ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Bachelor Thesis Econometrie & Operationele Research

Forecasting African GDP Growths Using Factor Models and Machine Learning Methods

Author:
Solange Eersel
447252

Supervisor:
P. H. B. F. Franses
Second assessor:
A. M. Schnuncker

July 7, 2019

Abstract

This paper analyses a hybrid forecast method to estimate forecasts for Ghana's Gross Domestic Product (GDP) growth. We use data of 51 African countries of the time period 1963-2016. *Kim & Swanson (2018)* [8] explored different methods to estimate forecasts using big datasets. In their research they use hybrid methods, which are mixtures of well known factor models such as PCA, ICA and SPCA, and data shrinkage methods, such as Boosting. We use a hybrid model, by applying Boosting followed by ICA, from *Kim & Swanson*, to predict forecasts and answer our research question, whether we can accurately forecast Ghana's GDP growth using lagged GDP growths from other African countries. First we will use simulations to obtain the accuracy of our hybrid method. Next we compare our forecast results with those of a benchmark model, using the Mean Square Forecast Error (MSFE), and we can conclude that our hybrid model makes better predictions for Ghana's GDP growth, than the benchmark model. We also conclude that using lags of different countries as predictors can help create a good forecasting model.

Contents

1 Introduction

Africa is the world's poorest inhabitant continent, but changes are happening rapidly. Africa is economically the second fastest growing region since 2000 and the continent is expected to reach "middle income" status within ten years with the current economic growth. This is why the growth of the African economy has been a very interesting topic these last few years. A measure of the economic growth is the Gross Domestic Product (GDP), which is a broad measurement of a nation's overall economic activity[1]. One very interesting question is, whether the high GDP growth of the different African countries are somehow connected to each other. If this is the case, we can use this information for future predictions of the GDP growth. This is valuable information because it allows policymakers, businesses and economists to analyze the affects on the growth and on the overall economy and it can be a helpful tool for reaching financial goals.[2].

Using Big Data to forecast low frequency macroeconomic variables, such as the GDP growth, is a common research topic that has been researched for example in the paper of *Kim & Swanson (2018)* [8]. The problem with big datasets is that they tend to have large dimensions. A large dimension can cause problems, because adding too many variables to a model can cause over-fitting. This is the reason why dimension reduction is of such big importance in big datasets. There are several methods to reduce the number of variables for creating a forecasting model. One popular and well-known method is with the use of factor models, this method makes use of diffusion indices and is used in the paper of *Stock & Watson (2002a)* [9]. There are different methods to find unobserved factors, such as Principal Component Analysis (PCA), Independent Component Analysis (ICA) and Sparse Principal Component Analysis (SPCA). All of these methods differ slightly, but they all come down to finding a small set of common factors from a large set of unobserved variables, which can best explain the complete dataset. Besides factor models we can also reduce the dimension of a big dataset using shrinkage methods, one of these shrinkage methods is Boosting, which is a machine learning method. One very interesting method in the paper of *Kim & Swanson (2018)* [8] is a combination of a factor-type method and a reduction shrinkage technique. Instead of using just factor models or shrinkage methods they have used a hybrid method, which they concluded to perform very well compared to traditional forecasting methods.

Now that we know that hybrid methods can predict good forecasting models for large datasets, it would be interesting to know if these techniques would also work for forecasting African GDP growths. This leads us to the research question, whether we can use dimension reduction techniques, such as a hybrid between factor-methods and machine

[1]Investopedia: Gross Domestic Product (GDP) [12]
[2]Our World in Data: Economic growth [11]

learning shrinkage methods, to forecast the GDP growth of Ghana, using the lagged GDP growth from other African countries. We would also want to know, after creating such a forecast model, if we can validly explain why certain variables were selected by the hybrid method and if these selected countries have a clear connection to Ghana.

To research this, we will first investigate the accuracy of a hybrid model, which consists of Boosting followed by ICA, with different Data Generating Processes (DGPs). Next we will forecast Ghana's GDP growth, using a dataset containing lagged GDP growths of many African countries. To compare how well our model predicts GDP growth forecasts, we will use the Mean Square Forecast Error (MSFE) to compare the forecasts of our hybrid model with those of a baseline forecasting model. Our empirical results show that our hybrid model predicts better forecasts than the baseline model for the given forecasting period.

Our thesis is structured as follows, in section 2 we give more information about the African economy, the literature we used for the research is described in section 3, followed by a clear description of the data in section 4. Next, section 5 gives an extensive explanation of the methodology and finally we will show our results and give a conclusion to our research in section 6 and 7.

2 African Economy

Since the industrial revolution, which created a big gap between rich and poor countries, Africa is world's poorest inhabitant continent. Their low GDPs are caused by many different factors, such as political corruption and colonialism. Most countries are very secluded from each other. This is caused by bad infrastructure, language barriers and the big distances between different countries. This seclusion leads to less trade between African countries and other continents, this has a negative effect on the GDP, because import and export have a huge impact on a country's economy. However, things are changing rapidly and the African economy is currently booming. This recent growth is mainly caused by sales in commodities, services and manufacturing, and also by the increasing political stability. Some oil-rich and mineral rich countries like Algeria, Libya and Botswana belong to the richest nations in the world, while others like Zimbabwe and the Democratic Republic of Congo remain very poor because of political corruption and warfare. Overall we can say that the economic growth for most African countries is very positive.

One very interesting country to look at is Ghana. Ghana is a politically, economically and demographically diverse country. Ghana's economic growth stabilized in the early 1990s and because of this Ghana gained middle-income status in 2011. Since 2005 Ghana's economy grew significantly, this growth was caused by the country's main com-

modity exports, gold and cocoa. Another growth boost took place in 2011 with the kickoff of commercial oil production. Besides these things Ghana is also rich in diamonds, manganese ore, bauxite, and oil. In 2015 the growth declined due to declining commodity prices, energy rationing, and a fiscal crisis in 2013. After this slight dip the growth picked up again and Ghana's economy is currently still very strong[3].

Now that we know that Ghana is one of Africa's leading economic countries, it would be interesting to know, what will happen to the future of their economic growth. And since many African economies are significantly growing, it is valuable information to know which economies are related to each other, these connections could be caused by similar resources or because of trade between the countries. If we know that Ghana's GDP growth follows that of other economies, we can use this knowledge for future predictions.

3 Literature

Many papers already discuss different methods of dimension reduction, based on diffusion indices and other shrinkage methods, for big datasets.

The main paper that we will use for our research is that of *Kim & Swanson (2018)* [8]. In their paper they have a big dataset consisting of many macroeconomic time series, which they want to use to forecast 11 different macroeconomic variables. Because they have such a big dataset they use different dimension reduction techniques, to find models that can best predict forecasts for their 11 macroeconomic variables. They combine different techniques with each other to find the best model. They make use of diffusion index models, using PCA and ICA, and combine these techniques with other shrinkage methods. PCA is the most common known way to create factors from unobserved variables, but they also use ICA to see if this method maybe has some additional added value. In contrast to PCA, which estimates factors that have the largest shares of variance, ICA looks for factors that have the largest independence between each other.

The machine learning shrinkage method that we will use in our research, is Boosting. In the paper of *Kim & Swanson (2018)* [8] they use the"Component-Wise L_2Boosting" algorithm as described in *Bai & Ng (2009)* [3]. This is a dimension reduction technique where they use "weak learners" to find predictors that best fit their model, which they then use to predict their variable of interest with. This variable of interest is in our case, Ghana's GDP growth.

Kim & Swanson (2018) [8] make use of different specification methods to obtain their forecast model. They first use diffusion index models to obtain factors from their complete dataset, followed by shrinkage methods which they then only use on their estimated

[3]The World Bank Group: Ghana's growth history, New growth momentum since the 1990s helped put Ghana at the front of poverty reduction in Africa [13]

factors. They also use opposite methods where they first use shrinkage methods to filter their complete dataset, followed by constructing factors from the filtered variables. With the latter method being the specification method we will use in our research.

The forecasting model that we ended up using is that of *Kim & Swanson (2018)* [8] which is based on the model of *Stock & Watson (2002a,b)* [9] [10], *Bai & Ng (2006)* [2] and *Kim & Swanson (2014a)* [7]. This model is basically an Autoregressive forecasting model with added factors obtained from the specification methods described above.

After *Kim & Swanson (2018)* [8] used various methods, with different combinations of factor and shrinkage methods, they found out that taking ICA into consideration can get better results than PCA for larger time horizons. This is the reason we will use ICA. They also concluded that simple time series models do not dominate hybrid methods, that use both factor methods and machine learning shrinkage methods. These hybrid methods can be very promising for forecasting macroeconomic variables. Their research is similar to our own research in the sense that they use a big dataset containing monthly low-frequency data of many different macroeconomic time series, to predict a forecast for 11 macroeconomic variables. While we use a dataset containing yearly time series data of the GDP growth of many different countries, to predict a forecast for the GDP growth of one country. Because their research is somewhat similar to ours we will use a similar approach and similar techniques for our own research.

4 Data

The data we use, provides a complete balanced panel data set on annual observations on 52 African Gross Domestic Product (GDP) growth time series for the period 1963-2016 ($N = 52$, $T = 54$). For 34 countries the full data can be obtained from the World Bank[4] and for 18 countries there are missing data

For the missing observations, estimates of real GDP are imputed using a Principal Components regression. The imputation method for the data is described in *Franses & Vasilev*[4]. They collect data, again from the World Bank, in three categories, that is, demographic variables, production variables and financial variables.

