A Neural Network Ensemble Approach for GDP Forecasting *

'Declarations of interest: none'.

July 2021

Abstract

We propose an ensemble learning methodology to forecast the future US GDP growth release. Our approach combines a Recurrent Neural Network (RNN) with a Dynamic Factor model accounting for time-variation in the mean with a Generalized Autoregressive Score (DFM-GAS). We show how this combination improves forecasts in the aftermath of the 2008-09 global financial crisis, as well as in the latest COVID-19 recession, by reducing the root mean squared error for the short-term forecast horizon. Thus, we evaluate the predictive ability of each component of the ensemble by considering variations in the mean, as the latter are potentially caused by recessions affecting the economy. For our scope, we employ a set of predictors encompassing a wide range of variables measured at different time frequencies. Thus, we provide dynamic coefficients for predictors after an interpretable machine learning routine to assess how the model reflects the evolution of the business cycle.

Keywords: macroeconomic forecasting; machine learning; neural networks; dynamic factor model; Covid-19 crisis; Mixed frequency.

JEL codes: C53, E37, 051

*We are grateful to participants at the conference on "Machine Learning of Dynamic Processes and Time Series Analysis" held at Scuola Normale Superiore in Pisa, and to participants at the 41st International Symposium of Forecasting. We want to thank Marco Bee, Fulvio Corsi, and Luca Trapin for their valuable comments.
1 Introduction

Forecasting the future state of an economy has considerably improved since the financial crisis in 2008-2009, thanks to the availability of different and heterogeneous data sources with mixed frequencies and methodological advances [Bańbura et al., 2013, Buono et al., 2017]. The seminal work by Giannone et al. [2008b] introduced a dynamic factor model (DFM) to nowcast the current and future GDP quarters based on a wide set of monthly indicators. A few years later, Andreou et al. [2013] used daily financial data to forecast macroeconomic variables with a Mixed-Data Sampling (MIDAS) regression. More recently, machine learning methods have been used to scale up opportunities of modelling and predicting economic indicators [Athey and Imbens, 2017, Athey, 2018]. For example, elastic net regressions and random forest methods were used to nowcast the Lebanese GDP in Tiffin [2016]. More in general, Coulombe et al. [2020] introduce the concept of machine learning for improvements in macroeconomic predictions. Most recent applications include the nowcast of the US GDP by using a sparse-group LASSO, as in Babii et al. [2020]. Yet, among different machine learning approaches, neural networks are those that most capture the attention of scholars thanks to their natural application to time series. See for instance Kaastra and Boyd [1996] for a neural network design to forecast financial time series. Richardson et al. [2020] performed a horse-race between autoregressive (AR), DFM, and machine learning methods including neural networks to show that the latter performed better in nowcasting the GDP of New Zealand.

Interestingly, statistical learning is more useful for the prediction of macroeconomic indicators whenever complexity and structural breaks occur. Indeed, complexity may arise from a change in the data generating process that happens every time a structural break affects the time series, because what is observed in-sample has limited information on what happens in the out-of-sample window. This is typically the case in periods of extraordinary economic recessions, such as the 2008-2009 recession and the latest COVID-19 crisis. For this reason, most recent works have sought to improve on prediction accuracy in times of recession by looking for alternative techniques. Among others, Foroni et al. [2020] use MIDAS regression and attempt to adjust original nowcasts and forecasts during the COVID-19 crisis by an amount similar to the nowcast and forecast errors that can be retrieved from the latest financial crisis in 2008-2009, based on the assumption that these crisis are comparable. However, others claim the necessity of more sophisticated specifications. For example, a Markov-switching DFM is used by Carstensen et al. [2020] to predict recession periods through the German business cycle. Antolin-Diaz et al. [2021] use a Bayesian DFM to model time-varying parameters, as well as including newly available high-frequency data in a nowcasting exercise after the COVID-19 crisis outbreak. Within the context of Bayesian analysis, a recent paper of Cimadomo et al. [2021] studies how a large number of time series can be handled in a Bayesian vector autoregressive
model (BVAR) to improve both monitoring and nowcasting of the economic activity. Goulet Coulombe et al. [2021] exploit the ability of machine learning models to detect nonlinear patterns in the macroeconomic variables during the COVID-19 crisis.

Against the previous background, we propose combining a DFM model with a machine learning approach to build on the advantages they both provide in different moments of the business cycle. As a matter of fact, our ensemble package reveals that neural networks do improve the performance of a time-varying DFM in the prediction of the economic activity when the process presents dynamics in the first moment. We show that the DFM-GAS always outperforms its fixed-parameter counterpart. We also find that the neural network ensemble improves the forecast performance in the window we consider, especially for short-term forecast horizons.

As for data modelling, we argue that the variation in the mean can be partially explained by a mean shift that causes structural breaks in the data generating process. This is the reason why we use an out-of-sample window where we consider the crisis in 2008-2009 while implementing a Chow test to evaluate to what extent our model is able to predict in times of structural breaks. Therefore, we compute one-quarter to four-quarter ahead forecasts to evaluate differences between the time-varying DFM and neural networks over different time horizons, and we assess when it is suitable to put them together in an ensemble model to forecast along the good and bad turns of the business cycle. In fact, dynamic factor models are widely used within the context of macroeconomic nowcasting and forecasting, and many specifications include dynamics in the parameters as well as the possibility to model breaks along the economic cycle [Del Negro and Otrok, 2008, Camacho et al., 2012, Lee, 2012, Korobilis, 2013, Barigozzi et al., 2020]. We adopt a score-driven approach with a GAS, similarly to Creal et al. [2013], as a way of capturing parameter dynamics in the DFM specification. In particular, we implement an augmented specification of Giannone et al. [2008b] where the first moment of the estimated process is considered as a time-varying parameter modelled with a generalized autoregressive score (GAS) process.

Eventually, we show the advantages of adopting an ensemble made of neural network models combined with a time-varying dynamic factor model with a score specification (DFM-GAS) when dynamics in the mean induced by potential structural breaks affect the business cycle. We argue that models that are linear in construction do not perform properly whenever non-stationarity in the series arises, a point of view also expressed by Terasvirta and Anderson [1992] and D’Agostino et al. [2013]. In this setting, neural network models are particularly suitable to predict processes affected by mean-shifting because they take advantage of non-linear activation functions applied to the weighted
sum of neurons for each layer\textsuperscript{1}. There are both theoretical and empirical reasons why neural network models are worth using for modelling non-linear macroeconomic series. Lapedes and Farber \cite{1987} develop a simulation exercise to show that artificial neural networks accurately predict dynamic nonlinear systems, while Zhang et al. \cite{1998} highlight the convenience of using artificial neural networks as universal function approximators working without prior knowledge on the joint distribution of inputs and outputs. Besides capturing non-linearities, neural network models also avoid the curse of dimensionality - a well known issue both in macroeconomic and finance literature - because they can be represented as a composition of hierarchies of functions requiring only local computations \cite{2017}. Empirically, neural networks have been used beyond finance to compare with standard models. Loermann and Maas \cite{2019} find that multilayer perceptrons (i.e., artificial neural networks) outperform a standard DFM in both nowcasting and forecasting. Nonetheless, complex neural networks may have a huge number of neurons and layers, leading to interpretability issues. However, in the last few years we have witnessed remarkable progress in the interpretability of the results of neural networks \cite{2019}. For this reason, we introduce the computation of Shapley coefficients to provide an assessment of the predictive power for every input of the neural network, in a fashion similar to results presented in a standard regression table, as in \cite{2019}. The intuition is to propose an ensemble package that helps in shifting to neural networks when they are most needed, i.e., in periods of recession.

