorithm, lower and higher values considered. The learning rate has an almost direct influence on the number of Boosts. A higher learning rate will make the algorithm converge faster, since there is less shrinkage on the newly added boosting round. Because the number of Boosts has such a large variety the learning rate also needs to have some variety to make sure the optimal combination can be chosen.

Pedregosa et al. (2011) use a standard value for the number of Trees of 100. Since there are a lot of variables in the data sets a high value for the number of Trees will give a bigger chance to include the most important variables at least once. Thus a high value could improve the forest. The maximum depth and the minimal sample per leaf of each tree in the forest reduces the risk of over-fitting each tree. However, if a tree over-fits it is not that big of a problem. Therefore, the maximum depth and minimal sample per leaf are chosen to include quite a wide range.

Table 1: Hyperparameters that are being tested in the cross validation for a Random Forest and Extreme Gradient Boosting

Extreme Gradient Boosting			**Random Forest**		
Max. Depth	**Nr. of Boosts**	**Learning rate**	**Max. Depth**	**Nr. of Trees**	**Min. Sample Leaf**
1	10	0.1	1	100	1
2	30	0.3	2	200	2
3	50	0.5	3	300	3
4	70	0.7	4	400	4
5	90	0.9	5	500	5
	100	1	6	600	6
	300		7	700	7
	500		8	800	8
	700		9	900	9
	900		10	1000	10
	1000				

The different values for the hyperparameters in the cross validation in which the max depth restricts the amount of possible leaves; the Nr. of Boosts how many boosting rounds and trees the algorithm will create; the Learning rate decides how much each newly boosted tree their influence will be on the outcome; Nr. of Trees the amount of trees that will be created in the forest; the Min Sample Leaf the number of observations needed in each leaf.

3.7 Nowcasting Framework

In this paper the training data has two different starting dates namely January 1980 and January 2000. The nowcasts will be done only on the latest available data. On the first day of every quarter normally there have not been released any values for the quarter. Therefore, the first vintage of the data of every quarter will be used to nowcast the value of the previous quarter. This means that the vintage of the data of the first of January of 2017 will be used to nowcast Q4 of 2016 and that the vintages of the data from the 15th of January until the first of April of 2017 will be used to nowcast Q1 of 2017. This will result in only one nowcast for 2016 Q4 and 6 nowcasts for every following quarter.

4 Results

4.1 General Performance

Each nowcasting method has a model with different hyperparameters for each vintage of the data. Each of these models predict only the GDP growth for the latest quarterly data input, since there are 2 vintages of the data each month this results in 6 nowcasts for each quarter. Table 2 shows the mean squared error of the Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and the Hybrid Approach for the data set without the variables of the previous quarter (lag variables) with starting dates of the training data in January 1980 and January 2000. The variance of the squared errors can be found in the parenthesis. First thing to notice is the lower mean squared error for the more recent starting date of the training data for all the techniques. However, the smallest difference can be found with extreme gradient boosting and the hybrid approach. These two approaches also have a lower variance with the 1980 starting date of the training data than with the 2000 starting date of the training data were the Dynamic Factor Model and the Random Forest seem to have the lowest variance for their respective techniques.

Table 2: Mean Squared Errors of each method

Start training data	**Dynamic Factor Model**	**Random Forest**	**Extreme Gradient Boosting**	**Hybrid**
01-01-1980	1.335 (2.731)	0.827 (1.190)	0.732 (0.646)	0.682 (0.442)
01-01-2000	0.634 (0.672)	0.614 (0.601)	0.691 (0.997)	0.615 (0.638)

The mean squared error of a Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and Hybrid Approach with different starting date for the training data and the mean calculated over all the individual predictions from each vintage of the data. The variance of the squared error is shown in the parenthesis.

Table 3 shows the mean squared error of the Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and the Hybrid Approach for the data set with lag variables with starting dates of the training data in January 1980 and January 2000. The variance of the squared errors can be found in the parenthesis. When comparing Table 2 with Table 3 the most interesting thing to notice is the similar performance of the Random Forest, while the Extreme Gradient Boosting and Hybrid Approach perform better. Furthermore, the Hybrid Approach seems to have the lowest mean squared error and the lowest variance of the squared error which suggests it might be the best technique to use.

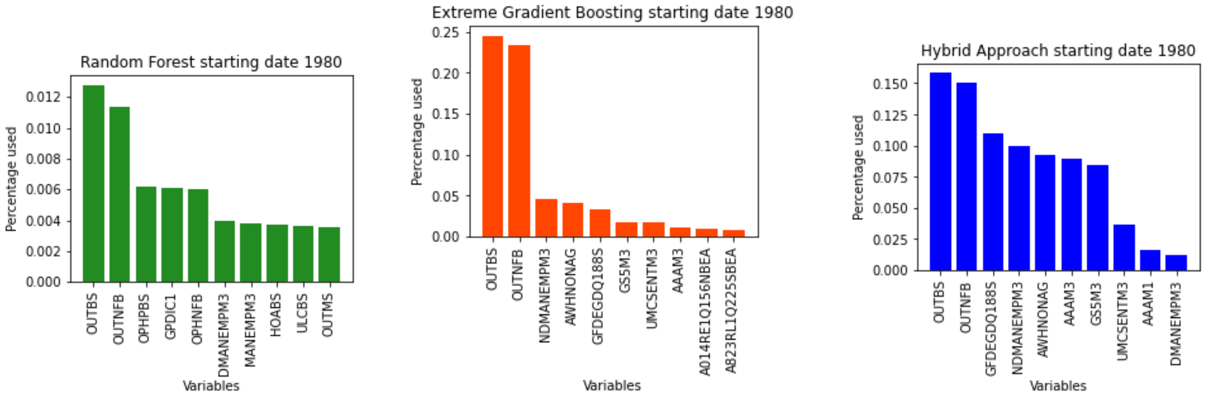
Table 3: Mean Squared Errors of each method including all lag variable in data

Start training data	**Dynamic Factor Model**	**Random Forest**	**Extreme Gradient Boosting**	**Hybrid**
01-01-1980	1.335 (2.731)	0.832 (1.050)	0.619 (0.540)	0.564 (0.412)
01-01-2000	0.634 (0.672)	0.598 (0.598)	0.552 (0.628)	0.537 (0.587)

The mean squared error of a Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and Hybrid Approach with different starting dates for the training data and the mean calculated over all the individual predictions from each vintage of the data with lag variables. The variance of the squared error is shown in the parenthesis.

4.2 Variables

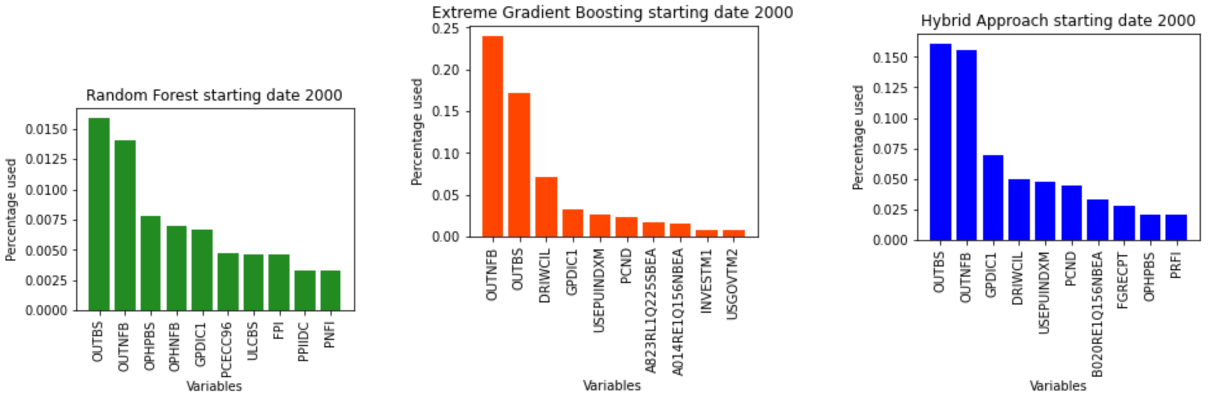
Figure 1 shows us the ten most used variables for each algorithm for the vintages of the data with starting date of the training data in 1980. The two most used variables are the same for all tree algorithms. However the percentage of time used is quite differently. The Random Forest only picks the most used variable 1.28% of the time while Extreme Gradient Boosting chose it 24.55% of the time. The Hybrid Approach selected it 15.83% of the time. This also results in more usage of other variables which can be seen further in the graph. This behavior is very easily explained by the selection of a random set of variables for each tree in a Random Forest. It is also noteworthy to see that the eight most used variables in Extreme Gradient Boosting can also be found in the eight most used variables of the Hybrid Approach. On the other hand it is seen that only one other variable besides the two most used variables of the Random Forest can be found in the top 10 most used variables of the Hybrid Approach.