They followed a procedure where for each country containing missing data, they collected variables from the three categories and created principal components (PCs) for each of these categories. The number of PCs used, is chosen using the Kaiser rule, the cumulative proportion of the variance and the individual proportion of the variance is in excess 5%. The variables they obtain, are then included in a Principal Component regression for a sub-sample. Next, out-of-sample forecasts are made for this sub-sample. If these predictions are adequate, meaning with Mean Squared Prediction Error about

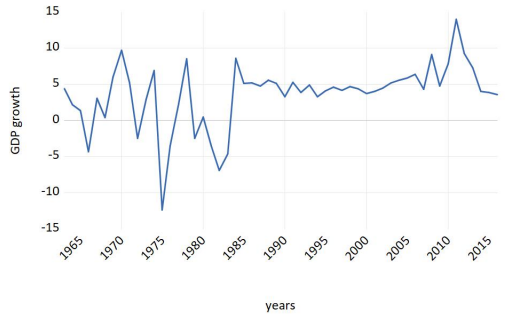
[4]World Bank: GDP growth (annual %)

equal to the in-sample Mean Squared Error, the Principal Component regression is run for the full sample. Finally they can make predictions for the missing observations.

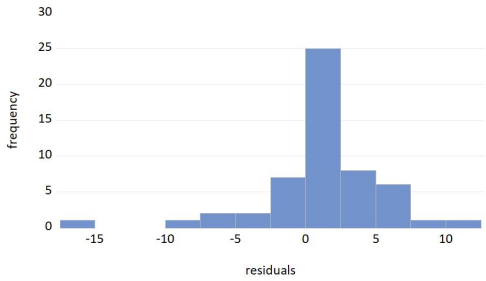
Franses & Vasilev[4] followed this procedure for all 18 countries with missing data, where each country may have different sets of demographic variables, production variables and financial variables.

4.1 Descriptive Statistics of Ghana

In figure 1 below the GDP growth time series for the period 1963-2016 and the residuals for an Autoregressive model with p=1 lag (AR(1) model) of this time-series are shown for the country Ghana.



(a) Time series of Ghana's GDP growth of the period 1963-2016.



(b) Histogram of the residuals of the AR(1) model

Figure 1: Ghana GDP growth time series and residuals

From figure 1a we can see that there were some fluctuations in the time series, with for example dips in 1975 and 1982 and a peak in 2011. From figure 1b we can conclude that these peaks and dips do not have much effect on our AR(1) model, because the distribution of the residuals have a skewness of -1.37, which is not an alarming large number. This means that there are no extreme outliers we have to be cautious for when estimating a model and we will not take structural breaks into consideration. Table 1 below contains the statistics of the GDP growth time series and gives a clear explanation of figure 1a.

Table 1: Descriptive statistics of Ghana's GDP growth of different periods.

	observations	mean	std. dev.	maximum	minimum
1963-1990	28	1.821	5.200	9.700	-12.400
1990-2005	14	4.457	0.649	5.600	3.300
2005-2016	12	6.708	3.051	14.000	3.600

In the table we can see that for the period 1963-1990 there are some fluctuations (dips

in 1975 and 1982) which are explained by the higher standard deviation of 5.2 and high and low maximum and minimum GDP growths. While the period 1990-2005 was really steady, with an average growth of 4.46%, a low standard deviation of 0.65 and a maximum and minimum close to the mean value. Lastly we can see that there was significantly more growth in the period 2005-2016, with a higher mean 6.71%, however there are also more fluctuations than the period before, with a peak in 2011 followed by a slight dip in 2015, this can be explained by the increased standard deviation of 3.05.

5 Methodology

In this section we will explain the different methods we will use for our research. We will use specification method 2 from *Kim & Swanson (2018)* [8], which means that we will first use Boosting as a variable selection tool on the complete dataset and after filtering our variables using Boosting, we will use ICA on these selected variables. Next we will use different DGPs to obtain the accuracy of our hybrid method. Finally we can use our hybrid method to predict out-of-sample forecasts and compare the predictions of our hybrid model with those of a baseline model.

5.1 Boosting

To reduce the dimension of the data, we will first use a machine learning data shrinkage method, on the complete dataset X. With X being the $T \times N$ matrix containing time series for all N predictors. The shrinkage method used in our research is the Boosting method from the paper of *Kim & Swanson (2018)* [8]. Boosting constructs a model consisting of different potential predictors that best fit Y, which in our case, is the $T \times 1$ vector containing the time series of Ghana's GDP growth. To find this good fitting model, Boosting makes use of "weak learners", to produce one final "committee". A "weak learner" is a simple model that only needs a few variables and has a large bias relative to the variance. The Boosting algorithm we use is very similar to the "Component-Wise L_2Boosting" algorithm of *Bai & Ng (2009)* [3]. In their paper they use the conditional mean as their "weak learner" $\hat{\mu}$.

Before starting the algorithm we first need to choose the potential predictors z_t, this matrix contains all possible variables that can be used to fit our model for Y. The z_t can conclude current, lagged and functions of y_t and X_t. For the Boosting algorithm we use similar potential predictors as *Bai & Ng* [3], where $z_t = (Z_t, Z_{t-1}, \ldots, Z_{t-pmax})'$, is a $pmax \times 1$ matrix with a chosen value for $pmax$ and $Z_t = (X_{1,t-1}, \ldots, X_{N,t-1})'$. Here Z_t is a $N \times 1$ matrix. The total number of elements in z_t is then equal to $R = N \times pmax$, where each potential predictor $z_{.,j}$ is a $T \times 1$ time series for the $j^{th} \in R$ element from the

complete set z_t.

The potential predictors that best fit the model for Y are selected using an ordinary least squares (OLS) regression, the predictors that obtain the smallest squared residuals are chosen for our model. The Boosting algorithm fits learners to the model, using one potential predictor at a time, so in each OLS regression we will use only one $z_{.,j}$, from the complete potential set z_t. The predictor we select in the i^{th} round, and thus obtained the smallest squared residual in the i^{th} round, is denoted by $z_{.,j_*^i}$.

Before starting the algorithm we will set $\hat{\mu}^0 = \bar{Y}$, for $t = 1, ..., T$

Algorithm 1 Component Wise L_2 Boosting
for $i = 1, \ldots, M$

1. $\forall t = 1, \ldots, T$ compute the "current residual" $u_t = y_t - \hat{\mu}^{i-1}$
2. $\forall j = 1, \ldots, R$ regress the $T \times 1$ current residual u on $z_{.,j}$ to obtain β_j, use an OLS regression to obtain this $\beta_j = (z'_{.,j} z_{.,j})^{-1} z'_{.,j} u$
3. $\forall j = 1, \ldots, R$ compute the residual $\hat{e_j} = u - z_{.,j}\beta_j$, compute the sum of squared residuals $SSR_j = \hat{e_j}' \hat{e_j}$. Let j_*^i denote the column selected at the i^{th} iteration where the minimum SSR_j is obtained, $SSR_{j_*^i} = \min_{j \in [1,\ldots,R]} SSR_j$.
4. Let $g_*^i = z_{.,j_*^i} \beta_{j_*^i}$
5. Update $\hat{\mu}^i = \hat{\mu}^{i-1} + \nu g_*^i$, where $0 \leq \nu \leq 1$ is the step length.

All the potential predictors $z_{.,j_*^i}$ that are selected, will be saved in the matrix W, which is a $T \times L$ matrix. All the duplicates of the potential predictors that were selected multiple times, will be removed from W, so we finally end up with L unique predictors. The set of predictors W that are selected by the Boosting algorithm will be used to construct factors using Independent Component Analysis described in section 5.2.

5.1.1 Choosing Number of Iterations in Boosting Algorithm

One important thing we need to keep in mind, is choosing a number of iterations M that does not cause over-fitting of the algorithm. *Bai & Ng (2009)* [3] propose a stopping parameter M using the information criterion,

$$IC(i) = log[\hat{\sigma}^{i^2}] + \frac{log(T) \cdot df^i}{T}, \tag{1}$$

where $\hat{\sigma}^{i^2} = \sum_{t=1}^{T} (y_t - \hat{\mu}^i)^2$. The degrees of freedom are computed using $df^i = trace(B^i)$, where $B^i = B^{i-1} + \nu \boldsymbol{P}^{(i)}(I_T - B^{i-1}) = I_T - \prod_{h=0}^{i} (I_T - \nu \boldsymbol{P}^{(h)})$, with $\boldsymbol{P}^{(i)} = z_{.,j_*^i} (z'_{.,j_*^i} z_{.,j_*^i})^{-1} z'_{.,j_*^i}$.

The starting value is given by $B^0 = \iota_T \iota_T'/T$, with ι_T a $T \times 1$ vector of ones. Next, we will find the iteration where the smallest value of the information criterion is reached,

$$M = \underset{i}{\operatorname{argmin}}\, IC(i). \tag{2}$$

We will use M as our number of iteration in Algorithm 1 in section 5.1.

5.2 Independent Component Analysis

After we have selected a subset of variables W, from the big dataset X, using the Boosting algorithm from section 1, we will use these filtered predictors to construct factors by performing Independent Component Analysis (ICA) from the paper of *Kim & Swanson (2018)* [8]. ICA assumes that the variables that are selected with Boosting, depend on several unobserved factors. The main assumption that ICA makes, which differs from the well-known PCA, is that the components are all statistically independent. ICA starts with source data S, which is statistically independent and mixed by the mixing matrix Ω. So our set of variables W is a combination between S and Ω,

$$W = S\Omega. \tag{3}$$

Here W is observed and S and Ω are unobserved. Because S and Ω are both unobserved we need another matrix that we can estimate. This will be our demixing matrix Ψ, this matrix transforms the observed W into the factors F.

$$F = W\Psi \tag{4}$$

We assume that both Ω and Ψ are square matrices and that $\Psi = \Omega^{-1}$ such that F is identical to S.