We test our model to predict the US quarterly GDP at different horizons in the period 2005Q2-2020Q1. The choice of the forecast window is crucial for our exercise, as we are able to trace how the models perform during the crisis in 2008-2009 and the following recovery. The weights of the ensemble are used to evaluate the forecast’s contribution to every component of the model in the final predictions. This is in line with the spirit of the work as we are evaluating the predictive ability of a time-varying dynamic factor model (DFM-GAS) against two types of recurrent neural networks: Long-Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). We made the choice of using a neural network with recurrent components in order to better capture the memory-dependence properties of the GDP series. When the structure of weights shifts almost completely towards the neural networks during a crisis we can conclude that these components perform better in recession periods.

The rest of the work is organized as follows. Section 2 describes our methodological approach and Section 3 describes data. Results are presented and discussed in Section 4. Section 5 summarizes findings and discusses promising future developments.

\textsuperscript{1}For an accurate mathematical representation, please see Géron \cite{2019}.
2 Methods

In our empirical analysis we combine a score-driven dynamic factor model (DFM-GAS) with recurrent neural networks (RNNs) in an ensemble model to predict the growth rate of the US GDP. Section 2.1 provides an overview of factor models with dynamics in mean, whereas Section 2.2 illustrates the recurrent neural network approach we use in this paper.

2.1 Score-driven dynamic factor model

The standard dynamic factor model (DFM) for GDP nowcasting was introduced by Giannone et al. [2008b]. The model uses the information available during the quarter for nowcasting the current period of economic activity measured by the GDP growth rate. The idea is to estimate the value of the GDP growth rate when it is not promptly available by using higher frequency variables released in a more timely manner. A vector of \( N \) monthly time series \( x_t = (x_{1t}, x_{2t}, ..., x_{Nt}) \) is transformed to satisfy the weak stationarity assumption so that the general DFM specification is given by the following equations:

\[
x_t = \mu + \Lambda f_t + \varepsilon_t \tag{2.1}
\]

\[
f_t = \sum_{i=1}^{p} A_i f_{t-i} + B u_t, \quad u_t \sim i.i.d.N(0, I_q) \tag{2.2}
\]

In equation (2.1), the monthly indicators are driven by two unobserved stationary stochastic processes that consist in the factor dynamics \( f_t \) (through \( \Lambda \)) and the random innovations \( \varepsilon_t \). The factors are modelled as a stable VAR(\( p \)) process. Both \( \varepsilon_t \) and \( u_t \) are normal and the vector of idiosyncratic components \( \varepsilon_t \) is unrelated to \( u_t \) at all lags, i.e. \( E[\varepsilon_t u_{t-k}'] = 0 \) for any \( k \). In this setting a number of factors driving the economy have to be specified: this number represents the dimension of the \( f_t \) vector. A number of lags \( p \) as well as a number of shocks \( q \) also have to be indicated for the \( f_t \) dynamics. It is worth noticing that the number of shocks \( q \) do not need to be equal to the number of factors because of matrix \( B \).

Parameters are estimated with a two-stage approach. In the first stage, a standardized balanced panel \( \bar{X}_t \) is used to estimate \( \Lambda \) and \( f_t \) by Principal Component Analysis (PCA). The estimators \( \hat{\Lambda} \) and \( \hat{f}_t \) are obtained by solving the following minimization problem:

\[
\min_{f_1, ..., f_T, \Lambda} \frac{1}{NT} \sum_{t=1}^{T} (\bar{X}_t - \Lambda f_t)' (\bar{X}_t - \Lambda f_t) \quad \text{s.t.} \quad N^{-1} \Lambda' \Lambda = I_r \tag{2.3}
\]
The variance-covariance matrix estimator for $\varepsilon_t$ is given by:

$$
\hat{\Psi} = \text{diag}\left( \frac{1}{T} \sum_{t=1}^{T} \left( \bar{X}_t - \hat{\Lambda} \hat{f}_t \right) \left( \bar{X}_t - \hat{\Lambda} \hat{f}_t \right)' \right) \tag{2.4}
$$

The estimated vector $\hat{f}_t$ is the principal component of $\bar{X}_t$ and the coefficients of the VAR equation (2.2) are estimated by ordinary least squares (OLS). In the second stage, Kalman smoothing (Durbin and Koopman [2012]) is used to re-estimate the factors for the unbalanced panel, $x_t$, considering the parameters obtained in the previous step.

Once monthly factors $\hat{f}_t$ are identified by principal component analysis (PCA) and Kalman smoothing, a bridge equation is used to estimate parameters and forecast the dependent variable, which is GDP at a quarterly frequency $^{2}$:

$$
y_t = \alpha + \beta' \hat{f}_t + e_t \tag{2.5}
$$

The $h$-step ahead forecast is computed as follows:

$$
y_{t+h} = \alpha + \beta' \hat{f}_{t+h} \tag{2.6}
$$

and $f_{t+h}$ is computed with equation 2.2 by using a lag $p$ that is at least large as the number of step-ahead forecasts $h$. When we are dealing with potential structural breaks in the GDP equation, such as recessions, we may want to account for time-varying features of the data generating process. In particular, it is reasonable to assume a potential change in the mean when structural breaks occur. Assuming a time-varying process for the first moment of the GDP can therefore help by improving in-sample fit as well as predicting the future evolution of the dependent variable. For this reason, we adopt an observation-driven approach to account for time-variation of the mean and this can be modelled with a Generalized Autoregressive Score (GAS) model. Indeed, it is reasonable to believe that also the second moment varies over time as in Antolin-Diaz et al. [2021]. However, we propose a model that can be more easily estimated - given that only one parameter varies - and which is mutually used with a recurrent neural network. We still believe that a model with variation in the second moment may be a solid option to forecast GDP in an ensemble with RNN.

In a score-driven framework, we define a set of observations for the dependent variable $Y^t = \{y_1, \ldots, y_t\}$, a set of time-varying parameters $F^t = \{\alpha_0, \alpha_1, \ldots, \alpha_t\}$ and a vector of static parameters $\theta$. The information set at time $t$ consists in $\{\alpha_t, F_t\}$ where:

---

$^{2}$Please note that, accordingly, factors are aggregated in order to represent quarterly quantities.
\[ \mathcal{F}_t = \{ Y^{t-1}, F^{t-1}, X^t \}, \text{ for } t = 1, \ldots, n \]

We assume \( y_t \) to have the following observation density:

\[ y_t \sim p(y_t \mid \alpha_t, \mathcal{F}_t; \theta) \quad (2.7) \]

the vector of time-varying parameters has the following specification:

\[ \alpha_{t+1} = \omega + \sum_{i=1}^{p} A_i s_{t-i+1} + \sum_{j=1}^{q} B_j \alpha_{t-j+1} \quad (2.8) \]

which is determined by an autoregressive component and by \( s_{t-i} \), defined as:

\[ s_t = S_t \cdot \nabla_t, \quad \nabla_t = \frac{\partial \ln p(y_t \mid \alpha_t, \mathcal{F}_t; \theta)}{\partial f_t}, \quad S_t = S(t, \alpha_t, \mathcal{F}_t; \theta) \quad (2.9) \]

In this way, the time-varying vector is updated to the next period using the score function \( \nabla_t \). \( S_t \) is a scaling matrix used to control the parameter updates driven by the score.

In this exercise we assume GDP to have a time-varying mean with Gaussian innovations, such that the observation density of \( y_t \) is defined as:

\[ p(y_t \mid f_t, \mathcal{F}_t; \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[ -\frac{(y_t - \alpha_t - \beta'f_t)^2}{2\sigma^2} \right] \quad (2.10) \]

In this case, the time-varying parameter is a scalar value representing the first moment of the dependent variable. In order to simplify computation, we specify a GAS of order 1 for \( \alpha_t \), which allows a parameter updating based on the previous score and lagged value.