(a) Random Forest **(b)** Extreme Gradient Boosting **(c)** Hybrid Approach

Figure 1: Percentage of the 10 most used variables in each method with start date of the training data in 1980 without lag variables.

Figure 2 shows us the ten most used variables for each algorithm for the vintages of the data with starting date of the training data in 2000. When comparing these results to the results of the earlier starting date of the training data shown in Figure 1, they seem very similar. Most noteworthy is the switch between the two most used variables for Extreme Gradient Boosting. The other two methods have the same most used variable. The most used variable for the Random Forest is only used 1.59% of the time. Although the Extreme Gradient Boosting is now using a different variable the most, it uses it with the highest usage percentage for all algorithms, 24.02%. The Hybrid Approach is in the middle with 16.12%.

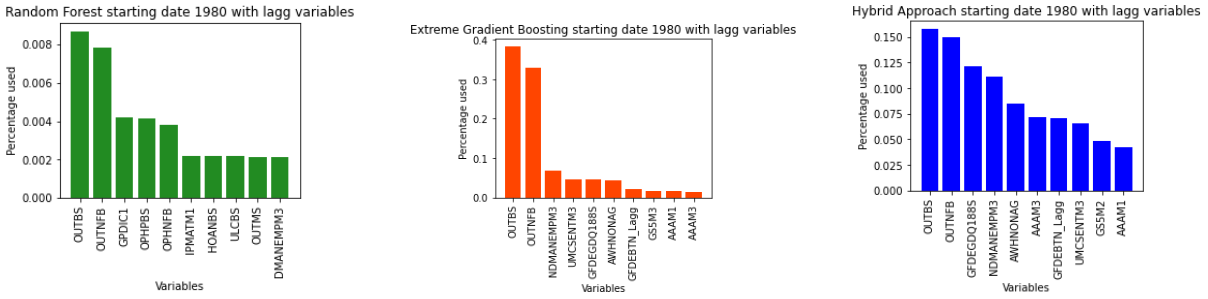


(a) Random Forest **(b)** Extreme Gradient Boosting **(c)** Hybrid Approach

Figure 2: Percentage of the 10 most used variables in each method with start date of the training data in 2000 without lag variables.

Figure 3 shows us the ten most used variables for each algorithm for the vintages of the data with starting date of the training data in 1980 with lag variables. Section

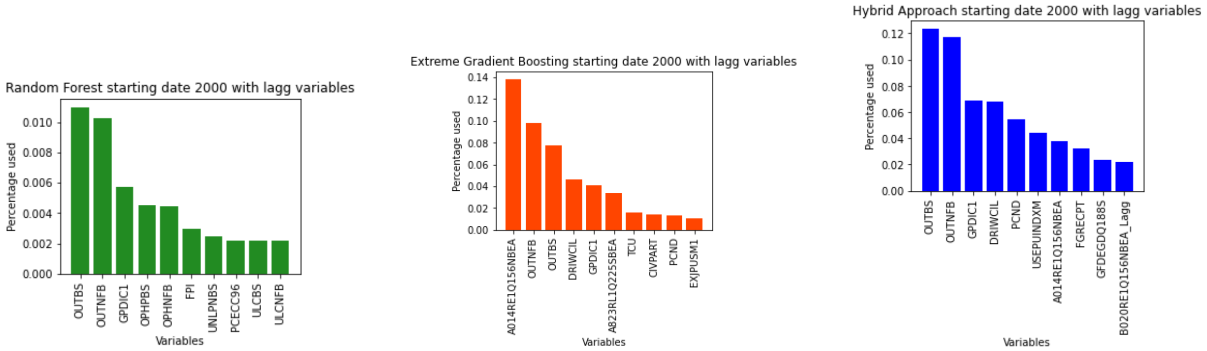
4.1 showed us that the Extreme Gradient Boosting and Hybrid Approach worked better with lag variables and the Random forest seemed to have the same performance. This can be seen by the inclusion of a lag variable in the 10 most used variables for Extreme Gradient Boosting and the Hybrid Approach and the lack thereof in the 10 most used variables for the Random Forest. Figure 3 also shows us that the Random Forest most used variable is used only 0.87% of the time which is quite a bit lower than the 1.28% without lag variables which could be expected since the Random Forest selects a lot of different variables because of the limitation of variables that are available to be chosen in each tree. The Extreme Gradient Boosting most used variable is used 38.26% of the time. Which is a lot more than the one without lag variable where it was only used 24.55% of the time. Which is an interesting finding since it could have been expected to be less due to having more variables available. The Hybrid Approach most used variable is chosen 15.76% of the time which is quite similar to the 15.83% without lag variables.



(a) Random Forest **(b)** Extreme Gradient Boosting **(c)** Hybrid Approach

Figure 3: Percentage of the 10 most used variables in each method with start date of the training data in 1980 with lag variables.

Figure 4 shows us the ten most used variables for each algorithm for the vintages of the data with starting date of the training data in 2000 with lag variables. Comparing these results with the earlier starting date shown in Figure 3 shows they are quite different except the Random Forest where the five most used variables are the same. The most used variables in Extreme Gradient Boosting are quite different from before. Now the most used variable was not even included in the top 10 most used variables for the vintages of the data with an earlier starting date of the training data. This variable is also only used 13.85% of the time while the most used variable with Extreme Gradient Boosting with the earlier starting date of the training data of the vintage of the data is used 38.26% of the time. The Hybrid Approach also seems to be different, with the exception of the two most used variables. However, the most used variable is only used 12.34% which is less than the 15.76% with the earlier starting date of the training data of the vintage of the data.



(a) Random Forest **(b)** Extreme Gradient Boosting **(c)** Hybrid Approach

Figure 4: Percentage of the 10 most used variables in each method with start date of the training data in 2000 with lag variables.

4.3 Performance per Quarter

Table 4 shows the GDP for each nowcast quarter with the MSE of the Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and the Hybrid Approach with the start of the training data in 1980. Each individual nowcast for each vintage of the data can be found in Table 14 in Appendix B. Table 4 shows that Q4 of 2018 is the quarter where the Dynamic Factor Model, Random Forest and Hybrid Approach have the worst performance. At the same time this is the quarter where the GDP is lowest. Since the Dynamic Factor Model overestimated the GDP, other values which have an positive effect on the GDP also have been overestimated if they were not known at the time of the nowcast. For example the estimation for the two most used variables in Random Forest and Hybrid Approach were twice as high as their true value. Extreme Gradient Boosting seems less prone to predict high or low values than the other techniques. This results in the biggest MSE for Extreme Gradient Boosting in Q4 2017, where the GDP has the biggest difference from the average.

Table 4: Mean Squared Errors of each method not including lag variables with the training data starting in 1980

Nowcast Quarter	**GDP**	**Dynamic Factor Model**	**Random Forest**	**Extreme Gradient Boosting**	**Hybrid**
2016 Q4	2.541	0.494 (N.A.)	0.027 (N.A.)	0.022 (N.A.)	0.006 (N.A.)
2017 Q1	2.282	0.618 (0.164)	0.348 (0.024)	0.342 (0.202)	0.703 (0.340)
2017 Q2	1.719	1.765 (0.387)	1.511 (0.297)	1.405 (0.155)	0.961 (0.388)
2017 Q3	2.947	0.248 (0.015)	0.026 (0.001)	0.011 (0.000)	0.034 (0.003)
2017 Q4	3.878	0.125 (0.031)	0.549 (0.040)	2.261 (1.000)	1.263 (0.027)
2018 Q1	3.779	0.092 (0.009)	0.297 (0.014)	1.440 (0.001)	1.036 (0.039)
2018 Q2	2.701	0.734 (0.264)	0.297 (0.024)	0.046 (0.001)	0.024 (0.001)
2018 Q3	2.117	2.363 (0.282)	1.303 (0.062)	0.257 (0.040)	0.431 (0.046)
2018 Q4	1.320	5.292 (0.871)	3.671 (0.460)	1.595 (0.330)	2.095 (0.142)
2019 Q1	2.932	0.118 (0.032)	0.031 (0.001)	0.614 (0.138)	0.395 (0.083)
2019 Q2	1.491	3.697 (0.487)	1.838 (0.399)	0.584 (0.038)	0.887 (0.196)
2019 Q3	2.572	0.527 (0.315)	0.120 (0.053)	0.228 (0.010)	0.393 (0.171)
2019 Q4	2.366	0.582 (0.020)	0.062 (0.002)	0.115 (0.000)	0.075 (0.001)

The GDP with the mean squared error of a Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and Hybrid Approach for each quarter. The variance of the squared error is shown in the parenthesis.