The main goal of ICA is to find independent components, this can be done by estimating the demixing matrix Ψ. Because there is no direct measure for independence, we will use "nongaussianity" as a measure for independence. We use nongaussianity because a gaussian variable is not able to create independent variables, so the contrary to gaussianity, which is nongaussianity, will then lead to independence. Entropy is a useful tool for measuring nongaussianity. We will use a modified version of entropy called negentropy, which is the optimal estimator for nongaussianity. The approximation that we will use from *Kim & Swanson (2018)*[8] for negentropy is,

$$N(F) \propto [E\{G(F)\} - E\{G(\nu)\}]^2 \tag{5}$$

with,

$$G(y) = \frac{1}{a_1} log\, cosh\, a_1 y, \quad where\, 1 \leq a_1 \leq 2. \tag{6}$$

The algorithm we will use for our research is the "FastICA" algorithm from *Hyvärinen & Oja (2000)*[5], this algorithm efficiently minimizes negentropy. "FastICA" follows maximal nongaussianity of the matrix $W\Psi = \{W\Psi_1, \ldots, W\Psi_L\}$, with Ψ_j uncorrelated column vectors of Ψ. The goal of the FastICA algorithm is to find an optimal vector Ψ_j, such that $W\Psi_j$ maximizes nongaussianity.

The variance of $W\Psi_j$ is constrained to be unity, so the norm of Ψ_j is also constrained to be unity. Denote g as the derivative of G in equation 6. When equation 5 is maximized, the maximum is obtained at $E\{G(F)\} = E\{G(W\Psi)\}$. Under the constraint of unity we have, $E\{G(X\Psi_j)^2\} = ||\Psi_j|| = 1$, we can obtain the optimum of $E\{G(W\Psi)\}$ at $E\{G(W\Psi)\} - \lambda\Psi_j = 0$. Solving this leads to,

$$\Psi^* = E\{Wg(W\Psi_j)\} - E\{g'(W\Psi_j)\}\Psi_j, \tag{7}$$

which is the demixing matrix yielding components with minimized negentropy and thus maximal nongaussianity. A more detailed explanations of these calculations can be found in the paper of *Hyvärinen & Oja (2000)*[5]. The "FastICA" algorithm we use to calculate the optimal demixing matrix Ψ, can be found below.

Algorithm 2 FastICA

1. Choose initial demixing vector Ψ, given from the loadings of r ordinary principal components.
2. For $j = 1, \ldots, r$
   (a) find demixing vectors that obtain components with minimum negentropy.
   $\Psi_j^* = E\{Wg(W\Psi_j)\} - E\{g'(W\Psi_j)\}\Psi_j$.
   (b) Update Ψ_j, $\Psi_j = \Psi_j^* / \parallel \Psi_j^* \parallel$. If convergence is not reached, go back to step 2a.
   If convergence is reached set $\Psi_j^+ = \Psi_j$.
3. To make sure that the j independent components are uncorrelated for $j \geq 2$, we set $\Psi_j^+ = \Psi_j^+ - \sum_{h=1}^{j-1} \Psi_j^{+\prime}\Psi_h\Psi_h$ followed by $\Psi_j^+ / \sqrt{\parallel \Psi_j^{+\prime}\Psi_j^+ \parallel}$.

Once we have estimated Ψ, the independent components are equal to $F = W\Psi$. And we can acquire the impact that a specific variable from W has on the factors F using the demixing matrix Ψ.

5.3 Choosing Number of Components

In factor analysis it is a crucial step to choose the number of components. Here we again have the problem that too many factors can lead to over-fitting of the model and too few might cause the loss of useful information. Because the calculation for the number of components for ICA is quite complicated, we will simply use a selection criterion on components constructed using PCA and use this number of components in our FastICA algorithm. There are many different techniques for determining the number of components for PCA. In *Kim & Swanson (2014a)*[7] they use a selection criterion based on the work of *Bai & Ng (2002)*[1]

$$PC(r) = V(r, \hat{F}_{r,t}) + r\hat{\sigma}^2 \left(\frac{(L+T-r)ln(LT)}{LT} \right), \quad \text{for } r = 1, \dots L. \tag{8}$$

Here L is the number of predictors that were chosen in the Boosting algorithm.

Because a factor model is linear and $\hat{F}$ is observed after estimation, we can estimate the factor loadings λ_i by applying an OLS regression to each equation. After obtaining the factor loadings, we can calculate $V(r, \hat{F}_{r,t})$, which is the sum of squared residuals and measures how well the model fits,

$$V(r, \hat{F}_{r,t}) = \min_{\Lambda} \frac{1}{LT} \sum_{i=1}^{L} \sum_{t=1}^{T} (X_{it} - \lambda'_{i,r} F_{r,t})^2. \tag{9}$$

Since we lose efficiency when adding too many factors to our model, we need a penalty term which will increase when we add more factors to the model. The penalty term we will use is $\hat{\sigma}^2 \left(\frac{(L+T-r)ln(LT)}{LT} \right)$ with $\hat{\sigma}^2$ given below,

$$\hat{\sigma}^2 = V(r_{max}, \hat{F}_{r_{max},t}), \quad \text{with, } r_{max} = L. \tag{10}$$

After calculating our selection criterion for all the possible numbers of components L, we will choose the number of components where the smallest $PC(r)$ is obtained,

$$r^* = \underset{r \in [1,L]}{\operatorname{argmin}} \, PC(r). \tag{11}$$

Finally we can use r^* as our initial number of components in the FastICA algorithm.

5.4 Hybrid Forecasting Model

After using Boosting and ICA to reduce the dimension of our variables, we will use the following model from *Kim & Swanson (2018)* [8] based on the model of *Stock & Watson*

(2002a,b) [9] [10], *Bai & Ng (2006)* [2] and *Kim & Swanson (2014a)* [7] to forecast the GDP growth of our country of interest,

$$\hat{Y}_{t+1} = D_t\beta_D + \hat{F}_{i,t}\beta_F + \varepsilon_{t+h}. \tag{12}$$

In this equation $\hat{Y}_{t+1}$ is the one-step-ahead forecast of the GDP growth of our country of interest, Ghana. D_t is a 1-year lag of Ghana's GDP growth, $\hat{F}_{i,t}$ are our estimated factors, obtained by using Boosting followed by ICA and ε_{t+h} is a disturbance term. We will first estimate $\hat{\beta}_D$ and $\hat{\beta}_F$ using an in-sample dataset and regressing Y_t on D_{t-1} and $\hat{F}_{i,t-1}$. Next we will use the estimates $\hat{\beta}_D$ and $\hat{\beta}_F$ and the predictors D_t and $\hat{F}_{i,t}$ in our ex-ante forecasting model, to predict $\hat{Y}_{t+1}$. We will make predictions for 5 years, using a prediction horizon of $h = 1$ and a recursive window.

5.5 Baseline Forecasting Model

To examine how well our forecasting model from section 5.4 performs, we will compare the forecast predictions with those of a baseline forecasting model. The baseline forecasting model we will use is a Univariate Autoregression model (AR(p)),

$$\hat{Y}_{t+1} = \hat{\alpha} + \hat{\phi}(L)Y_t \tag{13}$$

where our number of lags $p = 1$ is selected, using Schwarze Information Criterion (SIC). The model is estimated using least squares. We will again make predictions for 5 years, using a prediction horizon of $h = 1$ and a recursive window.

The forecast performance is evaluated using the Mean Square Forecast Error (MSFE), which measures the predictive accuracy of the forecast model,

$$MSFE = \sum_{t=R-h+2}^{T-h+1} (Y_{t+1} - \hat{Y}_{t+1})^2. \tag{14}$$

Here R is the total in-sample period, T is the total time period and $h = 1$ is the forecast horizon. Y_{t+1} is the actual value of our country of interest and $\hat{Y}_{t+1}$ is the forecast prediction. The better forecast model is the one which obtains the smallest *MSFE* value.

5.6 Data Generating Processes

To get a better understanding of diffusion indices and to see whether our hybrid model gives accurate predictions with the given number of predictors N and observations T, we will simulate different data generating processes (DGPs). In our DGPs we will first

generate factors from an Autoregressive model with $p = 1$ lag,

$$F_{i,t} = \phi F_{i,t-1} + \eta_t. \tag{15}$$

Where $F_{i,0} = 0$, $\phi = 0.4$ and $\eta_t \sim$ i.i.d. $N(0, \sigma_F^2)$. The index $i = (1, ..., K)$, where K is the total number of factors and the index $t = (1, ..., T)$, where T is the total number of observations.

We also generate our factor loadings $\lambda_{i,j} \sim$ i.i.d. $N(1, \sigma_\lambda^2)$. Next we will use our generated factors, F, and our generated factor loadings $\lambda_{i,j}$, to create the explanatory variables X,

$$\begin{aligned} X_{1,t} &= \lambda_{1,1}F_{1,t} + \cdots + \lambda_{1,r}F_{r,t} + e_{1,t} \\ X_{2,t} &= \lambda_{2,1}F_{1,t} + \cdots + \lambda_{2,r}F_{r,t} + e_{2,t} \\ &\vdots \\ X_{N,t} &= \lambda_{N,1}F_{1,t} + \cdots + \lambda_{N,r}F_{r,t} + e_{N,t} \end{aligned} \tag{16}$$

In the same way as the explanatory variables X, we will create our explained variable Y, only this time using equal factor loadings of one, $\lambda_{Y,1} = \cdots = \lambda_{Y,r} = 1$,

$$Y = \lambda_{Y,1}F_{1,t} + \cdots + \lambda_{Y,r}F_{r,t} + e_{Y,t}. \tag{17}$$

Some noise is added to the explanatory and explained variables using an error term $e_i \sim$ i.i.d. $N(0, \sigma_e^2)$.