We set the scaling parameter as the inverse of the information matrix with respect to the time-varying parameter (in this case the information matrix is a scalar):

The equation for the time-varying mean is the following:

\[ \alpha_t = \gamma s_{t-1} + \alpha_{t-1} \quad (2.11) \]

The original process should include a constant and a parameter for the lagged value of \( \alpha_t \). The intuition behind our approach is to estimate a random walk process without an intercept. Indeed, the parameter of \( \alpha_{t-1} \) is restricted to be equal to one. In this way we also simplify computation as we only have one parameter to estimate from equation 2.11. We estimate the model via maximum likelihood where the time-varying process is identified recursively in the normal log-likelihood equation \(^3\).

\(^3\)A grid-search algorithm is implemented to initialize the maximum likelihood estimation procedure.
2.2 Recurrent neural networks

Artificial neural networks are machine learning algorithms widely used for prediction purposes. These models take a collection of numerical inputs multiplied by weights by means of a forward pass process, therefore creating linear combinations between them. The linear combinations are passed through the network (from bottom to top), activating neurons with the use of an activation function that is in general nonlinear. Neurons are activated for one or more layers in the network until an output is computed. This forward pass mechanism is clearly unsupervised in the sense that an output variable is computed simply by a non-linear combination of some inputs. In other words, neural networks do not need a mapping function from input to output as they are only required to learn the underlying input structure in order to produce an output. This is why they are particularly useful for nowcasting and forecasting problems, where a time series of known inputs is used to predict an output variable that is generally unknown.

Specifically, we propose Recurrent Neural Networks (RNN) to forecast the quarterly GDP growth rate. A RNN works as a feed-forward neural network: the latter makes the neuron activation flow in just one direction - from input to output - while the former has also connections pointing backwards. In a multi-layer perceptron, which is one of the most common feed-forward neural networks, at every time step \( t \) the neuron receives a set of inputs measured at time \( t \). In a recurrent neural network, the neurons receive inputs measured at time \( t \) as well as output created at \( t-1 \). In this sense, a RNN stores memory of the previous output, which is a non-linear combination of the inputs measured in the previous step.

Within the context of GDP forecasting, the network collects information regarding two components: macroeconomic indicators are passed through the layers at every time-step and they are non-linearly combined with the output generated at the previous time observation. The following equation represents the output generated at time \( t \):

\[
Y_t = \phi (X_t \cdot W_x + Y_{t-1} \cdot W_y + b)
\]

\[
= \phi \left[ \begin{bmatrix} x_t & Y_{t-1} \end{bmatrix} \cdot W + b \right] \text{ with } W = \begin{bmatrix} W_x \\ W_y \end{bmatrix}
\]

- \( Y_t \) is a \( m \times n_{\text{neurons}} \) matrix containing the layer’s outputs at time step \( t \) for each instance in the mini-batch, where \( m \) is the number of observations in the mini-batch and \( n_{\text{neurons}} \) is the number of neurons;
- \( X_t \) is a \( m \times n_{\text{inputs}} \) matrix containing the inputs for all observations;

Results of a simulation exercise are available upon request.

\( ^4 \)Please, see Géron [2019] for more details.
• \( W_x \) is a \( n_{\text{inputs}} \times n_{\text{neurons}} \) matrix;

• \( W_y \) is a \( n_{\text{neurons}} \times n_{\text{neurons}} \) matrix containing the connection weights for the outputs of the previous time step;

• The weights’ matrices \( W_x \) and \( W_y \) are often concatenated into a single weight matrix \( W \) of shape \( (n_{\text{inputs}} + n_{\text{neurons}}) \times n_{\text{neurons}} \);

• \( b \) is a vector of size \( n_{\text{neurons}} \) containing each neuron’s bias term.

In this model, \( Y_t \) is a function of \( X_t \) and \( Y_{t-1} \), which is a function of \( X_{t-1} \) and \( Y_{t-2} \) and so on. This makes the output at time \( t \) a function of all the previous time-step inputs. The recursive structure of a RNN is optimal for time series analysis as it stores memories of previous time information. This enables us to avoid the use of too many lagged inputs, mitigating the risk of overfitting, as the autoregressive component is already captured by the model structure.

The RNN hyperparameters are trained with the classic backpropagation algorithm. Directly after the forward pass and the computation of an output, a loss is calculated for the entire training set by comparing the predicted output with the actual one (supervised part of the neural network). With the backward pass, the weights are updated according to the loss. The latter mechanism works thanks to the stochastic gradient descent algorithm. That is, the gradient of the loss function gives the direction to move weights onto the next iteration. We use two different types of RNN: i) Long Short-Term Memory (LSTM); ii) and Gated Recurrent Unit (GRU)\(^5\).

**LSTM** was introduced by Hochreiter and Schmidhuber [1997], and its main feature is the identification of a short-term and a long-term state. The algorithm is able to recognize an important input and store it in the LTSM. The network will learn and extract information on the input whenever this is needed. In practice, the LSTM works by managing two vectors: \( h_{t-1} \) and \( c_{t-1} \). \( h_{t-1} \) is the short-term state and represents the output generated at time \( t - 1 \), while \( c_{t-1} \) is the long-term memory component. In an LSTM cell the current input vector \( x_t \) and the previous output \( h_{t-1} \) are fed to four fully connected layers. One of them is the main layer and it has the role of analyzing the two vectors creating the current output. The other three layers are gate controllers and they use a logistic activation function\(^6\):

• **Forget gate** controls which part of the LTSM is not significant for the current output estimation;

\(^5\)Among the advantages of using RNNs, we consider the possibility to solve the problem of the vanishing gradient, see for example Hochreiter [1998].

\(^6\)Outputs range from 0 to 1, and the gate is opened when the output is 1.
• Input gate controls which part of the output from the main layer contributes to the long-term state;

• Output gate controls which part of the long-term state should be read as an output for the current time step.

Therefore, LSTMs are able to capture short-term as well as long-term dependencies in the data.

GRU was introduced by Cho et al. [2014] as a simplified version of the LSTM, given that it performs in a similar way. It follows the same concept as long/short-term dependencies, but here the two state vectors are stacked in a single one. A GRU cell is composed of a reset gate and an update gate:

• Reset gate controls the significance of past output on current information. If past information does not appear to be important then the reset gate is opened, so that past output does not affect current input structure;

• Update gate controls whether current input should be ignored in the prediction of current output. When the update gate is fully opened, a short-circuit connection is created, making current output completely dependent on past output.

As for the estimation, neural networks have a large number of hyperparameters that need to be tuned in the training and validation process. We train the following hyperparameters: i) number of layers; ii) number of nodes; iii) number of epochs; iv) activation function; iv) optimizer for SGD; v) batch size.

2.3 Shapley values and gradient explainer

In this section, we provide a brief overview of a methodology whose aim is to understand the role of economic indicators in predicting the US GDP. The method combines the concepts of Shapley values (Joseph [2019]) and LIME (Ribeiro et al. [2016]) in a tool providing local explainability of machine learning models. This approach is based on a coalitional game-theoretical methodology by Shapley [1953] that treats the features of a machine learning models as players while the gain is the prediction of a data instance.

A Shapley value is simply the marginal contribution of a player within a game. In extending this concept to statistical learning, we assume that features are players and the model’s outcome of the instance is the game payoff. The game is now a prediction task that the features produce combined in the model’s equation.