Table 5 shows the GDP for each nowcast quarter with the MSE of the Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and the Hybrid Approach with the start of the training data in 2000. Each individual nowcast for each vintage of the data can be found in Table 15 in Appendix B. Table 5 shows that Q4 of 2018 is the quarter where the Dynamic Factor Model, Random Forest and Hybrid Approach have the highest MSE. This is the same quarter where these techniques have the highest MSE as with the longer training data. The reason is the same as with the longer training data. Variables that are not known yet are predicted, by the Dynamic Factor Model, with a higher value than the true value of the variable. Both the Dynamic Factor Model and Random Forest have a lower MSE with the training data starting in 2000. The lower MSE can be explained by the lower predictions of the GDP. The GDP is generally predicted lower because the training data set from 2000 has a lower average GDP than the training data set from 1980.

Table 5: Mean Squared Errors of each method including no lag variable in 2000 data

Nowcast Quarter	**GDP**	**Dynamic Factor Model**	**Random Forest**	**Extreme Gradient Boosting**	**Hybrid**
2016 Q4	2.541	0.003 (N.A.)	0.088 (N.A.)	0.287 (N.A.)	0.006 (N.A.)
2017 Q1	2.282	0.105 (0.007)	0.037 (0.002)	0.027 (0.001)	0.013 (0.000)
2017 Q2	1.719	0.567 (0.034)	0.687 (0.118)	0.182 (0.075)	0.560 (0.211)
2017 Q3	2.947	0.048 (0.006)	0.098 (0.006)	0.391 (0.199)	0.099 (0.012)
2017 Q4	3.878	1.071 (0.152)	1.207 (0.039)	3.156 (0.552)	1.101 (0.168)
2018 Q1	3.779	0.743 (0.112)	1.302 (0.059)	1.333 (0.348)	0.869 (0.080)
2018 Q2	2.701	0.041 (0.003)	0.020 (0.001)	0.018 (0.000)	0.133 (0.002)
2018 Q3	2.117	0.639 (0.023)	0.374 (0.048)	0.365 (0.219)	0.415 (0.063)
2018 Q4	1.320	2.605 (0.174)	2.477 (0.296)	1.556 (1.298)	2.526 (0.782)
2019 Q1	2.932	0.051 (0.002)	0.118 (0.017)	0.394 (0.097)	0.366 (0.099)
2019 Q2	1.491	1.741 (0.277)	1.060 (0.299)	0.727 (0.277)	1.231 (0.642)
2019 Q3	2.572	0.077 (0.025)	0.055 (0.000)	0.164 (0.005)	0.152 (0.021)
2019 Q4	2.366	0.025 (0.001)	0.015 (0.000)	0.044 (0.000)	0.019 (0.000)

The mean squared error of a Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and Hybrid Approach for each quarter. The variance of the squared error is shown in the parenthesis.

Table 6 shows the GDP for each nowcast quarter with the MSE of the Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and the Hybrid Approach with the start of the training data in 1980 that include lag variables. Each individual nowcast for each vintage of the data can be found in Table 16 in Appendix B. Table 6 shows that Q4 of 2018 is the quarter where the Dynamic Factor Model, Random Forest and Hybrid Approach have the worst performance. This is similar to the results shown in Table 4 and 5. The reason of the bad performance is also the same as in Table 4 and 5. However, there is an improvement in the Hybrid Approach with a lower MSE and lower variance of the MSE in this quarter. This improvement can also be seen in the general performance as described in Section 4.1. Extreme Gradient Boosting has the biggest improvement in Q2 of 2017. This improvement shows that adding the lag variables can help quite a lot in some instances.

Table 6: Mean Squared Errors of each method including all lag variable in 1980 data

Nowcast Quarter	**GDP**	**Dynamic Factor Model**	**Random Forest**	**Extreme Gradient Boosting**	**Hybrid**
2016 Q4	2.541	0.494 (N.A.)	0.022 (N.A.)	0.221 (N.A.)	0.093 (N.A.)
2017 Q1	2.282	0.618 (0.164)	0.262 (0.034)	0.040 (0.000)	0.044 (0.003)
2017 Q2	1.719	1.765 (0.387)	1.468 (0.032)	0.158 (0.010)	0.240 (0.025)
2017 Q3	2.947	0.248 (0.015)	0.027 (0.003)	0.467 (0.083)	0.166 (0.026)
2017 Q4	3.878	0.125 (0.031)	1.505 (0.173)	2.205 (0.821)	1.483 (0.433)
2018 Q1	3.779	0.092 (0.009)	0.434 (0.010)	1.440 (0.001)	1.036 (0.039)
2018 Q2	2.701	0.734 (0.264)	0.360 (0.011)	0.046 (0.001)	0.024 (0.001)
2018 Q3	2.117	2.363 (0.282)	1.002 (0.018)	0.210 (0.026)	0.347 (0.048)
2018 Q4	1.320	5.292 (0.871)	3.570 (0.462)	1.433 (0.296)	1.887 (0.071)
2019 Q1	2.932	0.118 (0.032)	0.053 (0.003)	0.511 (0.089)	0.423 (0.078)
2019 Q2	1.491	3.697 (0.487)	1.332 (0.087)	0.604 (0.034)	0.919 (0.091)
2019 Q3	2.572	0.527 (0.315)	0.067 (0.008)	0.223 (0.022)	0.211 (0.020)
2019 Q4	2.366	0.582 (0.020)	0.049 (0.004)	0.160 (0.000)	0.071 (0.001)

The mean squared error of a Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and Hybrid Approach for each quarter. The variance of the squared error is shown in the parenthesis.

Table 7 shows the GDP for each nowcast quarter with the MSE of the Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and the Hybrid Approach with the start of the training data in 2000 that include lag variables. Each individual nowcast for each vintage of the data can be found in Table 17 in Appendix B. Table 7 shows that every technique has the highest MSE in Q4 of 2018. Extreme Gradient Boosting seems to have made an improvement in Q4 of 2017 which results in the highest MSE for Q4 2018 which is more in line with the other techniques.

Table 7: Mean Squared Errors of each method including all lag variable in 2000 data

Nowcast Quarter	**GDP**	**Dynamic Factor Model**	**Random Forest**	**Extreme Gradient Boosting**	**Hybrid**
2016 Q4	2.541	0.003 (N.A.)	0.036 (N.A.)	0.372 (N.A.)	0.083 (N.A.)
2017 Q1	2.282	0.105 (0.007)	0.054 (0.002)	0.068 (0.006)	0.063 (0.003)
2017 Q2	1.719	0.567 (0.034)	0.442 (0.031)	0.400 (0.160)	0.387 (0.077)
2017 Q3	2.947	0.048 (0.006)	0.240 (0.008)	0.084 (0.004)	0.081 (0.021)
2017 Q4	3.878	1.071 (0.152)	1.912 (0.059)	1.236 (0.389)	1.076 (0.063)
2018 Q1	3.779	0.743 (0.112)	1.003 (0.004)	0.823 (0.103)	0.701 (0.035)
2018 Q2	2.701	0.041 (0.003)	0.013 (0.000)	0.071 (0.008)	0.035 (0.001)
2018 Q3	2.117	0.639 (0.023)	0.265 (0.031)	0.645 (0.485)	0.284 (0.074)
2018 Q4	1.320	2.605 (0.174)	2.329 (0.377)	2.209 (2.214)	2.284 (0.462)
2019 Q1	2.932	0.051 (0.002)	0.180 (0.021)	0.311 (0.085)	0.185 (0.026)
2019 Q2	1.491	1.741 (0.277)	0.724 (0.132)	0.522 (0.297)	1.290 (1.285)
2019 Q3	2.572	0.077 (0.025)	0.029 (0.001)	0.137 (0.010)	0.101 (0.039)
2019 Q4	2.366	0.025 (0.001)	0.083 (0.001)	0.146 (0.003)	0.030 (0.001)

The mean squared error of a Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and Hybrid Approach for each quarter. The variance of the squared error is shown in the parenthesis.