After obtaining our variables, X and Y, we can use them to examine how well our forecast model performs. We will execute simulations for two different methods. First we will use a method where we only perform PCA without a machine learning shrinkage method. We use PCA to construct factors for our generated data, X. The number of components we choose, are determined using the selection criterion described in section 5.3. We will use these DGPs to determine how well our selection criterion works and whether this selection criterion is valid for the number of observation T and the number of predictors N of our actual dataset. The number of observations T, the number of predictors N and the number of generated factors K will be varied to analyse how these changes will affect our results. The goal of the simulations is to obtain the success-rate, which is the fraction of the correct amount of components that are chosen with our selection criterion using PCA. A success occurs when our selection criterion selects the same amount of factors as the amount of generated factors K, that were used to create our data X.

$$\begin{aligned} &\textbf{dgp1}: \ N = 50, && K \in [1, 10] && T = 50 \\ &\textbf{dgp2}: \ N = 50, && K = 3, && T \in [10, 150] \\ &\textbf{dgp3}: \ N \in [10, 150], && K = 3, && T = 50 \end{aligned}$$

Next we will analyse how well our hybrid method, which consists of first use Boosting followed by ICA, can re-estimate factors. For these simulations we will use different DGPs to see how the step length ν affects the success-rate. Another important thing is to again see how the number of observations T and number of predictors N will affect this success-rate. This is of big importance because *Kim & Swanson (2014a)*[7], who originally used our hybrid method, had data with much larger dimensions. So there is a possibility that our dataset has too small dimensions for the hybrid method to re-estimate the correct number of factors K. If the hybrid model is not able to re-estimate the correct number of factors K, the factors the model estimates might not be reliable for estimating forecast predictions.

dgp4 : $N = 50,$	$K = 1$	$T = 50$	$\nu \in [0, 1]$	*with boosting*
dgp5 : $N = 50,$	$K \in [1, 3]$	$T = 50$	$\nu = 1$	*with boosting*
dgp6 : $N = 50,$	$K = 2$	$T \in [50, 200, 600]$	$\nu = 1$	*with boosting*
dgp7 : $N \in [50, 100, 200],$	$K = 2$	$T = 50$	$\nu = 1$	*with boosting*

Last but not least we will use a DGP to determine how well our hybrid model actually estimates forecast predictions for Y. We will do this by using the MSFE as described in section 5.5 and compare these with the MSFE values of our baseline model. We will also analyze the distributions of the forecast errors of both of the models, to examine whether they are equal to the distribution of the residuals used to create our data X.

$$\textbf{dgp8} : \; N = 50, \quad K = 1 \quad T = 50 \quad \nu = 0.5 \qquad \textit{with boosting}$$

We will simulate the DGPs without Boosting for $M_1 = 1000$ times and because the simulations for Boosting take more time we will simulate these for $M_2 = 500$.

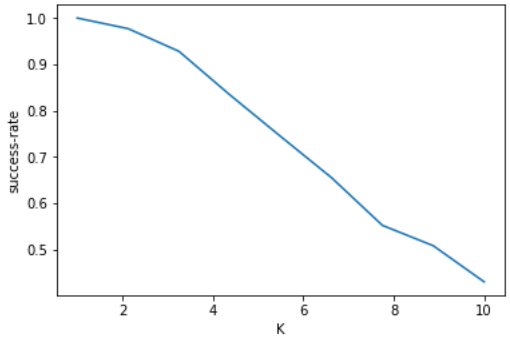
6 Results

In this section we shall present the results obtained from our Data Generating Processes, our hybrid model forecast predictions, our Autoregressive model forecast predictions, the Mean Squared Forecast Errors obtained from these two models and the predictors that have a big impact on our hybrid model.

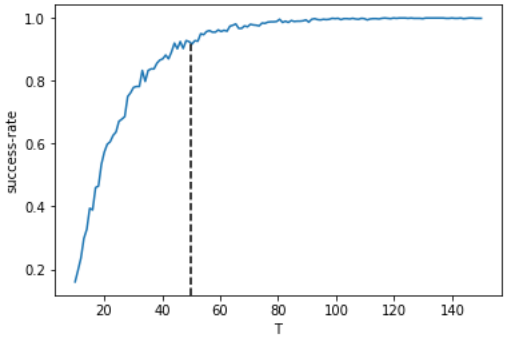
6.1 Simulations

For our first three DGPs from section 5.6 we want to find the accuracy of our selection criterion described in section 5.3. The success-rates of these simulation can be found in

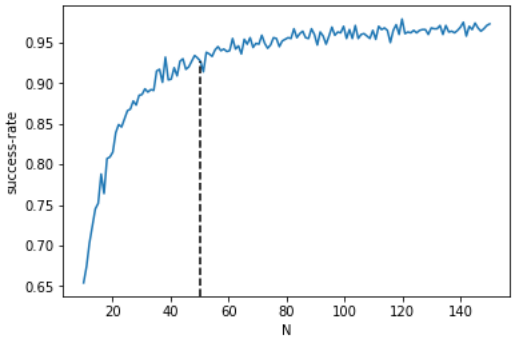
figure 2 below.



(a) **dgp1** with varying number of components K on the x-axis



(b) **dgp2** with varying number of observations T on the x-axis

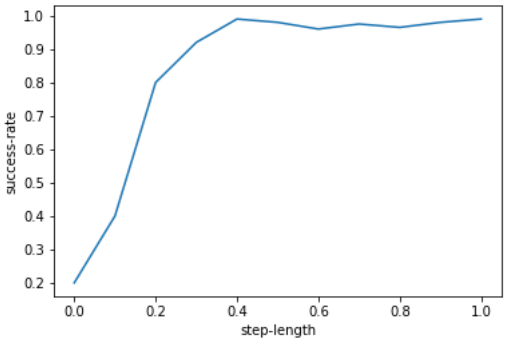


(c) **dgp3** with varying number of predictors N on the x-axis

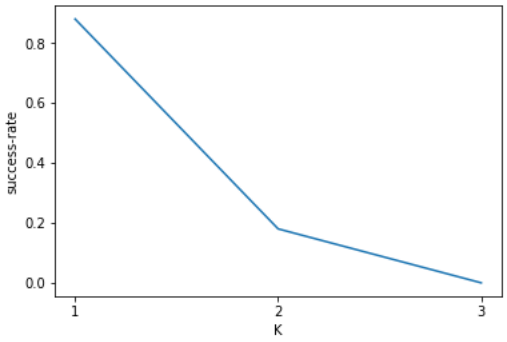
Figure 2: Simulation results of the DGPs with only PCA, with the success-rates on the y-axis and varying variables on the x-axis

In figure 2a, we can see that our success-rate decreases when the number of generated factors K increases. We see that the success-rate for $K = 1$ is approximately equal to 1 and that this success-rate decreases, almost linearly, to 0.4 for $K = 10$. From figure 2b and 2c we can see that the success-rate converges to approximately 1 when the number of observations T and the number of predictors N increase. We see that at $T = 50$ and $N = 50$ the success-rate is approximately 0.9.

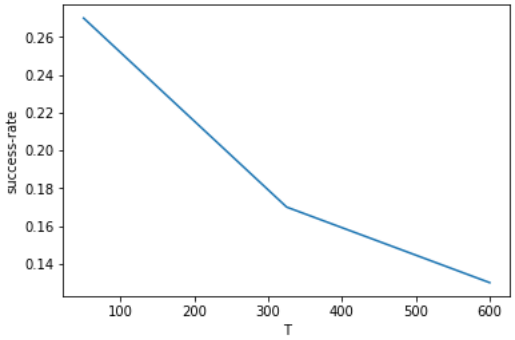
For our next DGPs, we use Boosting followed by PCA. The success-rates of these DGPs can be found in figure 3 below.



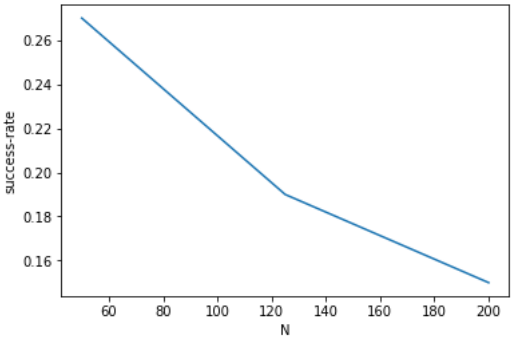
(a) **dgp4** with varying step-length ν on the x-axis



(b) **dgp5** with varying number of factors K on the x-axis



(c) **dgp6** with varying number of observations T



(d) **dgp7** with varying number of predictors N on the x-axis

Figure 3: Simulation results of the DGPs with Boosting followed by PCA with the success-rates on the y-axis and varying variables on the x-axis

From figure 3a we can see that the success-rate converges to approximately 1 when the step-length ν from the Boosting algorithm increases. From figure 3b we can see that the success-rate decreases when the number of generated components K increases. The success-rate for $K = 1$ is approximately 0.9 but for $K = 2$ this success-rate decreased significantly to about 0.22. From figures 3c and 3d we can see that the success-rate decreases when the number of observations T and the number of predictors N increases. So overall we see that the success-rates are large when we use a step-length, ν, larger than 0.3, and we see that the success-rate is only large for a K=1. We also see that the success-rate decreases for an increasing N and T.

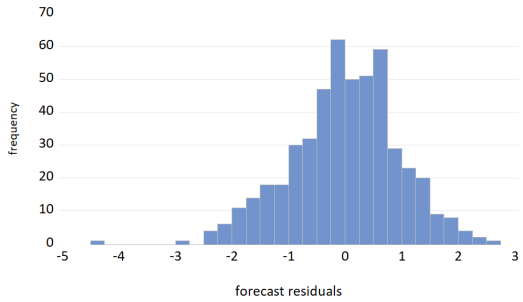
Now that we know how well our hybrid model can re-estimate factors, we want to know how well it can actually estimate forecasts. To test the forecast accuracy of the hybrid model we compare the forecast predictions with that of an AR(1) model using the MSFE. The one-step-ahead MSFE values for a period of 5 years, predicted with a forecast horizon of h=1 and a recursive window, can be found in the table 2 below.

Table 2: MSFE values of forecasts with h=1 and a recursive window of **dgp8**

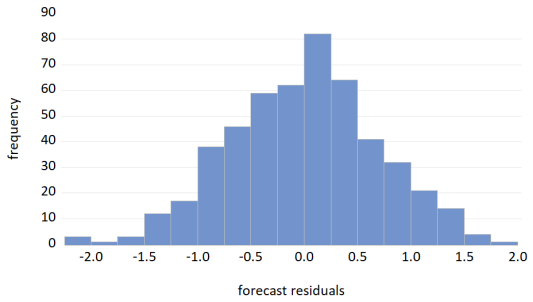
	hybrid model	AR(1) model
MSFE	5.380	5.335

From the table we can see that the the AR(1) model obtained the smallest MSFE value, but the difference between the two MSFE values is very small.