In machine learning, the main problem in applying Shapley values is the presence of missing values: one has to assign some values in order to mimic a model where a particular
feature is absent. As a solution, we use the gradient explainer, which is a tool built according to Erion et al. [2021] and based on the concept of integrated gradient by Sundararajan et al. [2017]. Gradient explainer computes a feature’s contribution by evaluating the prediction of the fully specified model with the prediction that the model would make with the absent feature. The integrated gradient can be used for this purpose in deep learning models and is defined as:

$$
IG_i(x, x') := (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\delta f (x' + \alpha (x - x'))}{\delta x_i} d\alpha
$$

(2.12)

$x_i$ is the actual feature, while $x'_i$ represents the absent feature that would be replaced by a baseline. Integrated gradients make a path from baseline to actual feature by cumulating gradients. Integrated gradient computes which feature mostly contributes to the network’s output (by computing the gradient) along the path from baseline to the actual model. Indeed the problem here is one of missingness and the purpose of expected gradient is to find a baseline feature without arbitrarily assigning values to it. Expected gradient for feature $i$ is defined as:

$$
\text{ExpectedGradients}_i(x) := \int_{x'} x_i - x'_i \times \int_{\alpha=0}^{1} \frac{\delta f (x' + \alpha (x - x'))}{\delta x_i} d\alpha \quad p_D (x') dx'
$$

(2.13)

where $D$ is the underlying data distribution. Following the axioms shown in the paper of Erion et al. [2021], it can be re-written as:

$$
\text{ExpectedGradients}_i(x) := \mathbb{E}_{x' \sim D, \alpha \sim U(0,1)} \left[ (x_i - x'_i) \times \frac{\delta f (x' + \alpha (x - x'))}{\delta x_i} \right]
$$

(2.14)

where $x'_i$ is drawn from the training data while $\alpha$ from $U(0,1)$. More generally, gradient-based methods consider a feature salient if it makes a significant impact on the specified model when it is varied locally (in a point of the test set): indeed, that is what the gradient computes for each instance. Other gradient-based method for deep learning models are DeepLIFT Shrikumar et al. [2017] and GradSHAP Lundberg et al. [2018]. With this setup, we can compute coefficients for all the features and each data point in the test set. That is why gradient explainer is a local explainability model: this does not allow to estimate a global coefficient - as it is done in OLS - but provides a
set of local coefficients. Therefore, gradient explainer provides a linear interpretation of the model. The advantage of this method is that we can evaluate the coefficients dynamically over time, understanding the role of different indicators in the business cycle. The disadvantage is that the dynamics is reliable for the observations in the test set, but cannot be generalized for any data point and this is because of the local nature of the dynamic coefficients.

3 Data

We use two set of data to evaluate alternative versions of the model. The first dataset encompasses standard monthly predictors. We use data similar to the ones used in the seminal work by Giannone et al. [2008b] from the Federal Reserve Economic Database (FRED) for a total of 138 monthly predictors. All in all, the first dataset includes predictors from a wide range of economic releases, including information on manufacturing industries, money and credit, labor and wages, industrial production, prices, incomes, housing, interest rates, and the financial sector. In the Appendix (Section A), we report all the predictors.

Note that Giannone et al. [2008b] lists about 200 macroeconomic variables. Unfortunately, we are not able to source some predictors originally used by Giannone et al. [2008b] because they are not available from public data sources.
Figure 1: US Gross Domestic Product in the time window 1970Q1-2021Q1. The plotted blue line is the actual GDP value, whereas the gray shaded areas are NBER recession periods.

Figure 1 shows the official US real GDP rate for the period 1970Q1-2021Q1 as well as NBER recession periods in the gray shaded areas. The process seems to experience variation in mean: from our perspective, the mean-variation is partially explained by the dynamics induced by the downturns along the business cycle.

The second dataset includes also weekly and daily indicators, such as non-conventional predictors like different uncertainty measures. Particularly, we employ weekly initial claims as a high-frequency proxy of the unemployment level, as well as data from the Nasdaq stock market, variables capturing economic uncertainty, a financial stress index and exchange rates from the major currencies.

---

9Official NBER recession dates found at: https://www.nber.org/research/data/us-business-cycle-expansions-and-contractions.
10The employment of high-frequency and big data is also present in the financial econometrics literature (Bellini et al. [2020]).
11US/UK, Japan/UK, Canada/US
Table 1: Higher-frequency indicators added to dataset 1 to obtain dataset 2.

| Variables or groups of variables | Frequency of the indicators in the group | Missing values at the beginning of the series | N. of variables |
|----------------------------------|------------------------------------------|-----------------------------------------------|-----------------|
| Initial claims                   | $W$                                      | False                                         | 1               |
| Nasdaq stock market              | $D$                                      | True                                          | 10              |
| Financial stress index           | $W$                                      | True                                          | 1               |
| Uncertainty                      | $D, W, M, Q$                             | True                                          | 16              |
| Exchange rates                   | $D$                                      | False                                         | 3               |

In the second dataset, there are predictors started to be measured for a more limited span of the time series, because they appeared later in time, therefore the models have to understand that there are some predictors with missing values for initial dates. Given the unconventional nature of the task, we use the second version of the dataset only for RNN. The two datasets are separated as we want to test whether granular movements in the predictors could significantly improve forecast performances, especially in times of recessions.

4 Empirical analysis

In this section, we analyze the accuracy of recurrent neural networks, dynamic factor models, and ensemble models in predicting US GDP growth rates. The $h$-step predictions from one quarter ahead to four quarters ahead are compared in terms of forecast errors. Specifically, the root mean squared error (RMSE) of the DFM-GAS, RNN and ensemble models in the out-of-sample window are considered. We discuss the results of DFM-GAS and RNN, then we move to consider the coefficient estimation for the DFM, hyperparameters tuning for RNN, and the selection of models in the ensemble. In doing so, we construct a weight function based on the averaged inverse of the mean squared error. A test for structural breaks devised by Chow [1960] is implemented to evaluate whether the GDP process experiences changes in the parametrization of the process. A second analysis aims at evaluating the prediction algorithm during the ongoing Covid recession. We present insights on model’s interpretability in the final section of the empirical analysis.

12 Indeed the initial missing values do not allow to correctly compute the factors for the DFM-GAS and therefore the model’s performance would be compromised. For this reason the version 2 is used only on the RNN.
4.1 Dynamic factor model estimation

The factors are extracted from the panel of monthly indicators after a principal component analysis, and then they are re-estimated with Kalman smoothing. After factor estimation, the GDP equation can be identified by regressing the quarterly variable over the aggregated factors. For a fixed DFM parameter, the coefficients are estimated with least squares, while in the GAS setting the time-varying parameters are estimated with a maximum likelihood algorithm (MLE) by means of numerical methods. The MLE algorithm operates in a univariate environment since we only allow the mean of the process to vary over time and conditional normal distribution is assumed. Score-driven dynamics are induced with a GAS specification on the mean parameter of the process. We opted for a GAS specification to keep the model simple and the interpretation straightforward. The number of factor loadings is two as in Giannone et al. [2008a], while the number of shocks to the factors $q$ is chosen through information criteria by Bai and Ng [2007]. The $h$-step ahead forecast is carried out with a fixed window: the in-sample window is used to estimate the parameters and consists of 142 observations from 1970Q1 to 2005Q1 (approximately 70% of the observations) while the out-of-sample window goes from 2005Q2 to 2020Q1 for a total of 59 observations (about 30% of the observations). The parameters of DFMs are estimated in the in-sample window and they are used to forecast in the out-of-sample window. We avoid parameters updating with a rolling window scheme to test how good the models are in capturing the data generating process by using only the in-sample data. The same applies for the RNN forecast.