5 Conclusion & Discussion

This paper looks at a new approach to nowcast the US GDP. This new approach is a combination of Extreme Gradient Boosting and Random Forest. This Hybrid approach is compared to the two methods of which it consists and the Dynamic Factor Model as benchmark.

The Dynamic Factor Model is used to deal with the ragged edge issue where it uses it smoothed series as input variables for the delayed data. Each model is than trained by each vintage of the data and gives one nowcasts for the latest available quarter. The algorithms are compared by the MSE and the variance of the squared error.

Looking at the results it shows that the Dynamic Factor Model can be tough to beat but there seems to be a big difference with the different starting dates for vintage of the data. When the vintage of the data has a more recent starting date the Dynamic Factor Model seems to perform better. This is because the in-sample average GDP of the shorter data set is more similar with the out of sample GDP than the in-sample GDP of the longer data set. The results also show that the Random Forest does not perform better when adding the variables of the previous quarter (lag variables). Extreme Gradient Boosting and the Hybrid Approach seem to perform better when adding lag variables. Extreme Gradient Boosting and the Hybrid Approach also have a smaller variance when having a longer vintage of the data to train with. The opposite is true for the Dynamic Factor Model and Random Forest.

While looking at the percentage use of variables it shows that the Random Forest spreads the usage of variables more evenly than Extreme Gradient Boosting. This also explains why Random Forest does not perform better when adding lag variables. Extreme Gradient Boosting is less likely to add more variables but can be very heavily invested in some variables. Here the Hybrid Approach seems to make the most advantage. Since it selects the variables more evenly, similar to the random forest, it takes in more information from the variables. However, the variable selection of the extreme gradient boosting reduces the noise in the input data which results in a lower MSE.

The promising results of the Hybrid Approach should make it an interesting method to explore in future research in this field or any other field when having a lot of different variables.

References

Bok, B., Caratelli, D., Giannone, D., Sbordone, A., and Tambalotti, A. (2017). Macroeconomic nowcasting and forecasting with big data (830).

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

Genuer, R., Poggi, J.-M., and Tuleau-Malot, C. (2015). VSURF: an R package for variable selection using random forests. *The R Journal*, 7(2):19–33.

Jansen, W. J., Jin, X., and de Winter, J. M. (2016). Forecasting and nowcasting real GDP: Comparing statistical models and subjective forecasts. *International Journal of Forecasting*, 32(2):411–436.

Jiang, H., Deng, Y., Chen, H.-S., Tao, L., Sha, Q., Chen, J., Tsai, C.-J., and Zhang, S. (2004). Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC bioinformatics*, 5(1):1–12.

McCracken, M. and Ng, S. (2020). FRED-QD: A quarterly database for macroeconomic research. Technical report, National Bureau of Economic Research.

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

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Richardson, A., van Florenstein Mulder, T., and Vehbi, T. (2021). Nowcasting GDP using machine-learning algorithms: A real-time assessment. *International Journal of Forecasting*, 37(2):941–948.

Soybilgen, B. and Yazgan, E. (2021). Nowcasting US GDP using tree-based ensemble models and dynamic factors. *Computational Economics*, 57(1):387–417.

Speiser, J. L., Miller, M. E., Tooze, J., and Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. *Expert systems with applications*, 134:93–101.

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

Appendices

A Data

All these series id's can be directly downloaded by the use of the FRED API [1]. The transformations are the same as in McCracken and Ng (2016) and McCracken and Ng (2020). Namely, (1) No transformation; (2)Δx_t; (3)$\Delta^2 x_t$; (4) $log(x_t)$; (5) $\Delta log(x_t)$; (6) $\Delta^2 log(x_t)$; (7) $\Delta(x_t/x_{t-1} - 1.0)$. Tables 8, 9, 10, 11, 12, 13 show how all the variables are transformed and which series were included in the FRED-MD and which in the FRED-QD.

Table 8: Transformation 1

FRED-MD	FRED-QD
CES0600000007	A014RE1Q156NBEA
AWHMAN	A823RL1Q225SBEA
TB3SMFFM	TCU
TB6SMFFM	BAA10YM
T1YFFM	DRIWCIL
T5YFFM	
T10YFFM	
AAAFFM	
BAAFFM	
VXOCLS	

The variables that follow no transformation in the FRED-MD and FRED-QD

[1] $https://fred.stlouisfed.org/docs/api/fred/$

Table 9: Transformation 2

FRED-MD	FRED-QD:
CUMFNS	CIVPART
UNRATE	LNS14000012
UEMPMEAN	LNS14000025
AWOTMAN	LNS14000026
ISRATIO	AWHNONAG
UMCSENT	USEPUINDXM
FEDFUNDS	B020RE1Q156NBEA
TB3MS	B021RE1Q156NBEA
TB6MS	GFDEGDQ188S
GS1	GFDEBTN
GS5	
GS10	
AAA	
BAA	

The variables that follow Δx_t in the FRED-MD and FRED-QD

Table 10: Transformation 4

FRED-MD	FRED-QD:
HOUST	
HOUSTNE	
HOUSTMW	
HOUSTS	
HOUSTW	
PERMIT	
PERMITNE	
PERMITMW	
PERMITS	
PERMITW	

The variables that follow $log(x_t)$ in the FRED-MD and FRED-QD

Table 11: Transformation 5

FRED-MD	FRED-QD
RPI	PCECC96
W875RX1	PCDG
DPCERA3M086SBEA	PCESV
CMRMTSPL	PCND
INDPRO	GPDIC1
IPFPNSS	FPI
IPFINAL	Y033RC1Q027SBEA
IPCONGD	PNFI
IPDCONGD	PRFI
IPNCONGD	GCEC1
IPBUSEQ	FGRECPT
IPMAT	SLCE
IPDMAT	EXPGSC1
IPNMAT	IMPGSC1
IPMANSICS	DPIC96
IPB51222S	OUTNFB
IPFUELS	OUTBS
CLF16OV	OUTMS
CE16OV	IPB51110SQ
UEMPLT5	IPB51220SQ
UEMP5TO14	USPRIV
UEMP15OV	USEHS
UEMP15T26	USINFO
UEMP27OV	USPBS
PAYEMS	USLAH
USGOOD	USSERV
CES1021000001	USMINE
USCONS	CES9091000001
MANEMP	CES9092000001
DMANEMP	CES9093000001
NDMANEMP	LNS13023621
SRVPRD	LNS13023557
USTPU	LNS13023705
USWTRADE	LNS13023569
USTRADE	LNS12032194
USFIRE	HOABS
USGOVT	HOAMS
ACOGNO	HOANBS
DGORDER	HOUST5F
ANDENO	RSAFS
AMDMUO	INVCQRMTSPL
BUSINV	WPU0531
M2REAL	WPU0561
EXCAUS	AHETPI
EXUSUK	COMPRMS
EXJPUS	COMPRNFB
EXSZUS	RCPHBS
	OPHMFG
	OPHNFB
	OPHPBS
	ULCBS
	ULCMFG
	ULCNFB
	UNLPNBS
	IMFSL
	M1REAL
	MZMREAL
	CONSUMER
	REVOLSL
	TOTALSL
	TABSHNO
	TLBSHNO
	TNWBSHNO
	HNOREMQ027S
	TFAABSHNO
	USSTHPI
	SPCS10RSA
	SPCS20RSA
	EXUSEU
	TLBSNNCB
	TTAABSNNCB
	TNWMVBSNNCB
	TLBSNNB
	TABSNNB
	TNWBSNNB
	CNCF

The variables that follow $\Delta log(x_t)$ in the FRED-MD and FRED-QD

Table 12: Transformation 6

FRED-MD	FRED-QD
M1SL	PCECTPI
M2SL	PCEPILFE
BOGMBASE	GDPCTPI
TOTRESNS	GPDICTPI
BUSLOANS	IPDBS
REALLN	DGDSRG3Q086SBEA
NONREVSL	DDURRG3Q086SBEA
WPSFD49207	DSERRG3Q086SBEA
WPSFD49502	DNDGRG3Q086SBEA
WPSID61	DHCERG3Q086SBEA
WPSID62	DMOTRG3Q086SBEA
PPICMM	DFDHRG3Q086SBEA
CPIAUCSL	DREQRG3Q086SBEA
CPIAPPSL	DODGRG3Q086SBEA
CPITRNSL	DFXARG3Q086SBEA
CPIMEDSL	DCLORG3Q086SBEA
CUSR0000SAC	DGOERG3Q086SBEA
CUSR0000SAD	DONGRG3Q086SBEA
CUSR0000SAS	DHUTRG3Q086SBEA
CPIULFSL	DHLCRG3Q086SBEA
CUSR0000SA0L2	DTRSRG3Q086SBEA
CUSR0000SA0L5	DRCARG3Q086SBEA
PCEPI	DFSARG3Q086SBEA
DDURRG3M086SBEA	DIFSRG3Q086SBEA
DNDGRG3M086SBEA	DOTSRG3Q086SBEA
DSERRG3M086SBEA	CPILFESL
CES0600000008	PPIACO
CES2000000008	WPSFD4111
CES3000000008	PPIIDC
MZMSL	CUSR0000SEHC
DTCOLNVHFNM	
DTCTHFNM	
INVEST	