Finally we want to examine the distribution of the forecast errors of the hybrid and AR(1) model. These distributions can be found in table 4 below.



(a) Distribution of the forecast residuals of the hybrid model



(b) Distribution of the forecast prediction of the AR(1) model

Figure 4: Forecast prediction residual results of **dgp8**

From these figures we can see that both of the distributions look approximately normal. The hybrid model has a kurtosis and skewness of respectively 3.527 and -0.314 and those of the AR(1) model are respectively 2.910 and -0.062.

6.2 Forecast Predictions

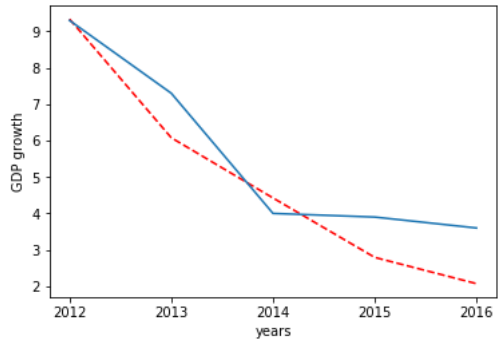
After obtaining the simulation results, we use our hybrid model on the African GDP growth dataset. First we estimate forecast predictions using our hybrid model described in section 5.4. We do this using different values for *pmax* and use these different values to create different sets of potential predictors as described in section 1. The MSFE values using these different values for *pmax* can be found in table 3 below.

Table 3: MSFE of forecast predictions using different values for *pmax*

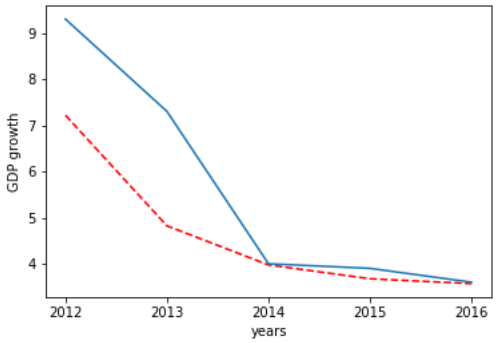
pmax	2	3	4	5
MSFE	25.218	14.208	5.239	7.636

In table 3 we can see that the smallest MSFE value is obtained at *pmax*=4. Next we make forecast predictions for the period 2012-2016 , using *pmax*=4, a forecast horizon

of h=1 and a recursive window, using the models described in section 5.4 and 5.5. The forecast predictions can be found in figure 5.



(a) The solid line gives the actual GDP-growth values and the dashed line gives the forecast predictions estimated with the hybrid model



(b) The solid line gives the actual GDP-growth values and the dashed line gives the forecast predictions estimated with the AR(1) model.

Figure 5: Forecast predictions for the hybrid-model and the AR(1) model.

The MSFE values for these one-step-ahead forecasts of the period 2012-2016 can be found in table 4.

Table 4: MSFE values for one-step-ahead forecasts of the period 2012-2016 with h=1 and a recursive window

	hybrid model	AR(1) model
MSFE	5.239	10.535

In this table we can see that the hybrid model obtains the smallest MSFE value.

Next we calculate the MSFE for each year's individual forecast prediction, these values can be found in table 5.

Table 5: MSFE values of each year's individual forecast with h=1 and a recursive window

	2012	2013	2014	2015	2016
hybrid model	0.001	1.493	0.180	1.227	2.337
AR(1) model	4.337	6.145	0.001	0.051	0.001

In this table we can see that the MSFE values for each years individual forecast prediction, fluctuate for the hybrid model and seem to increase in the period 2014-2016. While for the AR(1) model these values first start of high and seemingly decrease in the period 2014-2016.

6.3 Factors Explained

Now that we know how well our hybrid model performs compared to the baseline model, we are interested in knowing which predictors, thus which countries and their lags, have a big influence on Ghana's future GDP growth. The ICA factors are created using formula 4 in section 5.2. Where Ψ can give us the impact each predictor has on the factors. The scree plots giving the 10 greatest impacts on the factor are given in figure 6 below.

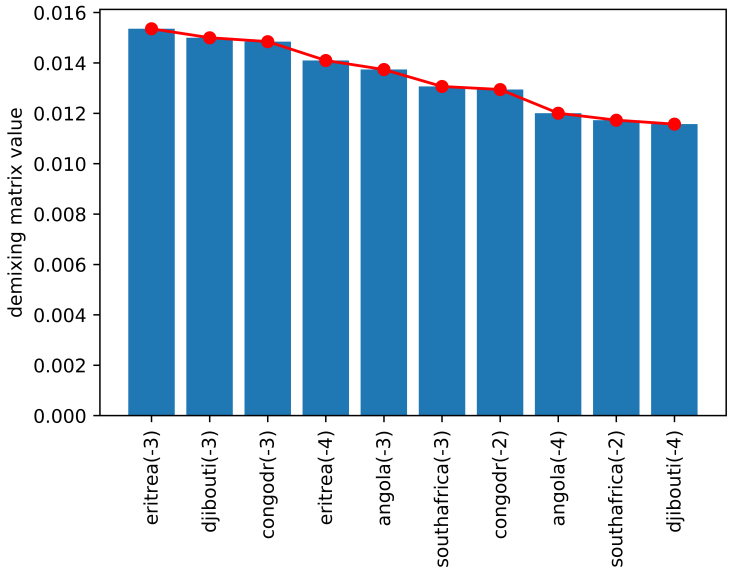


Figure 6: Scree plot giving the 10 predictors with the greatest impact on the factor used to forecast Ghana's 2012 GDP growth

In figure 6 we can see the 10 countries that have the biggest impact in creating the factor used to estimate a forecast for Ghana's 2012 GDP growth.

The scree plots of the impact all the predictors have on the factor for Ghana's 2012 GDP growth and the scree plots of the impacts on the factors used for the forecast predictions for the period 2013-2016 can be found in the appendix.

From all the scree plots of the period 2012-2016 we can see that the predictors that overall have the greatest impact on the GDP growth forecasts of Ghana, are different lags of Eritrea, Djibouti, the Democratic Republic of Congo, Angola and South Africa.

7 Conclusion

To give an answer to our research question, we used a hybrid model, by applying Boosting followed by ICA, to forecast Ghana's GDP growth using lagged GDP growths from other African countries.

First we executed simulations to acquire the accuracy of the methods used to forecast Ghana's GDP growth. From the results of the first three DGPs from section 6.1,

performed with only PCA, we can conclude that our factor selection criterion works well for estimating a small number of factors. For the size of our dataset, which consists of approximately 50 predictors and 50 observations, the selection criterion gives good results, with about a 90% success-rate. Next, from the results of the DGPs with Boosting followed by PCA, we can conclude that the accuracy of the success-rate increases with a larger step-length ν, this can be explained by the fact that a larger step-length ν, results in the selection of more predictors in the Boosting algorithm, which means that there is more information to re-estimate the correct amount of factors. Furthermore we conclude that the method only works well when estimating one factor and we notice that the success-rate decreases when we generate data with a larger number of observations T and predictors N. The reduction of the success-rate with a larger number of factors K, observations T and predictors N, can be explained by the impreciseness of the Boosting algorithm. Boosting is not very interpretable in the way it selects predictors. The algorithm roughly decides which predictors to filter out, resulting in those predictors not being taken into consideration for PCA. The ignored predictors could contain a great amount of valuable information and the ignorance of these predictors causes the success-rate to decrease. So when using larger numbers of factors, observations and predictors the success-rate decreases even more, because the greater amount of data available, the more valuable predictors Boosting will filter out. Although the success-rate, which measures the successful re-estimation of factors, decreases when using the method with Boosting, does not imply that Boosting does not give added value when estimating forecasts. From the results of the final DGP we can conclude that the AR(1) model only makes slightly better predictions than the hybrid model, but since the difference between the MSFE values is so small, there is no clear "better" model for our simulated data. We can also conclude that the forecast errors of both the hybrid and AR(1) model are normally distributed, which is good since the data is also created with normally distributed error terms. So as a final conclusion of the DGPs we can say that our estimated factors from the methods using only PCA are more reliable than those estimated with Boosting followed by PCA. Although this is the case, the Boosting algorithm does make quite accurate forecast predictions and thus give an added value when estimating forecasts.

Next we have estimated forecast predictions for Ghana's GDP growth with the hybrid model and the AR(1) model, these results can be found in section 6.2. We can conclude that the hybrid model overall makes better predictions for the period 2012-2016. However, looking at the MSFE values for each year individually, we can conclude that there are years that the AR(1) model makes better predictions than the hybrid model. All in all, we can say that the hybrid model outperforms the AR(1) model and that using lags of other African country's GDP growth, does give added value in predicting Ghana's GDP growth.

Furthermore we have looked at the results in section 6.3 to examine the predictors with

the highest impact on the factor used to estimate our GDP growth forecast. The predictors with the most influence are different lags of, Eritrea, Djibouti, the Democratic Republic of Congo, Angola and South Africa. Like we stated before, the Boosting algorithm does not select predictors with an underlying meaning. This causes for the selection of a few predictors that might be unexpected. Some of the predictors do have some clear connections to Ghana. First of all, Angola's main resource is petroleum oil which is one of Ghana's top export products. Secondly, South-Africa contains one of Africa's richest gold mines just like Ghana and lastly, cacao beans are Ghana's third top export product which also happens to be one of the Democratic Republic of Congo's most important products. These similarities could explain why lags of these country's GDP growth have a big impact on Ghana's GDP growth.