4.2 Recurrent neural network tuning

In the RNN environment, hyperparameters are tuned with backpropagation and a simple cross-validation algorithm is implemented to avoid overfitting. We split the data into training, validation, and test sets. The training set consists in the first 70% of the observations (1970Q1-2005Q1); the validation set is made by the last 20% of the observations in the training set; the test set includes the last 30% of the observations (2005Q2-2020Q1). The choice of the validation within the training set was made in order to use as many data as possible to train the network structure. Neural networks are all based on a 3-layer structure: the first layer consists of the RNN cell, either LSTM or GRU, the second is a hidden layer, and the final layer is the one generating the prediction output. Based on this structure, we tune a number of nodes and epochs with a grid-search algorithm. In the first step the optimal number of nodes is found for networks with a different number of epochs. Then, the optimal number of epochs is chosen by plotting accuracy in the validation set of the networks with an optimal number of nodes (see Section B in the Appendix). We fixed the seed for the initialization of every training epoch, so as to ensure
replicability of the results.

4.3 Ensemble model

The DFM and RNN models are combined in an ensemble model. The model selection is based on choosing the best DFM (between standard DFM and DFM-GAS) and the best RNN (between LSTM and GRU) in terms of prediction accuracy in the out-of-sample window. Therefore, the ensemble process is a weighted average of two different predictions. The weights of the ensemble are computed on each observation by averaging the inverse mean squared error of the out-of-sample performance. In this sense, ensemble weights will be equally initialized for the prediction of the first out-of-sample observation. From the second observation we can generalize a formula for the weights function of Model 1 in an ensemble composed of two models:

\[ W^{*}_{M_1, T+n} = \frac{1}{MSE_{M_1, T+n} + MSE_{M_2, T+n}} \] (4.1)

for \( n = 2, 3, \ldots, N \), where \( T \) is the last observation of the in-sample set, whereas \( T + N \) is the last observation of the out-of-sample set. The mean squared error is defined as:

\[ MSE^{*}_{M_1, T+n} = \frac{1}{n} \sum_{i=T}^{n} (Y_i - \hat{Y}_i)^2 \] (4.2)

where \( Y_i \) is the actual GDP and \( \hat{Y}_i \) the prediction. Therefore, the weight function has more points to compute the mean squared error for further observations from \( T \): in this sense the ensemble accuracy increases through the out-of-sample window. To facilitate the comparison of different models, we use another weight function evaluating the predictive performance of the models in the out-of-sample window. The weights are computed for comparing models pairwise (ensemble/DFM-GAS and RNN/DM-GAS) for every observation in the out-of-sample set such that:

\[ W_{M_1, t} = \frac{1}{MSE_{M_1, t} + MSE_{M_2, t}} \] (4.3)

Here only the observations at time \( t \) are used to compete the MSE, therefore comparing the predicted and the actual value of the GDP. In this way, the distance between prediction and actual measure is inversely proportional to the weights, enabling the evaluation of the performance of the models over time.
4.4 Model performance

For every $h$-step forecast, two factor models and two neural network predictions are evaluated in the out-of-sample set. The best factor model and the best neural network are combined in the ensemble to maximize forecast accuracy. The weights in equation 4.3 are used to compare models in the forecast window. The same set of macroeconomic indicators is used as inputs for DFM-GAS and RNN. Within the DFM framework the dimensionality is controlled by the number of factors. In the case of Neural Networks, the activation of layers determines the variable importance to the prediction. Every model is evaluated individually in an out-of-sample window for the period 2005Q2 to 2020Q1, with data available since 1970Q1 for training and estimation of parameters.

Table 2 reports the out-of-sample performances for all the models. The forecast accuracy is evaluated by the root mean squared error (RMSE) and results are reported for each of the models, for one up to four periods ahead. The maximum prediction horizon is therefore one year.\textsuperscript{13}

Instead of merely comparing different models, we use this exercise to see whether neural networks can help in the prediction during recessions, at least for some forecast horizons. In particular, we focus our attention on the recession period given by the 2008-09 financial crisis, since the Chow test confirms there has been a structural break during the 2008-09 crisis (see Secton C in the Appendix).

\begin{table}[h]
\centering
\begin{tabular}{lcccc}
\hline
 & $h = 1$ & $h = 2$ & $h = 3$ & $h = 4$ \\
\hline
Dynamic Factor Model, DFM (dataset 1) & 2.9137 & 2.9941 & 3.0262 & 3.0017 \\
t.v. Dynamic Factor Model, DFM-GAS (dataset 1) & 2.3191 & 2.5239 & 2.5787 & 2.4569 \\
Long short-term Memory, LSTM (dataset 1) & 2.1585 & 2.2865 & 2.3972 & 2.3520 \\
Long short-term Memory, LSTM (dataset 2) & 2.2024 & 2.2792 & 2.4297 & 2.4571 \\
Ensemble model (dataset 1) & 2.0130 & 2.2578 & 2.3327 & 2.3094 \\
Ensemble model (dataset 2) & 2.0987 & 2.1955 & 2.3265 & 2.3901 \\
\hline
\end{tabular}
\caption{The forecast accuracy of different models: the root mean squared errors (RMSE) are reported for one quarter ($h = 1$) to four quarters ahead ($h = 4$). The ensemble consists in the combination of the best-performing models: one factor model (DFM-GAS) and one neural network (LSTM). Dataset 1 includes only monthly predictors.}
\end{table}

Results in Table 2 show that the DFM-GAS is the best performing dynamic factor model in all time horizons. This makes perfect sense since the underlying process of the GDP has a mean dynamics that can be better predicted from a model accounting for variation

\textsuperscript{13}Intuitively, a longer forecast horizon implies a higher prediction error by increasing forecast uncertainty: only current and past quarter information is available to predict the next periods of GDP growth. Potential events such as endogenous and exogenous shocks occurring during the horizon will not be included in the model’s information \cite{Giannone2004}.
in the first moment. In the RNN framework, the LSTM always outperforms GRU even if performances look very similar, especially for longer forecast horizons. When using only monthly predictors (dataset 1), the LSTM always outperforms the DFM-GAS. However, for longer forecast horizons the DFM-GAS handles the comparison better: for the one-year-ahead forecast the root mean squared errors of the DFM-GAS and RNN models are similar and this suggests that the advantage of using the RNN is relevant only in the short-run. The ensemble model is the combination of the best predictions from factor models and recurrent neural networks: this means that the ensemble always uses the DFM-GAS and LSTM for all the forecast horizons, outperforming other models for all the forecast horizons. Table 2.1 shows that differences between the ensemble and the DFM-GAS decrease with a longer forecast horizon, when dataset 1 is considered. This suggests some sort of trade-off that makes it worth using a more complicated model combining a neural network and the DFM-GAS for short horizons only. For horizon $h = 1$, where the difference between the RMSE of the ensemble and that of the DFM-GAS is approximately 0.306, for $h = 4$ (four quarters ahead forecast) the difference becomes 0.147 since the DFM-GAS has a slower decrease in accuracy for longer forecast horizons. All in all, we find that there is a distinct advantage in using a neural network ensemble for a short-term forecast ($h = 1$) using a set of data entirely composed of monthly indicators.

The analysis of dataset 2, the one including weekly and daily indicators, poses some challenges because it contains short series with missing values at the beginning. In this exercise, we use the DFM as a benchmark model and we run the RNN model (LSTM) to dataset 2 with the same hyperparameters structure tuned with dataset 1. The choice of using exactly the same parameterization (number of nodes and layers, activation function, loss function, optimizer, validation set, batch size for the training, number of epochs) further generalizes the LSTM model, mitigating the risk of overfitting. In this setting, the performance of the LSTM is similar in both versions of the model, as well as the one of the ensemble. This suggests that the model’s specification is robust to change in the dataset and the ensemble still outperforms the DFM-GAS. On top of that, the LSTM may potentially increase its performance if the tuning is conducted again to adjust the model’s specification to the new dataset version.