The variables that follow $\Delta^2 log(x_t)$ in the FRED-MD and FRED-QD

Table 13: transformation 7

FRED-MD	FRED-QD
NONBORRES	

The variables that follow $\Delta(x_t/x_{t-1} - 1.0)$ in the FRED-MD and FRED-QD

B Performance

Table 14: The results of the vintages of the data with starting date 1980 without lag variables

Date	GDP	Dynamic Factor Model		Random Forest		Extreme Gradient Boosting		Hybrid	
		Prediction	$Error^2$	Prediction	$Error^2$	Prediction	$Error^2$	Prediction	$Error^2$
01/01/2017	2.541	3.244	0.494	2.704	0.027	2.690	0.022	2.465	0.006
15/01/2017	2.282	3.126	0.712	2.962	0.462	2.690	0.166	2.419	0.019
01/02/2017	2.282	2.393	0.012	2.631	0.122	2.678	0.156	2.062	0.048
15/02/2017	2.282	2.925	0.414	2.845	0.317	2.678	0.156	3.137	0.732
01/03/2017	2.282	2.980	0.487	2.822	0.291	2.677	0.156	3.174	0.796
15/03/2017	2.282	3.241	0.921	2.846	0.318	2.677	0.156	3.354	1.150
01/04/2017	2.282	3.359	1.160	3.042	0.578	3.405	1.260	3.496	1.475
15/04/2017	1.719	3.260	2.374	2.934	1.478	2.680	0.924	3.187	2.155
01/05/2017	1.719	2.667	0.898	2.733	1.029	2.653	0.873	2.385	0.443
15/05/2017	1.719	3.100	1.909	2.923	1.449	3.010	1.668	2.649	0.865
01/06/2017	1.719	3.309	2.528	3.319	2.562	3.013	1.674	2.747	1.057
15/06/2017	1.719	2.905	1.407	2.903	1.402	3.013	1.674	2.502	0.614
01/07/2017	1.719	2.933	1.475	2.789	1.146	2.989	1.615	2.514	0.633
15/07/2017	2.947	3.462	0.264	3.139	0.037	2.989	0.002	2.589	0.128
01/08/2017	2.947	3.025	0.006	2.980	0.001	3.043	0.009	2.702	0.060
15/08/2017	2.947	3.518	0.326	3.230	0.080	3.061	0.013	3.057	0.012
01/09/2017	2.947	3.519	0.327	3.147	0.040	3.072	0.016	2.995	0.002
15/09/2017	2.947	3.471	0.274	2.964	0.000	2.802	0.021	2.914	0.001
01/10/2017	2.947	3.487	0.291	2.964	0.000	3.002	0.003	2.999	0.003
15/10/2017	3.878	3.228	0.423	3.032	0.715	1.950	3.718	2.660	1.483
01/11/2017	3.878	3.375	0.253	2.956	0.849	2.042	3.371	2.677	1.441
15/11/2017	3.878	3.619	0.067	3.118	0.577	2.603	1.625	2.761	1.248
01/12/2017	3.878	3.800	0.006	3.255	0.388	2.607	1.616	2.790	1.183
15/12/2017	3.878	3.896	0.000	3.238	0.410	2.607	1.616	2.846	1.065
01/01/2018	3.878	3.940	0.004	3.282	0.355	2.605	1.621	2.801	1.160
15/01/2018	3.779	3.803	0.001	3.273	0.256	2.605	1.379	2.850	0.863
01/02/2018	3.779	3.292	0.237	3.159	0.384	2.581	1.436	2.643	1.290
15/02/2018	3.779	3.475	0.092	3.177	0.363	2.581	1.436	2.736	1.088
01/03/2018	3.779	3.367	0.170	3.114	0.443	2.580	1.437	2.689	1.188
15/03/2018	3.779	3.843	0.004	3.369	0.168	2.562	1.480	2.766	1.026
01/04/2018	3.779	3.993	0.046	3.372	0.165	2.566	1.471	2.908	0.760
15/04/2018	2.701	4.015	1.726	3.448	0.558	2.566	0.018	2.921	0.048
01/05/2018	2.701	3.265	0.318	3.050	0.122	2.421	0.078	2.671	0.001
15/05/2018	2.701	3.517	0.665	3.132	0.185	2.541	0.026	2.623	0.006
01/06/2018	2.701	3.583	0.777	3.303	0.362	2.539	0.026	2.600	0.010
15/06/2018	2.701	3.408	0.499	3.261	0.314	2.539	0.026	2.571	0.017
01/07/2018	2.701	3.347	0.417	3.192	0.240	2.386	0.100	2.456	0.060
15/07/2018	2.117	3.404	1.656	3.137	1.039	2.386	0.072	2.415	0.088
01/08/2018	2.117	3.537	2.014	3.214	1.203	2.426	0.095	2.971	0.729
15/08/2018	2.117	3.639	2.315	3.217	1.208	2.779	0.438	2.701	0.341
01/09/2018	2.117	3.615	2.243	3.210	1.193	2.357	0.057	2.744	0.393
15/09/2018	2.117	3.812	2.873	3.438	1.744	2.781	0.440	2.820	0.493
01/10/2018	2.117	3.872	3.079	3.312	1.428	2.780	0.440	2.852	0.540
15/10/2018	1.320	3.919	6.753	3.430	4.454	2.780	2.133	2.873	2.412
01/11/2018	1.320	3.371	4.206	3.171	3.426	2.354	1.071	2.564	1.549
15/11/2018	1.320	3.686	5.600	3.397	4.314	2.773	2.112	2.865	2.389
01/12/2018	1.320	3.726	5.788	3.315	3.983	2.773	2.112	2.886	2.453
15/12/2018	1.320	3.501	4.757	3.028	2.919	2.354	1.071	2.716	1.949
01/01/2019	1.320	3.476	4.650	3.031	2.929	2.354	1.069	2.667	1.816
15/01/2019	2.932	3.588	0.430	3.214	0.079	2.772	0.026	2.734	0.039
01/02/2019	2.932	3.420	0.238	3.121	0.036	2.346	0.344	2.573	0.129
15/02/2019	2.932	2.993	0.004	2.777	0.024	2.011	0.848	2.029	0.815
01/03/2019	2.932	3.098	0.027	2.793	0.019	2.166	0.586	2.343	0.348
15/03/2019	2.932	2.949	0.000	2.968	0.001	1.964	0.938	2.169	0.583
01/04/2019	2.932	3.019	0.007	3.100	0.028	1.961	0.944	2.257	0.456
15/04/2019	1.491	3.042	2.406	2.970	2.188	1.961	0.220	2.149	0.433
01/05/2019	1.491	3.596	4.432	3.175	2.837	2.333	0.709	2.628	1.292
15/05/2019	1.491	3.441	3.804	2.832	1.798	2.324	0.694	2.304	0.661
01/06/2019	1.491	3.514	4.090	2.848	1.842	2.323	0.693	2.307	0.666
15/06/2019	1.491	3.377	3.556	2.610	1.252	2.323	0.693	2.326	0.697
01/07/2019	1.491	3.465	3.897	2.545	1.110	2.193	0.493	2.746	1.574
15/07/2019	2.572	3.350	0.605	2.936	0.132	2.183	0.152	2.993	0.177
01/08/2019	2.572	3.839	1.606	3.332	0.578	2.819	0.061	3.682	1.233
15/08/2019	2.572	2.935	0.132	2.638	0.004	2.035	0.288	2.061	0.261
01/09/2019	2.572	2.894	0.103	2.547	0.001	2.035	0.288	2.043	0.280
15/09/2019	2.572	3.095	0.273	2.511	0.004	2.035	0.288	2.101	0.221
01/10/2019	2.572	3.238	0.443	2.579	0.000	2.035	0.289	2.140	0.187
15/10/2019	2.366	3.103	0.544	2.686	0.102	2.035	0.109	2.050	0.100
01/11/2019	2.366	3.229	0.745	2.740	0.140	2.024	0.117	2.148	0.047
15/11/2019	2.366	3.005	0.409	2.542	0.031	2.024	0.117	2.015	0.123
01/12/2019	2.366	3.063	0.486	2.573	0.043	2.026	0.115	2.111	0.065
15/12/2019	2.366	3.109	0.553	2.493	0.016	2.026	0.115	2.112	0.064
01/01/2020	2.366	3.236	0.758	2.559	0.037	2.026	0.116	2.140	0.051