To summarize everything we can thus conclude that the hybrid model we have created, by filtering a complete set op potential predictors with Boosting followed by performing ICA on these filtered predictors, give accurate forecast predictions for Ghana's GDP growth for the period 2012-2016. We also conclude that adding a factor containing lags of other African country's GDP growth to a regular AR(1) model, will give better forecast predictions than a traditional AR(1) model. This hybrid model cannot give the underlying meaning of the selected predictors, because of the way the filtering of the Boosting algorithm works. All things considered we conclude that our hybrid model is a reliable estimator for future GDP growths for Ghana and thus a helpful tool to further analyze Ghana's growing economy.

References

[1] Bai J. & Ng S. (2002). Determining the number of factors in approximate factor models. *Econometrica* 70 (1) 191-221.

[2] Bai J. & Ng S. (2006). Confidence Intervals for Diffusion Index Forecasts and Inference for Factor-Augmented Regressions. *Econometrica* 74 (4) 1133-1150

[3] Bai J. & Ng S. (2009). Boosting diffusion indices. *Journal of Applied Econometrics* 24 (4) 607-629.

[4] Franses P. H. & Vasilev S. (2019). Real GDP growth in Africa, 1963-2016. *Econometric Institute, Erasmus School of Economics*

[5] Hyvärinen A. & Oja E.(2000). Independent component analysis: algorithms and applications. *Neural Networks* 13 (4-5) 411-430

[6] Kim H.H. & Swanson N.R. (2013). Mining big data using parsimonious factor and shrinkage methods. *SSRN Electronic Journal.*

[7] Kim H.H. & Swanson N.R. (2014a). Forecasting financial and macroeconomic variables using data reduction methods: New empirical evidence. *Journal of Econometrics* 1778 (2) 352-547

[8] 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 (2018) 339–354.

[9] Stock J. H. & Watson M. W. (2002a). Forecasting Using Principal Components From a Large Number of Predictors. *Journal of the American Statistical Association* 97 1167-1179

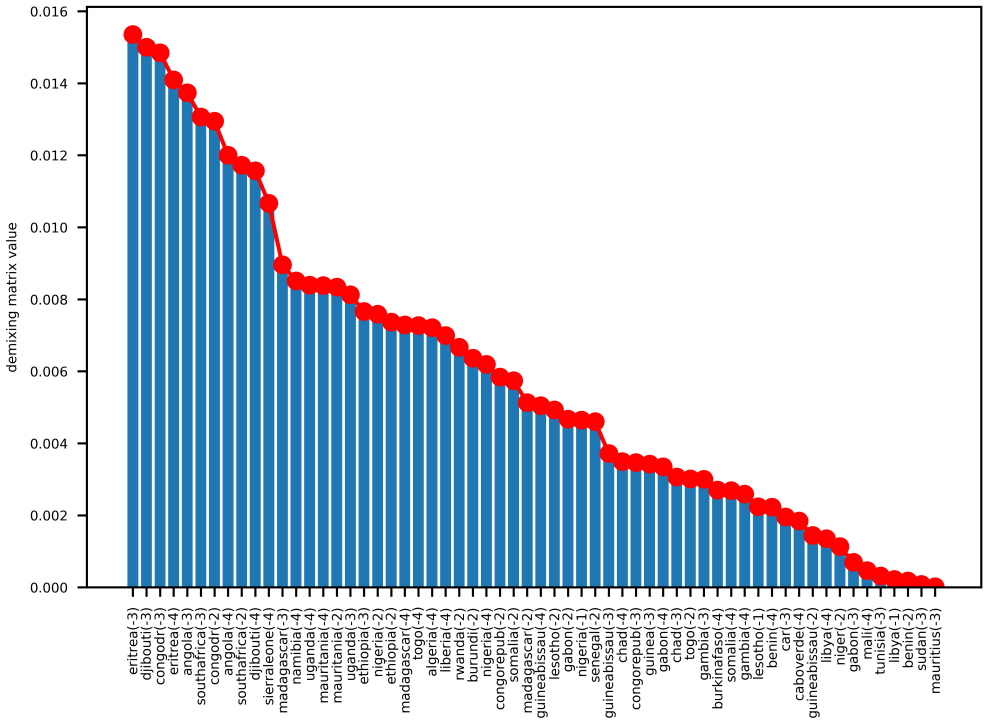
[10] Stock J. H. & Watson M. W. (2002b). Macroeconomic Forecasting Using Diffusion Indexes. *Journal of Business and Economic Statistics* 20 (2) 147-162

[11] Roser M. (2019), Economic Growth. Published online at OurWorldInData.org. *https://ourworldindata.org/economic-growth*

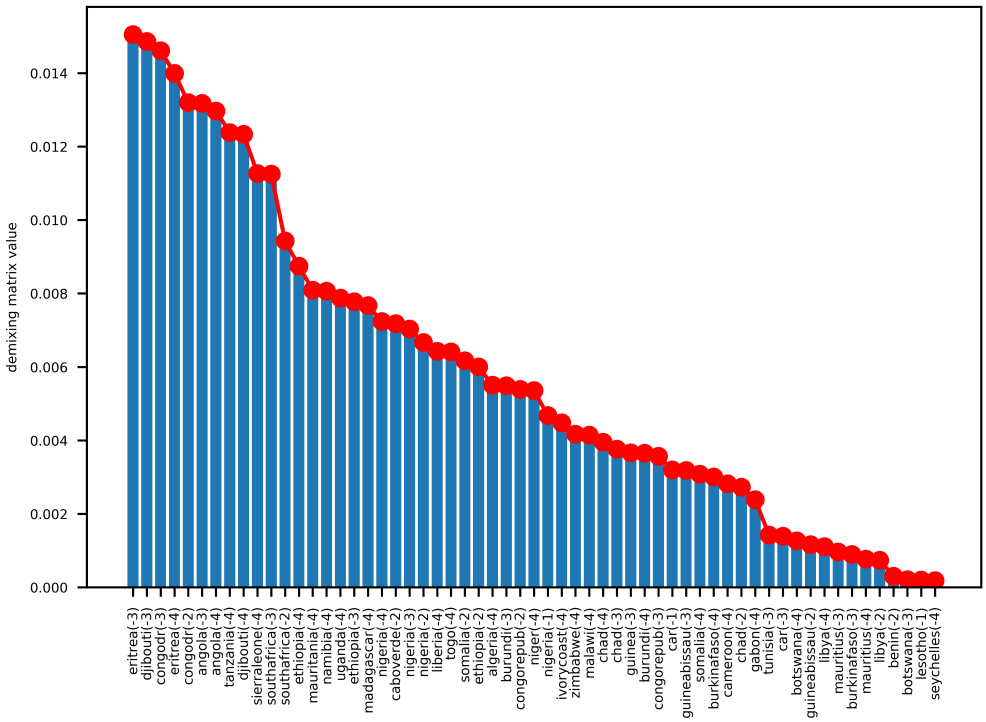
[12] Chappelow J. (2019), Gross Domestic Product (GDP). Published online at Investopedia.com. *https://www.investopedia.com/terms/g/gdp.asp*

[13] Geiger M., Tanaka T. & Nuamah C. (2018). Ghana's growth history: New growth momentum since 1990s helped put Ghana at the front of poverty reduction in Africa. Published online at blogs.worldbank.org/africacan. *https://blogs.worldbank.org/africacan/ghanas-growth-history-new-growth-momentum-since-the-1990s-helped-put-ghana-at-the-front-of-poverty*