### 4.5 Forecast comparison with weights and RMSE difference

In this section we analyze how the weights of the ensemble components, i.e. RNN and DFM-GAS, change in time. We consider the case of the US GDP one quarter ahead forecast, using weights constructed as in equation 4.3. To this end, we evaluate the differences between the ensemble and the DFM-GAS with weights constructed as in equation 4.3. 

\[ \text{RMSE}_{\text{ensemble}} - \text{RMSE}_{\text{DFM-GAS}} \]

This comparison shows that the ensemble outperforms the DFM-GAS by a margin of 0.147 for a four quarters ahead forecast, which is a significant improvement over the 0.306 difference observed for a one-quarter ahead forecast. This suggests that the ensemble model is more robust and less dependent on the specific dataset, making it a more reliable choice for long-term forecasts. The RMSE differences also highlight the importance of selecting the appropriate model based on the forecast horizon, as the advantage of using a neural network ensemble diminishes with increasing forecast horizons.

\[ \text{RMSE}_{\text{ensemble}} - \text{RMSE}_{\text{DFM-GAS}} \]

Moreover, the analysis of dataset 2, which includes weekly and daily indicators, presents some challenges due to the presence of short series with missing values at the beginning. In this setting, the LSTM model is trained with the same hyperparameters structure as the one for dataset 1, ensuring that the model can be generalized across different datasets. The performance of the LSTM model is found to be similar in both versions of the model, which indicates that the model’s specification is robust to changes in the dataset. Furthermore, this suggests that the LSTM model may have the potential to improve its performance if the tuning process is adjusted to accommodate the characteristics of the new dataset version.

\[ \text{RMSE}_{\text{ensemble}} - \text{RMSE}_{\text{DFM-GAS}} \]

Finally, it is worth noting that the ensemble model is a combination of the best predictions from factor models and recurrent neural networks, which allows it to leverage the strengths of both approaches. By using the DFM-GAS and LSTM for all forecast horizons, the ensemble model outperforms other models for all forecast horizons, making it a versatile and effective choice for both short-term and long-term forecasting tasks.

\[ \text{RMSE}_{\text{ensemble}} - \text{RMSE}_{\text{DFM-GAS}} \]

14Recently, Lewis et al. [2020] and Antolin-Diaz et al. [2021] proposed methods to integrate shorter series in a DFM framework to enhance accuracy in the estimates.
contribution of the models in the ensemble as regards the whole out-of-sample window. The difference between the predicted values of each model is used to understand how the performance of the ensemble, RNN and DFM-GAS vary over time.

Figure 2: US GDP one quarter ahead forecast with DFM-GAS ($h = 1$), LSTM and ensemble as in version 1 of the dataset (first panel); DFM-GAS weights constructed as in equation 4.3 (second panel). Gray shaded area is the NBER recession period of the 2008-09 crisis.

Figure 2 (top panel) shows real GDP growth rate with model predictions for the one-quarter-ahead forecast. The LSTM is the best performing single model with an RMSE of 2.1585. The performance accuracy is further improved with the ensemble model, which produces a lower RMSE of 2.0476. The LSTM forecast is clearly flattened compared to the DFM-GAS one. The intuition behind this relates to the factors estimation updated every time-step in the out-of-sample window: indeed, despite the fixed-window nature of our forecast, the factors need to be estimated with the newly released indicators that serve as predictors in the model. For the LSTM the new observations of the predictors are just fed into the trained model to compute the output, without an actual estimation and this may be the main reason of the lower volatility of that forecast.

In the time window considered, the period 2008-09 (gray shaded area) can be seen as a structural break in the US economic cycle that moved the mean of the process downward\textsuperscript{15}. The second panel shows the corresponding weights for the DFM-GAS, con-

\textsuperscript{15}See the results of the Chow test for structural breaks in the appendix
structured as in equation 4.3 and pairwise averaged in order to mitigate noise. Weights for the DFM-GAS are plotted in the out-of-sample window to evaluate its contribution compared to the ensemble. When weights are low it means that the RNN is outperforming the DFM-GAS for that set of observations. During the 2008-09 crisis the weights reach a minimum and it is clear that the ensemble is the best model at capturing the downturn and recovery of the economy.

![Figure 3](image1.png)

Figure 3: Averaged weights of the DFM-GAS in the ensemble model for $h = 2$ (left), $h = 3$ (middle), $h = 4$ (left).

Figure 3 shows the DFM-GAS weights throughout the entire out-of-sample window for different forecast horizons. It is clear that the time-varying DFM loses importance during the 2008-09 crisis because the weights almost completely move toward the RNN prediction, and this happens particularly for shorter forecast horizons. This means that despite potential differences in loss, the neural network ensemble outperforms the DFM-GAS during the 2008-09 crisis. The same conclusion is reached by comparing the DFM-GAS with the single RNN, as shown in the Appendix.

In Appendix we provide a visual comparison of the models’ predictions by plotting the absolute difference of the DFM-GAS and RNN forecasts. From the plots we notice that there is a significant difference between the models’ outcomes during the financial crisis, also supported by a Diebold-Mariano test, especially in the case of short-term forecast horizons.

### 4.6 Prediction during the (ongoing) COVID-19 crisis

In this section, we use our methodology to predict the COVID-19 shock to the US GDP. The economic crisis started during the second quarter of 2020 and, for the moment, official data up to the first quarter of 2021 are available. To improve predictions during the COVID-19 crisis, we enlarge the dataset of predictors adding weekly and daily indicators aggregated at a monthly frequency (version 2 of the dataset). We evaluate the effect of the new variables on the prediction performances of both the DFM-GAS and the RNNs. We test the models with the monthly dataset (dataset 1) and with the extended
dataset 2. The hyperparameters tuning is conducted again only for RNN in version 2 of the model, where the features change: indeed, version 1 is the most generalized model because it keeps the original network structure parameterized in the previous forecast exercise. Therefore, one neural network is tuned only before the 2008-09 crisis (version 1), and one neural network is re-tuned after the 2008-09 crisis. The rationale behind the re-tuning is to update the network structure to potential shifts of the economic cycle after the Great Recession. Please note that tuning can be done with a regular yearly frequency. With the new sample, the models are trained and estimated for 1970Q1-2015Q4, thus forecast performance is evaluated for 2016Q1-2021Q1.

| RMSE (2016Q1-2019Q4): | DFM-GAS (dataset 1) | LSTM (dataset 1) | LSTM (dataset 2) |
|------------------------|---------------------|------------------|------------------|
| 1.2657                 | 1.1203              | 1.1079           |

Table 3: RMSE of the one-quarter ahead forecast (2016Q1-2019Q4).

Table 3 reports RMSE up to 2019Q4 in the test set for a one-quarter ahead forecast. The macroeconomic environment was clearly in regular times and the forecast error is significantly reduced. The LSTM with high-frequency indicators performs best even if there are no significant differences in terms of forecast performance between the three models. When it comes to data points related to the COVID-19 crisis starting in 2020Q1, predictions become more challenging. The first quarter of 2020 sees first official cases on the records of the global pandemic outbreak. Thus, a massive drop in the US GDP is estimated for the second quarter of 2020. For the way the models take in the new inputs, it is almost impossible to predict 2020Q2. Indeed information about COVID-19 is recovered just after 2020Q1, and the one-quarter ahead prediction during that same quarter does not correctly recognize the shock that pandemic was causing to the economy, as shown in Figure 4.
Figure 4: Plot of the official GDP percent quarterly change with summarized timeline of COVID-19 related events. Predictions for DFM-GAS and the best performing LSTM model are provided, as well as the log difference of the economic policy uncertainty index (daily aggregated at monthly frequency).