The GDP and the prediction, squared error of a Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and Hybrid Approach for the vintages of the data with starting date 1980 without lag variables

Table 15: The results of the vintages of the data with starting date 2000 without lag variables

Date	**GDP**	**Dynamic Factor Model**		**Random Forest**		**Extreme Gradient Boosting**		**Hybrid**	
		Prediction	$Error^2$	Prediction	$Error^2$	Prediction	$Error^2$	Prediction	$Error^2$
01/01/2017	2.541	2.599	0.003	2.245	0.088	2.005	0.287	2.464	0.006
15/01/2017	2.282	2.476	0.038	2.456	0.030	2.005	0.077	2.310	0.001
01/02/2017	2.282	1.850	0.186	2.343	0.004	2.127	0.024	2.263	0.000
15/02/2017	2.282	2.426	0.021	2.392	0.012	2.134	0.022	2.454	0.030
01/03/2017	2.282	2.439	0.025	2.394	0.013	2.136	0.021	2.300	0.000
15/03/2017	2.282	2.677	0.156	2.613	0.109	2.377	0.009	2.306	0.001
01/04/2017	2.282	2.735	0.205	2.517	0.055	2.381	0.010	2.494	0.045
15/04/2017	1.719	2.574	0.731	2.592	0.763	2.267	0.300	2.540	0.674
01/05/2017	1.719	2.335	0.380	2.379	0.435	1.558	0.026	2.076	0.128
15/05/2017	1.719	2.543	0.679	2.631	0.833	1.821	0.011	2.648	0.864
01/06/2017	1.719	2.604	0.783	2.847	1.274	2.552	0.694	2.848	1.275
15/06/2017	1.719	2.348	0.395	2.335	0.380	1.549	0.029	2.229	0.261
01/07/2017	1.719	2.376	0.432	2.381	0.439	1.545	0.030	2.118	0.160
15/07/2017	2.947	2.688	0.067	2.762	0.034	1.858	1.187	2.643	0.093
01/08/2017	2.947	2.506	0.194	2.514	0.188	2.132	0.666	2.387	0.314
15/08/2017	2.947	2.822	0.016	2.682	0.070	2.574	0.139	2.688	0.067
01/09/2017	2.947	2.898	0.002	2.764	0.034	2.607	0.116	2.776	0.029
15/09/2017	2.947	2.879	0.005	2.503	0.197	2.580	0.135	2.724	0.050
01/10/2017	2.947	2.880	0.004	2.696	0.063	2.623	0.105	2.751	0.038
15/10/2017	3.878	2.580	1.685	2.663	1.475	2.469	1.986	2.587	1.666
01/11/2017	3.878	2.698	1.391	2.725	1.329	1.989	3.567	2.933	0.892
15/11/2017	3.878	2.874	1.007	2.727	1.324	1.935	3.775	2.703	1.380
01/12/2017	3.878	2.926	0.907	2.849	1.059	1.891	3.949	2.734	1.308
15/12/2017	3.878	3.020	0.736	2.836	1.086	2.197	2.827	3.060	0.670
01/01/2018	3.878	3.041	0.701	2.894	0.968	2.195	2.833	3.047	0.690
15/01/2018	3.779	2.939	0.706	2.746	1.067	2.241	2.367	2.964	0.664
01/02/2018	3.779	2.677	1.214	2.683	1.201	2.610	1.367	2.694	1.178
15/02/2018	3.779	2.861	0.843	2.776	1.006	2.610	1.367	2.710	1.143
01/03/2018	3.779	2.803	0.953	2.547	1.518	2.609	1.369	2.789	0.980
15/03/2018	3.779	3.133	0.418	2.577	1.445	2.964	0.664	2.889	0.792
01/04/2018	3.779	3.212	0.322	2.524	1.574	2.850	0.864	3.102	0.459
15/04/2018	2.701	3.074	0.138	2.950	0.062	2.850	0.022	3.057	0.126
01/05/2018	2.701	2.731	0.001	2.782	0.007	2.636	0.004	3.115	0.171
15/05/2018	2.701	2.895	0.037	2.851	0.022	2.848	0.022	3.113	0.169
01/06/2018	2.701	2.915	0.046	2.742	0.002	2.854	0.023	3.107	0.165
15/06/2018	2.701	2.824	0.015	2.749	0.002	2.549	0.023	3.013	0.097
01/07/2018	2.701	2.786	0.007	2.863	0.026	2.585	0.013	2.963	0.068
15/07/2018	2.117	2.728	0.373	2.779	0.437	2.585	0.219	2.855	0.544
01/08/2018	2.117	2.948	0.690	3.003	0.784	3.244	1.268	3.049	0.867
15/08/2018	2.117	2.911	0.630	2.704	0.344	2.188	0.005	2.582	0.216
01/09/2018	2.117	2.887	0.592	2.540	0.179	2.201	0.007	2.598	0.231
15/09/2018	2.117	2.984	0.751	2.621	0.253	2.702	0.342	2.664	0.299
01/10/2018	2.117	3.011	0.799	2.614	0.246	2.708	0.349	2.696	0.334
15/10/2018	1.320	3.138	3.306	3.119	3.237	3.122	3.249	3.065	3.044
01/11/2018	1.320	2.799	2.187	2.800	2.190	2.083	0.583	2.572	1.569
15/11/2018	1.320	2.963	2.701	3.007	2.847	2.794	2.173	3.167	3.411
01/12/2018	1.320	2.989	2.788	2.975	2.739	2.794	2.173	3.189	3.495
15/12/2018	1.320	2.854	2.355	2.715	1.947	2.083	0.583	2.667	1.816
01/01/2019	1.320	2.835	2.296	2.699	1.901	2.077	0.573	2.669	1.821
15/01/2019	2.932	2.865	0.005	2.914	0.000	2.786	0.021	2.880	0.003
01/02/2019	2.932	2.846	0.007	2.867	0.004	2.786	0.021	2.921	0.000
15/02/2019	2.932	2.646	0.082	2.558	0.140	2.104	0.686	2.145	0.620
01/03/2019	2.932	2.768	0.027	2.635	0.088	2.338	0.353	2.338	0.353
15/03/2019	2.932	2.596	0.113	2.578	0.125	2.133	0.639	2.060	0.761
01/04/2019	2.932	2.668	0.070	2.339	0.352	2.130	0.643	2.254	0.460
15/04/2019	1.491	2.588	1.203	2.442	0.905	2.130	0.408	2.331	0.705
01/05/2019	1.491	3.149	2.747	2.921	2.045	2.831	1.796	3.178	2.846
15/05/2019	1.491	2.762	1.616	2.525	1.069	2.208	0.513	2.438	0.897
01/06/2019	1.491	2.806	1.730	2.300	0.654	2.225	0.539	2.543	1.107
15/06/2019	1.491	2.714	1.496	2.583	1.191	2.235	0.554	2.446	0.911
01/07/2019	1.491	2.778	1.656	2.194	0.495	2.235	0.553	2.449	0.918
15/07/2019	2.572	2.771	0.040	2.774	0.041	2.723	0.023	2.915	0.118
01/08/2019	2.572	3.204	0.400	2.831	0.067	3.032	0.212	3.221	0.422
15/08/2019	2.572	2.517	0.003	2.366	0.042	2.127	0.198	2.201	0.138
01/09/2019	2.572	2.484	0.008	2.363	0.044	2.132	0.193	2.162	0.168
15/09/2019	2.572	2.572	0.000	2.280	0.085	2.132	0.193	2.343	0.052
01/10/2019	2.572	2.667	0.009	2.345	0.052	2.169	0.162	2.442	0.017
15/10/2019	2.366	2.410	0.002	2.378	0.000	2.115	0.063	2.170	0.038
01/11/2019	2.366	2.516	0.023	2.316	0.002	2.162	0.042	2.236	0.017
15/11/2019	2.366	2.430	0.004	2.164	0.041	2.162	0.042	2.228	0.019
01/12/2019	2.366	2.451	0.007	2.152	0.046	2.169	0.039	2.224	0.020
15/12/2019	2.366	2.567	0.040	2.346	0.000	2.169	0.039	2.472	0.011
01/01/2020	2.366	2.633	0.071	2.399	0.001	2.168	0.039	2.455	0.008