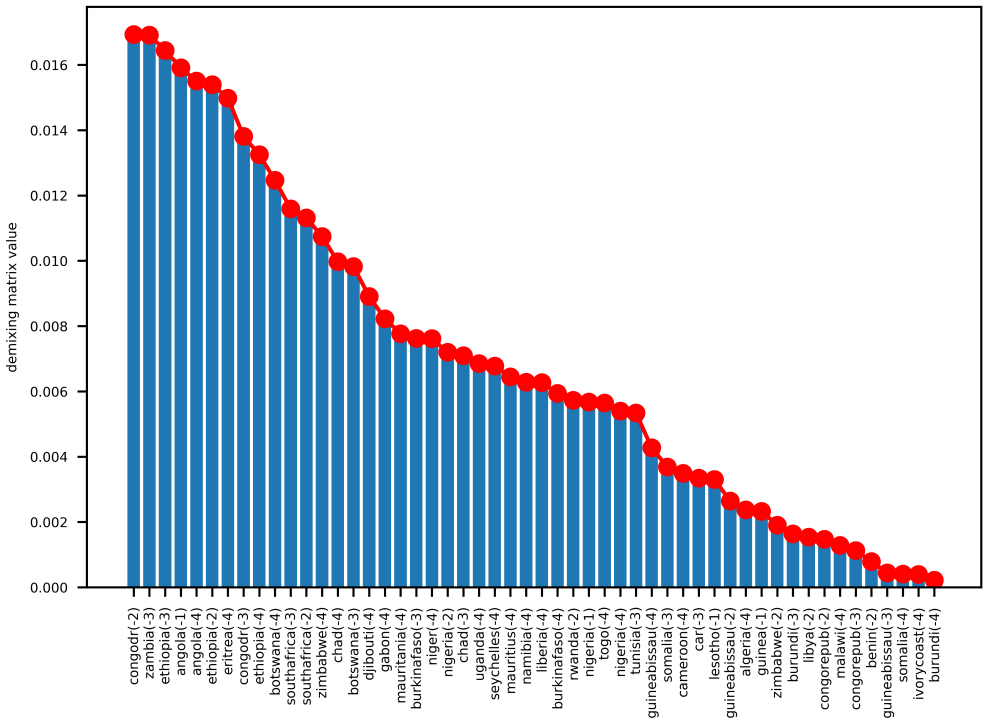
8 Appendix



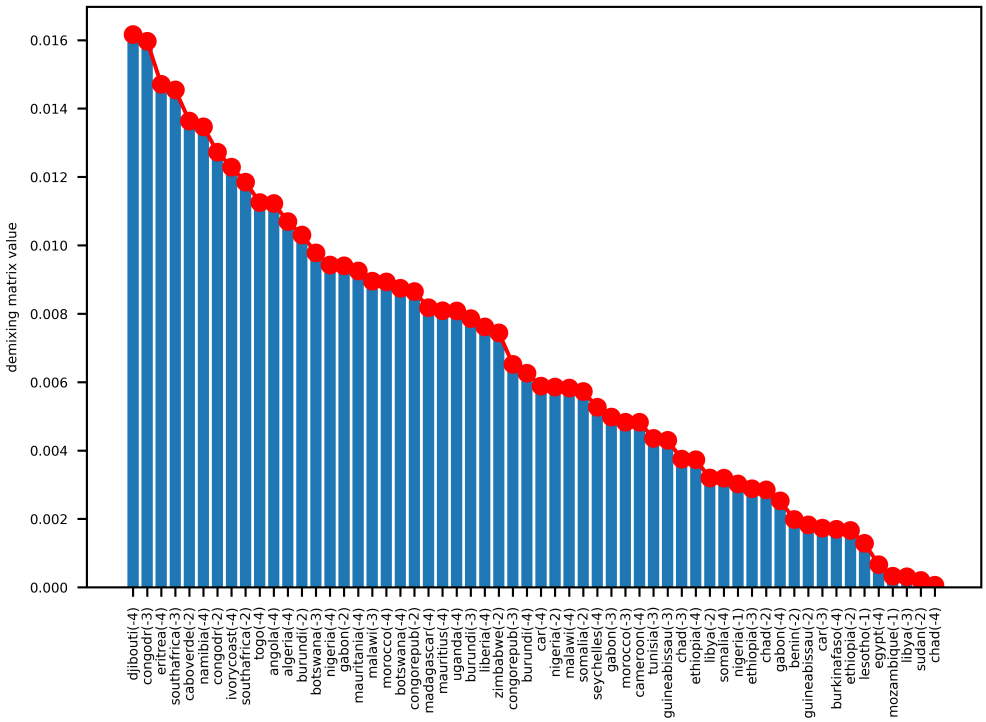
(a) Scree plot of the impacts for 2012s prediction



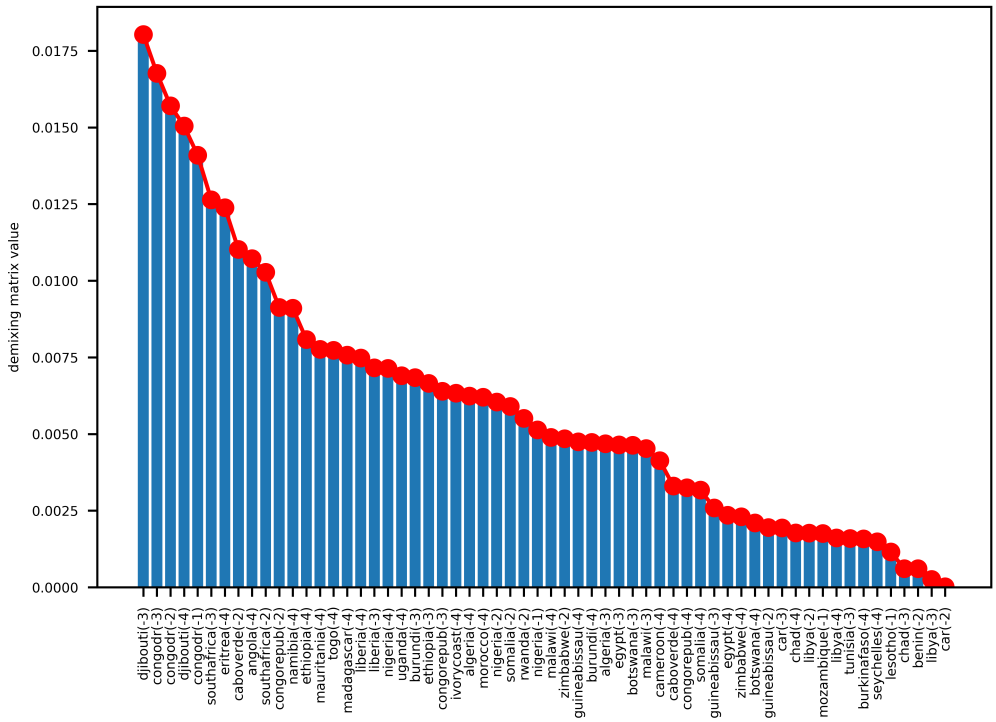
(b) Scree plot of the impacts for 2013s prediction



(c) Scree plot of the impacts for 2014s prediction

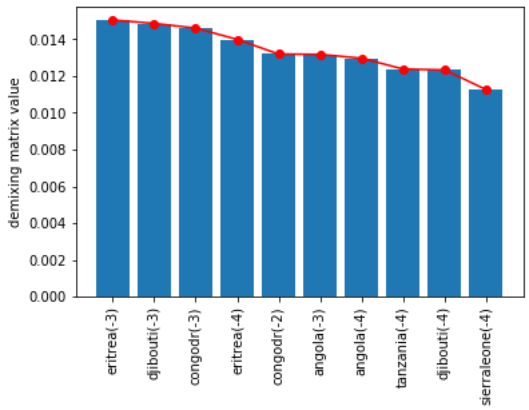


(d) Scree plot of the impacts for 2015s prediction

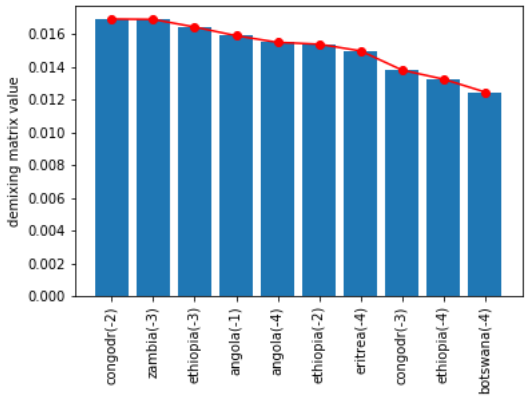


(e) Scree plot of the impacts for 2016s prediction

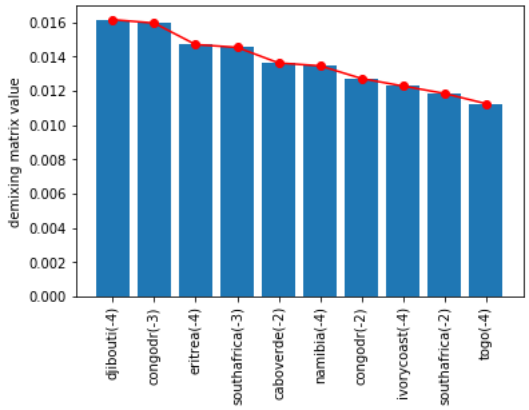
Figure 7: Scree plots giving the impacts on the factors for the period 2012-2016



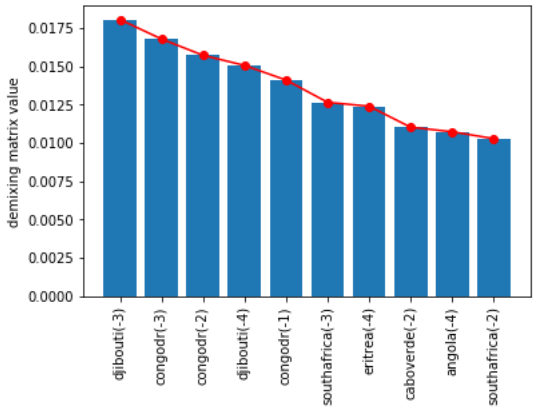
(a) Scree plot of the 10 largest impacts for 2013s prediction



(b) Scree plot of the 10 largest impacts for 2014s prediction



(c) Scree plot of the 10 largest impacts for 2015s prediction



(d) Scree plot of the 10 largest impacts for 2016s prediction

Figure 8: Scree plots giving the 10 largest impacts on the factors for the period 2013-2016

Listing 1: Boosting algorithm code

```
"""
Created on Thu May 16 20:47:38 2019
@author: Solange Eersel
Code for selecting predictors using Component Wise L2 Boosting
"""
import numpy as np
import pandas as pd
import statsmodels.api as sm
import math

file ='file:///C:/Users/Gebruiker/Documents/thesis/africa_gdp.xlsx'
xl = pd.read_excel(file, index_col=0,header=0,skiprows=range(1, 3))
df=pd.DataFrame(xl)
xl=xl.iloc[0:54,:]                    #time series for the years 1963-2016
ghana=xl.loc[:,df.columns=='ghana'] #time series for Ghana
X=xl.loc[:, df.columns != 'ghana']  #time series for all countries except Ghana

pmax=4      #maximum number of lags
v=0.5       #steplength
M=250       #Boosting iteration
T=46        #total time
N=53        #number of countries
At=math.log(T)
#Create vector of potential predictors
p=1
```

```
T_lag=T-1
z=X.iloc[0:T]
z.columns += '('+str(-1*pmax)+')'
while p<=pmax-1 :
    X_lag=X.shift(-1*p).iloc[0:T,:]
    X_lag.columns += '('+str(-1*(pmax-p))+')'
    z=pd.concat([z,X_lag], axis=1)
    p=p+1

mu=pd.DataFrame(np.zeros((T_lag, 1)))
B=pd.DataFrame(np.zeros((T_lag,T_lag)))

#Create mu(0), initial mu vector for first iteration
for t in range(0,T_lag) :
    mu.iloc[t,0]=y.mean().values
#Create B(0), initial B vector for first iteration
for t in range(0,T_lag):
    for i in range(0,T_lag):
        B.iloc[t,i]=1/(T_lag)

#Create all vectors needed in the boosting algorithm
r=z.shape[1]
u = pd.DataFrame(np.zeros((T_lag, 1)))
betas_boost= pd.DataFrame(np.zeros((r,1)))
ssr_boost=pd.DataFrame(np.zeros((r,1)))
InfoC=pd.DataFrame(np.zeros((M,1)))
Final_pred=pd.DataFrame()

At=math.log(T_lag)
#Begin Boosting algorithm
for i in range(0,M) :
    u=pd.DataFrame(y.iloc[1:T].values-mu.values)
    for r in range(0,r) :
        zj=pd.DataFrame(z.iloc[0:T_lag,r])
        reg = sm.OLS(u, zj).fit()
        resid=reg.resid
        ssr_boost.iloc[r,0]=np.dot(resid.transpose(), resid)
        betas_boost.iloc[r,0]=reg.params.values
    j=ssr_boost.idxmin()                                    #Find the argmin for the SSR vector
    g=z.iloc[0:T_lag,j].multiply(betas_boost.iloc[j,0].values)
    mu=mu.add(g.multiply(v).values)                         #Update the weak learner

    Final_pred=pd.concat([Final_pred,z.iloc[0:T,j]], axis=1) #take untill 2011

    #Calculate the Information Criterion to obtain optimum number of iterations
    sigma=np.sum(np.square(y.iloc[1:T].values-mu.values))    #Calculate the sigma squared matrix
    hat1=pd.DataFrame(np.linalg.pinv(z.iloc[0:T_lag,j].transpose().dot(z.iloc[0:T_lag,j])))
    hat2=pd.DataFrame(z.iloc[0:T_lag,j].values).dot(hat1)
    hat=pd.DataFrame(hat2.dot(z.iloc[0:T_lag,j].transpose().values))    #Calculate the P matrix
    imat=np.identity(T_lag)
    B=B.add((hat.multiply(v)).dot(imat-B))                  #Update the B matrix
    df=np.trace(B)                                          #degrees of freedom
    InfoC.iloc[i,0]=math.log(sigma)+(At*df)/(T_lag)         #Information Criterion

#Find smallest IC and select th predictors for that number of iteration
Mopt=InfoC.idxmin()+1 #optimal number of iterations
sel_pred=Final_pred.iloc[:,0:int(Mopt)]

#Remove predictors that are selected more than once
wl = sel_pred.loc[:,~sel_pred.columns.duplicated()]
```