The peak of the pandemic has been reached between March and Q2-2020. The pandemic had a severe impact on the US economy, with an official GDP fall of -32.8. The LSTM model (dataset 2) is the only model that correctly catches information on pandemics after the first quarter and predicts the sudden recovery of the GDP in the third quarter. On the other hand, the DFM-GAS and LSTM based on dataset 1 are flat in the estimates and do not correctly predict the third quarter of 2020. The log rate of the economic policy uncertainty index is a daily index that we aggregate at a monthly frequency, and it is more volatile around March 2020.

4.7 Model interpretation

To make the results of the RNN interpretable, we use expected gradients to estimate Shapley values for each features. With expected gradient we interpret the model’s prediction within the RNN framework. With this approach, we can get local coefficients for all the variables in the test set. Our goal is to evaluate how predictors contribute to the GDP forecast along the entire business cycle. After computations, we aggregate coefficients by averaging across groups of variables. With a gradient explainer, we can...

16 The estimation is performed with a python package called SHAP.
compute coefficients for every point in the test set, providing a local interpretation of the models. If a coefficient for a variable - or group of variables - increases over time, that does not necessarily mean that predictability is increasing: in that case, that feature of the model increases its impact on the estimated dependent variable. Therefore a high coefficient means that the feature has a significant impact (negative or positive) on the variable the model is predicting.

Figure 5: Shapley values for the LSTM (version 2 of the dataset) (time horizon without COVID-19 crisis).

Figure 5 shows the Shapley values for the best performing RNN in the period when excluding the COVID crisis. We present coefficients for the most influential groups in the dataset as well as commercial paper outstanding that have the most significant coefficient right before the financial crisis. This indicator represents the number of debt instruments issued by companies to finance short-term loans. The value constantly increased from 2004 until the financial crisis when it experienced a massive drop for the cut in the credit line. The Shapley coefficients are largely negative for high amounts of commercial paper outstanding. The Housing group includes variables summarizing the demand/supply of private houses in the US market. The group provides a positive contribution to the GDP before the financial crisis and starts to lower the economic cycle right after. In this case, the vertiginous decline of demand/supply of houses perfectly fits in the economic turmoil caused by the financial crisis. The Banking Reserves group includes predictors for reserves and borrowings from the Federal Reserve. Clearly, a rise in these variables may indicate that liquidity is injected to face the crisis. Its coefficients find stability following the economic cycle. We include the group of Uncertainty as well as the Financial Stress Index and Initial Claims from the extended dataset (dataset 2). The group of Uncertainty
indicators are based on indicators constructed at different frequencies (daily, monthly and quarterly), and related to policy-uncertainty. The indexes are constructed by Baker et al. [2016]. We report the Shapley values for monthly uncertainty 17: they do not seem to play a significant role in GDP forecasts and, more interestingly, they do not show signs of change during the financial crisis. The Shapley values related to these variables seem not to be impacted by the financial crisis. Initial Claims are the least significant variables for GDP prediction. They indicate the number of workers that formally make an inquire for receiving unemployment benefits. This is an important indicator that anticipates a potential higher level of unemployment. Indeed, the demand fall makes unemployment levels jump in the nearest quarter. When the shock is clearly defined on the demand side, the economy experiences an imminent increase in weekly claims. The financial nature of the crisis does not suddenly push weekly claims but generates unemployment in future periods, and this is in line with the nature of the 2008-09 recession. During the COVID-19 crisis, LSTM is able to capture economic recovery right after the fall of GDP in the second quarter of 2020.

Interestingly enough, Figure 6 shows that measures of uncertainty play a significant role in prediction during the pandemic, and this is in contrast with the financial crisis of the previous decade, when monthly uncertainty coefficients are somehow stable. Uncertainty represents a group of daily indexes contributing negatively to the prediction of the next quarter’s GDP release, while monthly and quarterly uncertainty pushes up the GDP estimates. This comes from the fact that initial concerns are firstly captured by the daily index and caused a negative impact on the GDP. Monthly and quarterly indexes officially measure uncertainty after the governments’ initial intervention decision to boost the economy during and after the lockdown. Therefore they incorporate ”hope” in the average economic agent lowering the uncertainty. Moreover, with the SHAP analysis of uncertainty group, we match results of Baker et al. [2020] stating that Covid significantly induced uncertainty, as well as results from Foroni et al. [2020] about the differences in the two crises 2008-09 and Covid: in the latter, we had better financial conditions and more timely policy response from the government in order to recover the economy from the lockdown. The unemployment group of indicators increases the SHAP coefficients during both two crises; therefore, the model follows the economic business cycle forecasting an economic recovery after a considerable increase in the unemployment level. Initial claims and uncertainty are higher frequency indicators (weekly and daily): they are more significant during the crisis and moves ”quicker” compared to the monthly/quarterly indicators. The role of variation in initial claims is different from the financial crisis in 2008-09: lockdown measures had a sudden impact on workers that immediately asked for unemployment benefits. The model captures this, indeed claims coefficient move faster

17Shapley values for uncertainty index measured at different frequencies are flat for all the period considered.
and more pronounced compared to the unemployment group.

**Figure 6:** Shapley values for the best performing RNN model (dataset 2), full time series including the COVID-19 crisis.

Housing and commercial paper coefficients are not affected by the COVID-19 crisis like they are during 2008-09, and this suggests that the RNN incorporates information about the type of the crisis through the Shapley coefficients. Financial stability was seriously compromised during the 2008-09 crisis, while unemployment levels were at their highest after lockdown measures. The initial period of the pandemic was imminently followed by a huge volatility on the financial markets. In this situation the financial stress index starts to play a significant role in the prediction of the future GDP, impacting negatively to its future estimate. From Figure 6 it is more evident that there were no predictors anticipating the vast drop in the economic activity due to the sudden lockdown. This was an unprecedented shock given by a pandemic that could be explained by a model by considering the number of COVID-19 cases and the number of full hospitals in the country. Despite the drop, the RNN model is able to re-shuffle the role of features to predict an imminent economic recovery. Coefficients are approximately zero for the whole sample before the crisis. This reflects stability of the economic cycle and suggests that GDP has an autoregressive, potentially stationary dynamics.

Overall, despite differences in the network structure of LSTM version 1 and 2 for the prediction in the sample with COVID, version 2 of the model has the best performance because it contains a range of predictors that are crucial for understanding the severity of the crisis. Indeed, the variables related to uncertainty played a central role in the estimates of the future GDP\(^{18}\). Despite the degree of information captured by the Shapley

\(^{18}\)The prompt response of the Shapley coefficients suggest that the model well adapts to the crisis.
values about the predictors’ role, we cannot make conclusion about any causal relationship between model’s inputs and dependent variable (GDP). Recent studies are giving address to the problem of causality (Moraffah et al. [2020]) and should be taken in consideration for further analysis.