The GDP and the prediction, squared error of a Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and Hybrid Approach for the vintages of the data with starting date 2000 without lag variables

Table 16: The results of the vintages of the data with starting date 1980 with lag variables

Date	**GDP**	**Dynamic Factor Model**		**Random Forest**		**Extreme Gradient Boosting**		**Hybrid**	
		Prediction	$Error^2$	Prediction	$Error^2$	Prediction	$Error^2$	Prediction	$Error^2$
01/01/2017	2.541	3.244	0.494	2.688	0.022	2.071	0.221	2.237	0.093
15/01/2017	2.282	3.126	0.712	2.786	0.254	2.071	0.044	2.307	0.001
01/02/2017	2.282	2.393	0.012	2.750	0.219	2.060	0.049	2.027	0.065
15/02/2017	2.282	2.925	0.414	2.580	0.089	2.060	0.049	2.276	0.000
01/03/2017	2.282	2.980	0.487	2.608	0.106	2.060	0.049	2.355	0.005
15/03/2017	2.282	3.241	0.921	2.836	0.307	2.060	0.049	2.524	0.059
01/04/2017	2.282	3.359	1.160	3.055	0.598	2.296	0.000	2.647	0.133
15/04/2017	1.719	3.260	2.374	2.957	1.534	2.062	0.118	2.340	0.386
01/05/2017	1.719	2.667	0.898	2.789	1.145	1.924	0.042	1.880	0.026
15/05/2017	1.719	3.100	1.909	2.912	1.424	2.133	0.171	2.238	0.270
01/06/2017	1.719	3.309	2.528	3.017	1.685	2.293	0.329	2.389	0.450
15/06/2017	1.719	2.905	1.407	2.947	1.509	2.153	0.188	2.128	0.167
01/07/2017	1.719	2.933	1.475	2.948	1.512	2.032	0.098	2.095	0.141
15/07/2017	2.947	3.462	0.264	2.964	0.000	2.032	0.839	2.280	0.445
01/08/2017	2.947	3.025	0.006	2.948	0.000	2.037	0.829	2.421	0.277
15/08/2017	2.947	3.518	0.326	3.319	0.138	2.362	0.342	2.635	0.098
01/09/2017	2.947	3.519	0.327	3.091	0.021	2.368	0.336	2.748	0.040
15/09/2017	2.947	3.471	0.274	2.877	0.005	2.470	0.228	2.660	0.083
01/10/2017	2.947	3.487	0.291	2.963	0.000	2.470	0.228	2.710	0.056
15/10/2017	3.878	3.228	0.423	2.900	0.957	2.040	3.378	2.204	2.802
01/11/2017	3.878	3.375	0.253	2.469	1.984	2.042	3.371	2.677	1.441
15/11/2017	3.878	3.619	0.067	2.458	2.015	2.603	1.625	2.761	1.248
01/12/2017	3.878	3.800	0.006	2.737	1.302	2.607	1.616	2.790	1.183
15/12/2017	3.878	3.896	0.000	2.679	1.437	2.607	1.616	2.846	1.065
01/01/2018	3.878	3.940	0.004	2.723	1.333	2.605	1.621	2.801	1.160
15/01/2018	3.779	3.803	0.001	3.057	0.522	2.605	1.379	2.850	0.863
01/02/2018	3.779	3.292	0.237	3.072	0.500	2.581	1.436	2.643	1.290
15/02/2018	3.779	3.475	0.092	3.089	0.477	2.581	1.436	2.736	1.088
01/03/2018	3.779	3.367	0.170	3.075	0.496	2.580	1.437	2.689	1.188
15/03/2018	3.779	3.843	0.004	3.207	0.328	2.562	1.480	2.766	1.026
01/04/2018	3.779	3.993	0.046	3.249	0.281	2.566	1.471	2.908	0.760
15/04/2018	2.701	4.015	1.726	3.400	0.487	2.566	0.018	2.921	0.048
01/05/2018	2.701	3.265	0.318	3.165	0.215	2.421	0.078	2.671	0.001
15/05/2018	2.701	3.517	0.665	3.303	0.362	2.541	0.026	2.623	0.006
01/06/2018	2.701	3.583	0.777	3.383	0.465	2.539	0.026	2.600	0.010
15/06/2018	2.701	3.408	0.499	3.275	0.328	2.539	0.026	2.571	0.017
01/07/2018	2.701	3.347	0.417	3.251	0.302	2.386	0.100	2.456	0.060
15/07/2018	2.117	3.404	1.656	3.058	0.885	2.386	0.072	2.415	0.088
01/08/2018	2.117	3.537	2.014	3.062	0.893	2.426	0.095	2.971	0.729
15/08/2018	2.117	3.639	2.315	3.113	0.991	2.712	0.354	2.613	0.246
01/09/2018	2.117	3.615	2.243	3.074	0.916	2.290	0.030	2.606	0.239
15/09/2018	2.117	3.812	2.873	3.216	1.206	2.714	0.356	2.711	0.352
01/10/2018	2.117	3.872	3.079	3.176	1.121	2.713	0.355	2.770	0.426
15/10/2018	1.320	3.919	6.753	3.490	4.711	2.713	1.942	2.785	2.146
01/11/2018	1.320	3.371	4.206	3.202	3.544	2.288	0.937	2.569	1.561
15/11/2018	1.320	3.686	5.600	3.256	3.749	2.706	1.923	2.777	2.123
01/12/2018	1.320	3.726	5.788	3.233	3.662	2.706	1.923	2.773	2.112
15/12/2018	1.320	3.501	4.757	3.028	2.920	2.288	0.937	2.627	1.708
01/01/2019	1.320	3.476	4.650	3.003	2.833	2.287	0.935	2.613	1.674
15/01/2019	2.932	3.588	0.430	3.141	0.044	2.705	0.052	2.687	0.060
01/02/2019	2.932	3.420	0.238	3.068	0.018	2.287	0.416	2.480	0.205
15/02/2019	2.932	2.993	0.004	2.545	0.150	1.952	0.960	2.031	0.812
01/03/2019	2.932	3.098	0.027	2.826	0.011	2.122	0.656	2.343	0.347
15/03/2019	2.932	2.949	0.000	2.671	0.068	2.234	0.488	2.255	0.459
01/04/2019	2.932	3.019	0.007	3.101	0.028	2.231	0.491	2.124	0.654
15/04/2019	1.491	3.042	2.406	2.651	1.345	1.969	0.228	2.071	0.336
01/05/2019	1.491	3.596	4.432	2.728	1.529	2.295	0.646	2.588	1.202
15/05/2019	1.491	3.441	3.804	2.787	1.680	2.321	0.688	2.487	0.991
01/06/2019	1.491	3.514	4.090	2.694	1.447	2.320	0.687	2.522	1.064
15/06/2019	1.491	3.377	3.556	2.419	0.860	2.320	0.687	2.449	0.917
01/07/2019	1.491	3.465	3.897	2.554	1.129	2.321	0.688	2.493	1.005
15/07/2019	2.572	3.350	0.605	2.781	0.044	2.321	0.063	2.360	0.045
01/08/2019	2.572	3.839	1.606	3.063	0.241	2.629	0.003	2.746	0.030
15/08/2019	2.572	2.935	0.132	2.621	0.002	2.009	0.317	2.101	0.222
01/09/2019	2.572	2.894	0.103	2.731	0.025	2.009	0.318	1.991	0.338
15/09/2019	2.572	3.095	0.273	2.835	0.069	2.009	0.318	2.000	0.327
01/10/2019	2.572	3.238	0.443	2.707	0.018	2.008	0.318	2.023	0.302
15/10/2019	2.366	3.103	0.544	2.693	0.107	2.008	0.128	2.014	0.124
01/11/2019	2.366	3.229	0.745	2.749	0.147	1.956	0.168	2.196	0.029
15/11/2019	2.366	3.005	0.409	2.448	0.007	1.956	0.168	2.041	0.105
01/12/2019	2.366	3.063	0.486	2.423	0.003	1.958	0.166	2.172	0.038
15/12/2019	2.366	3.109	0.553	2.443	0.006	1.958	0.166	2.095	0.073
01/01/2020	2.366	3.236	0.758	2.518	0.023	1.958	0.166	2.131	0.055