Listing 2: Selection criterion and ICA code

```
"""
Created on Thu May 23 14:26:57 2019
@author: Solange Eersel
Code for using PCA on the subset of variables chosen by component wise L2 Boosting. Choosing the
    number of factors using the SIC formula from Bai and Ng. Calculate ICA components using the
```

```
    FastICA algorithm
"""
import numpy as np
import pandas as pd
import statsmodels.api as sm
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.decomposition import FastICA

file ='predictors_0.5.xlsx'
wl = pd.read_excel(file,header=0)
T=46
wl=pd.DataFrame(wl.iloc[0:T,:].values)

N=wl.shape[1]
ssr_tot=0
loadings=pd.DataFrame(np.zeros((N,N)))
sic=pd.DataFrame(np.zeros((N,1)))

#PCA on subset of variables
scaler=StandardScaler()
pca=PCA()
W=scaler.fit_transform(wl)
F_pca=pd.DataFrame(pca.fit_transform(W))

#Calculating the loadings
reg=sm.OLS(wl, F_pca).fit()
loadings=reg.params
resids=reg.resid

#compute selction criterion
for i in range(0,N):
    ssr_tot=np.dot(resids.iloc[:,i].transpose(),resids.iloc[:,i])+ssr_tot
sigmasq=ssr_tot/(N*T)  #calculate the sigma squared

for r in range(1,N+1):
    ssr_tot1=0
    reg1=sm.OLS(wl, F_pca.iloc[:,0:r]).fit()
    loadings1=reg1.params
    resids1=reg1.resid
    for i in range(0,N):
        ssr_tot1=np.dot(resids1.iloc[:,i].transpose(),resids1.iloc[:,i])+ssr_tot1
    V=ssr_tot1/((N)*T)
    h=((N+T-(r))*np.log(N*T))/(N*T)
    sic.iloc[r-1,0]=V+(r)*sigmasq*h

j=sic.idxmin()+1          #find minimum value
reg_final=sm.OLS(wl, F_pca.iloc[:,0:int(j)]).fit()
loadings_final=reg_final.params

#estimate factors using ICA
ica=FastICA(n_components=int(j))
F_ica=ica.fit_transform(W) #use FastICA algorithm to compute the factors
```

Listing 3: Hybrid forecast model with recursive window and h=1 code

```
"""
Created on Tue Jun 11 14:49:41 2019
@author: Solange Eersel
Hybrid forecast model code using a recursive window and h=1
"""
import numpy as np
import pandas as pd
import statsmodels.api as sm
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.decomposition import FastICA
import math
```

```
file ='file:///C:/Users/Gebruiker/Documents/thesis/africa_gdp.xlsx'
xl = pd.read_excel(file, index_col=0,header=0,skiprows=range(1, 3))
df=pd.DataFrame(xl)
xl=xl.iloc[0:54,:]                    #time series for the years 1963-2016
ghana=xl.loc[:,df.columns=='ghana'] #time series for Ghana
X=xl.loc[:, df.columns != 'ghana'].reset_index(drop=True) #time series for all countries except
    Ghana

y_estimates=pd.DataFrame(np.zeros((54, 1)))
y_estimates.iloc[0:49]=ghana.iloc[0:49].reset_index(drop=True)

pmax=4      #maximum number of lags
v=0.5       #steplength
M=250       #Boosting iteration
T=46        #total time
N=53        #number of countries
y=ghana.shift(-4).iloc[0:T,:].reset_index(drop=True) #1966-2011

for T in range(46,51) :

    #BOOSTING CODE

    #SELECTION CRITERION AND ICA CODE

    #fit model for t using data untill t-1
    regress = sm.OLS(y.iloc[1:T].reset_index(drop=True), expl_lag).fit()
    betas=regress.params          #estimate forecast betas

    #predict forecast for t
    W=scaler.fit_transform(wl)
    F_ica=pd.DataFrame(ica.fit_transform(W))
    expl=pd.concat([y, F_ica], axis=1)
    y_predict=expl.dot(betas)
    y_estimates.iloc[T+3]=y_predict.iloc[T-1]
    y=pd.DataFrame(np.zeros((T+1, 1)))
    y=pd.DataFrame(y_estimates.iloc[3:T+4,0]).reset_index(drop=True)

y_hybrid=y.iloc[46:51]
```

Listing 4: AR(1) forecast model with recursive window and h=1 code

```
"""
Created on Thu Jun 20 14:33:32 2019
@author: Solange Eersel
Code to calculate forecasts for the AR(1) model
"""
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima_model import ARMA

file ='file:///C:/Users/Gebruiker/Documents/thesis/africa_gdp.xlsx'
xl = pd.read_excel(file, index_col=0,header=0,skiprows=range(1, 3))
df=pd.DataFrame(xl)
xl=xl.iloc[0:54,:]                    #time series for the years 1963-2016
ghana=xl.loc[:,df.columns=='ghana'] #time series for Ghana
y=ghana.iloc[0:49].reset_index(drop=True)
y_estimates=pd.DataFrame(np.zeros((54, 1)))
y_estimates.iloc[0:49]=ghana.iloc[0:49].reset_index(drop=True)
test=y

for T in range(49,54) :
    mod = ARMA(y, order=(1,0))
    res = mod.fit()
    value=res.predict(start=T, end=T)
    y_estimates.iloc[T]=(pd.DataFrame(value).values)
    y=pd.DataFrame(np.zeros((T+1, 1)))
    y=pd.DataFrame(y_estimates.iloc[0:T+1,0]).reset_index(drop=True)
```

Listing 5: MSFE code

```
"""
Created on Wed Jun 12 20:51:02 2019
@author: Solange Eersel
Code for calculating MSFE values for Pmax=2,3,4,5
"""
import numpy as np
import pandas as pd
file ='file:///C:/Users/Gebruiker/Documents/thesis/forecasts.xlsx'
xl = pd.read_excel(file)
df=pd.DataFrame(xl)
y_real=df.iloc[:,0]
y_ar=df.iloc[:,5]
y_hybrid=df.iloc[:,9]
msfe_ar=0
msfe_hybid=0
for t in range(0,5):
    msfe_ar=(y_real.iloc[t]-y_ar.iloc[t])**2+msfe_ar
    msfe_hybid=(y_real.iloc[t]-y_hybrid.iloc[t])**2+msfe_hybid
```

Listing 6: DGP code

```
"""
Created on Thu May 30 14:48:51 2019
@author: Solange Eersel
DGP code
"""
import numpy as np
import pandas as pd
import statsmodels.api as sm
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from statsmodels.tsa.arima_process import ArmaProcess

T, K, N, sigma1, sigma2=54, 1, 50, 0.5, 1
T2, v, pmax, M=50, 1, 4, 250
sim=1000
np.random.seed(12345)
begin, end=1, 11
hit_rates=np.empty((end,1))

for K in range(begin,end):
    succes=0
    for s in range(0,sim):
        fdgp=pd.DataFrame(np.ones((T, K)))
        #generate ar process
        for i in range(0,K):
            ar1 = np.array([1, -0.4])
            AR_object1 = ArmaProcess(ar1)
            fdgp.iloc[:, i] = AR_object1.generate_sample(nsample=T)

        #Create X giving with the factors, random loadings and an error
        X=pd.DataFrame(np.ones((T,N)))
        for i in range(0,N):
            rand_loadings=np.random.normal(1,sigma1,K)
            X.iloc[:, i]=fdgp.dot(rand_loadings)+np.random.normal(0,sigma2)

        Y=pd.DataFrame(fdgp.sum(axis=1)).iloc[4:T,:].reset_index(drop=True) #1966-2011

        #BOOSTING ALGORITHM CODE

        #SELECTION CRITERION CODE

        if j[0]==K: #if the amount of re-estimated factors is equal to original number of factors
            succes=succes+1
    succes_rate=succes/sim
    hit_rates[itering,0]=succes_rate
hit_rates=hit_rates[begin:end,:]
```