The GDP and the prediction, squared error of a Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and Hybrid Approach for the vintages of the data with starting date 1980 with lag variables

Table 17: The results of the vintages of the data with starting date 2000 with lag variables

Date	GDP	Dynamic Factor Model		Random Forest		Extreme Gradient Boosting		Hybrid	
		Prediction	$Error^2$	Prediction	$Error^2$	Prediction	$Error^2$	Prediction	$Error^2$
01/01/2017	2.541	2.599	0.003	2.350	0.036	1.931	0.372	2.253	0.083
15/01/2017	2.282	2.476	0.038	2.390	0.012	2.138	0.021	2.260	0.000
01/02/2017	2.282	1.850	0.186	2.295	0.000	2.032	0.063	1.942	0.115
15/02/2017	2.282	2.426	0.021	2.498	0.047	2.316	0.001	2.442	0.026
01/03/2017	2.282	2.439	0.025	2.517	0.055	2.311	0.001	2.439	0.025
15/03/2017	2.282	2.677	0.156	2.595	0.098	2.690	0.167	2.578	0.088
01/04/2017	2.282	2.735	0.205	2.615	0.111	2.676	0.156	2.635	0.125
15/04/2017	1.719	2.574	0.731	2.508	0.622	2.326	0.369	2.443	0.524
01/05/2017	1.719	2.335	0.380	2.245	0.277	1.656	0.004	2.010	0.085
15/05/2017	1.719	2.543	0.679	2.516	0.636	2.676	0.917	2.486	0.588
01/06/2017	1.719	2.604	0.783	2.455	0.542	2.651	0.869	2.596	0.769
15/06/2017	1.719	2.348	0.395	2.259	0.292	2.083	0.133	2.186	0.218
01/07/2017	1.719	2.376	0.432	2.253	0.285	2.050	0.110	2.094	0.141
15/07/2017	2.947	2.688	0.067	2.618	0.109	2.666	0.079	2.670	0.077
01/08/2017	2.947	2.506	0.194	2.354	0.353	2.484	0.214	2.340	0.369
15/08/2017	2.947	2.822	0.016	2.502	0.198	2.685	0.069	2.778	0.029
01/09/2017	2.947	2.898	0.002	2.512	0.189	2.754	0.037	2.867	0.007
15/09/2017	2.947	2.879	0.005	2.418	0.281	2.729	0.048	2.920	0.001
01/10/2017	2.947	2.880	0.004	2.390	0.310	2.716	0.053	2.904	0.002
15/10/2017	3.878	2.580	1.685	2.339	2.368	2.493	1.918	2.620	1.581
01/11/2017	3.878	2.698	1.391	2.484	1.942	2.698	1.392	2.872	1.013
15/11/2017	3.878	2.874	1.007	2.491	1.924	2.586	1.669	2.880	0.995
01/12/2017	3.878	2.926	0.907	2.560	1.738	2.650	1.507	2.873	1.009
15/12/2017	3.878	3.020	0.736	2.546	1.775	3.185	0.479	2.932	0.895
01/01/2018	3.878	3.041	0.701	2.565	1.725	3.209	0.447	2.897	0.963
15/01/2018	3.779	2.939	0.706	2.763	1.032	3.155	0.389	2.879	0.810
01/02/2018	3.779	2.677	1.214	2.777	1.005	2.774	1.011	2.841	0.881
15/02/2018	3.779	2.861	0.843	2.787	0.985	2.727	1.107	2.914	0.748
01/03/2018	3.779	2.803	0.953	2.750	1.060	2.697	1.171	3.057	0.521
15/03/2018	3.779	3.133	0.418	2.755	1.049	3.043	0.543	3.134	0.416
01/04/2018	3.779	3.212	0.322	2.837	0.887	2.933	0.716	2.867	0.832
15/04/2018	2.701	3.074	0.138	2.862	0.026	2.948	0.061	2.977	0.076
01/05/2018	2.701	2.731	0.001	2.720	0.000	2.714	0.000	2.824	0.015
15/05/2018	2.701	2.895	0.037	2.762	0.004	3.130	0.184	2.863	0.026
01/06/2018	2.701	2.915	0.046	2.631	0.005	3.118	0.173	2.978	0.076
15/06/2018	2.701	2.824	0.015	2.569	0.018	2.748	0.002	2.819	0.014
01/07/2018	2.701	2.786	0.007	2.546	0.024	2.645	0.003	2.640	0.004
15/07/2018	2.117	2.728	0.373	2.819	0.492	2.639	0.272	2.602	0.235
01/08/2018	2.117	2.948	0.690	2.816	0.488	3.485	1.869	3.021	0.817
15/08/2018	2.117	2.911	0.630	2.452	0.112	2.316	0.039	2.415	0.089
01/09/2018	2.117	2.887	0.592	2.528	0.168	2.340	0.049	2.431	0.098
15/09/2018	2.117	2.984	0.751	2.474	0.127	3.031	0.835	2.537	0.176
01/10/2018	2.117	3.011	0.799	2.569	0.204	3.016	0.807	2.655	0.289
15/10/2018	1.320	3.138	3.306	3.179	3.457	3.291	3.886	3.051	2.996
01/11/2018	1.320	2.799	2.187	2.741	2.020	2.372	1.106	2.569	1.559
15/11/2018	1.320	2.963	2.701	2.871	2.405	3.180	3.461	2.972	2.731
01/12/2018	1.320	2.989	2.788	2.878	2.429	3.138	3.306	3.038	2.953
15/12/2018	1.320	2.854	2.355	2.688	1.871	2.215	0.802	2.600	1.640
01/01/2019	1.320	2.835	2.296	2.658	1.790	2.153	0.695	2.670	1.822
15/01/2019	2.932	2.865	0.005	2.822	0.012	2.923	0.000	2.821	0.012
01/02/2019	2.932	2.846	0.007	2.759	0.030	2.840	0.009	2.751	0.033
15/02/2019	2.932	2.646	0.082	2.424	0.259	2.248	0.468	2.340	0.351
01/03/2019	2.932	2.768	0.027	2.578	0.125	2.535	0.158	2.639	0.086
15/03/2019	2.932	2.596	0.113	2.346	0.343	2.138	0.631	2.330	0.362
01/04/2019	2.932	2.668	0.070	2.375	0.311	2.157	0.602	2.414	0.268
15/04/2019	1.491	2.588	1.203	2.405	0.835	2.103	0.374	2.446	0.912
01/05/2019	1.491	3.149	2.747	2.653	1.349	2.768	1.630	3.386	3.590
15/05/2019	1.491	2.762	1.616	2.353	0.743	2.056	0.319	2.307	0.666
01/06/2019	1.491	2.806	1.730	2.305	0.663	2.027	0.287	2.339	0.719
15/06/2019	1.491	2.714	1.496	2.165	0.454	1.964	0.224	2.418	0.859
01/07/2019	1.491	2.778	1.656	2.038	0.299	2.036	0.297	2.487	0.991
15/07/2019	2.572	2.771	0.040	2.531	0.002	2.626	0.003	2.589	0.000
01/08/2019	2.572	3.204	0.400	2.879	0.094	3.120	0.300	3.279	0.500
15/08/2019	2.572	2.517	0.003	2.415	0.025	2.265	0.095	2.364	0.043
01/09/2019	2.572	2.484	0.008	2.416	0.024	2.158	0.172	2.329	0.059
15/09/2019	2.572	2.572	0.000	2.432	0.020	2.173	0.159	2.605	0.001
01/10/2019	2.572	2.667	0.009	2.472	0.010	2.268	0.093	2.587	0.000
15/10/2019	2.366	2.410	0.002	2.187	0.032	2.098	0.072	2.149	0.047
01/11/2019	2.366	2.516	0.023	2.008	0.128	1.891	0.226	2.148	0.047
15/11/2019	2.366	2.430	0.004	2.095	0.073	2.004	0.131	2.156	0.044
01/12/2019	2.366	2.451	0.007	2.114	0.063	1.958	0.166	2.160	0.042
15/12/2019	2.366	2.567	0.040	2.054	0.097	2.003	0.132	2.365	0.000
01/01/2020	2.366	2.633	0.071	2.043	0.104	1.980	0.149	2.328	0.001

The GDP and the prediction, squared error of a Dynamic Factor Model, Random Forest, Extreme Gradient Boosting and Hybrid Approach for the vintages of the data with starting date 2000 with lag variables