5 Conclusions

In this paper, we contribute to developing an integrated ensemble approach to predict future GDP releases. Although the contribution of machine learning to GDP forecasting has been broadly discussed in previous literature, no clear consensus has yet been reached on integrating it with traditional methods to improve prediction accuracy. In this work, we introduce an ensemble package in which recurrent neural networks are compared with time-varying DFMs at a first stage, and then they are mutually employed in an ensemble model that adapts the predictions according to the current phase of the economy. When the economy experiences a structural break, like in a sudden and unexpected shock, the data generating process may change and the main issue for the analyst is that he cannot be aware of neither the intensity nor the impact of the shock on the business cycle, be it permanent or temporary. In this context, we argue that every econometric model making solid assumptions about the underlying GDP specification suffers out-of-sample, even when accounting for parameter variations. This is the reason why we propose a combination that tries to put together the benefits of a time-varying DFM and a recurrent neural network (RNN). From an application to US GDP growth rates, we find that the ensemble outperforms a single time-varying DFM with a considerable gap for shorter forecast horizons. The marginal gains of using the ensemble decrease when longer forecast horizons are considered. Indeed, the difference in accuracy between the neural network ensemble and a single DFM-GAS is low for one-year forecasts, as confirmed in the RMSE difference analysis and by a Diebold-Mariano test. We believe the massive gap with short-term forecasts is that out-of-sample structural breaks imply a higher level of complexity, which is better handled with models such as neural networks. This is not the case for long-term forecasts. We found that RNN outperforms the DFM for one-quarter ahead forecast during the third quarter of the Covid recession (sudden economic recovery). With the SHAP values, we find essential features that acted as the main predictors during different business cycle periods: commercial and financial played a significant role in the 2008-09 crisis, while uncertainty and claims determined the considerable volatility of the GDP during the COVID-19 recession. In conclusion, we provide an interpretable solution to improve forecast performance - by combining different models and monitoring their performances over time. Future research is needed to address further issues related period of the business cycle.
to the implementation of neural network models to analyze macroeconomic processes. This regards more advanced model specification and a particular focus on the frequency of re-tuning the network: periodically and after important shifts of the economy. On the other side, advanced specifications of the DFM may be employed to generate more accurate predictions. Ideally, more structured methods can be used with non-conventional data sources, for instance high-frequency indicators and textual data. In this context, the interpretability tools give insights into how different predictors relate to the current (nowcast) and the future (forecast) economic activity.

References

Elena Andreou, Eric Ghysels, and Andros Kourtellos. Should macroeconomic forecasters use daily financial data and how? *Journal of Business & Economic Statistics*, 31(2): 240–251, 2013.

Juan Antolin-Diaz, Thomas Drechsel, and Ivan Petrella. Advances in nowcasting economic activity: Secular trends, large shocks and new data. 2021.

Susan Athey. The impact of machine learning on economics. In *The economics of artificial intelligence: An agenda*, pages 507–547. University of Chicago Press, 2018.

Susan Athey and Guido W Imbens. The state of applied econometrics: Causality and policy evaluation. *Journal of Economic Perspectives*, 31(2):3–32, 2017.

Andrii Babii, Eric Ghysels, and Jonas Striaukas. Machine learning time series regressions with an application to nowcasting. *arXiv preprint arXiv:2005.14057*, 2020.

Jushan Bai and Serena Ng. Determining the number of primitive shocks in factor models. *Journal of Business & Economic Statistics*, 25(1):52–60, 2007.

Scott R Baker, Nicholas Bloom, and Steven J Davis. Measuring economic policy uncertainty. *The quarterly journal of economics*, 131(4):1593–1636, 2016.

Scott R Baker, Nicholas Bloom, Steven J Davis, and Stephen J Terry. Covid-induced economic uncertainty. Technical report, National Bureau of Economic Research, 2020.

Marta Baïbura, Domenico Giannone, Michele Modugno, and Lucrezia Reichlin. Nowcasting and the real-time data flow. In *Handbook of economic forecasting*, volume 2, pages 195–237. Elsevier, 2013.

Matteo Barigozzi, Marc Hallin, Stefano Soccorsi, and Rainer von Sachs. Time-varying general dynamic factor models and the measurement of financial connectedness. *Journal of Econometrics*, 2020.
Vittorio Bellini, Massimo Guidolin, and Manuela Pedio. Can big data help to predict conditional stock market volatility? an application to brexit. In International Conference on Machine Learning, Optimization, and Data Science, pages 398–409. Springer, 2020.

Dario Buono, Gian Luigi Mazzi, George Kapetanios, Massimiliano Marcellino, and Fotis Papailias. Big data types for macroeconomic nowcasting. Eurostat Review on National Accounts and Macroeconomic Indicators, 1(2017):93–145, 2017.

Maximo Camacho, Gabriel Perez-Quiros, and Pilar Poncela. Markov-switching dynamic factor models in real time. 2012.

Kai Carstensen, Markus Heinrich, Magnus Reif, and Maik H. Wolters. Predicting ordinary and severe recessions with a three-state markov-switching dynamic factor model: An application to the german business cycle. International Journal of Forecasting, 36(3): 829 – 850, 2020. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2019.09.005. URL http://www.sciencedirect.com/science/article/pii/S0169207019302493.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014.

Gregory C Chow. Tests of equality between sets of coefficients in two linear regressions. Econometrica: Journal of the Econometric Society, pages 591–605, 1960.

Jacopo Cimadomo, Domenico Giannone, Michele Lenza, Francesca Monti, and Andrej Sokol. Nowcasting with large bayesian vector autoregressions. 2021.

Philippe Goulet Coulombe, Maxime Leroux, Dalibor Stevanovic, and Stéphane Surprenant. How is machine learning useful for macroeconomic forecasting? arXiv preprint arXiv:2008.12477, 2020.

Drew Creal, Siem Jan Koopman, and André Lucas. Generalized autoregressive score models with applications. Journal of Applied Econometrics, 28(5):777–795, 2013.

Antonello D’Agostino, Luca Gambetti, and Domenico Giannone. Macroeconomic forecasting and structural change. Journal of applied econometrics, 28(1):82–101, 2013.

Marco Del Negro and Chris Otrok. Dynamic factor models with time-varying parameters: measuring changes in international business cycles. FRB of New York Staff Report, (326), 2008.
James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*. Oxford university press, 2012.

Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott M Lundberg, and Su-In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence*, pages 1–12, 2021.

Claudia Foroni, Massimiliano Giuseppe Marcellino, and Dalibor Stevanović. Forecasting the covid-19 recession and recovery: Lessons from the financial crisis. 2020.

Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O’Reilly Media, 2019.

Domenico Giannone, Lucrezia Reichlin, and Luca Sala. Monetary policy in real time. *NBER macroeconomics annual*, 19:161–200, 2004.

Domenico Giannone, Lucrezia Reichlin, and David Small. Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics*, 55(4): 665–676, 2008a.

Domenico Giannone, Lucrezia Reichlin, and David Small. Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics*, 55(4): 665–676, 2008b.

Philippe Goulet Coulombe, Massimiliano Marcellino, and Dalibor Stevanovic. Can machine learning catch the covid-19 recession? *Available at SSRN 3796421*, 2021.

Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Andreas Joseph. Shapley regressions: A framework for statistical inference on machine learning models. 2019.

Iebeling Kaastra and Milton Boyd. Designing a neural network for forecasting financial. *Neurocomputing*, 10:215–236, 1996.

Dimitris Korobilis. Assessing the transmission of monetary policy using time-varying parameter dynamic factor models. *Oxford Bulletin of Economics and Statistics*, 75(2): 157–179, 2013.
Alan Lapedes and Robert Farber. Nonlinear signal processing using neural networks: Prediction and system modelling. Technical report, 1987.

Jim Lee. Measuring business cycle comovements in europe: Evidence from a dynamic factor model with time-varying parameters. Economics Letters, 115(3):438–440, 2012.

Daniel J Lewis, Karel Mertens, James H Stock, and Mihir Trivedi. Measuring real activity using a weekly economic index. Federal Reserve Bank of New York, 920, 2020.

Julius Loermann and Benedikt Maas. Nowcasting us gdp with artificial neural networks. 2019.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In Proceedings of the 31st international conference on neural information processing systems, pages 4768–4777, 2017.

Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. Nature biomedical engineering, 2(10):749–760, 2018.

Raha Moraffah, Mansooreh Karami, Ruocheng Guo, Adrienne Raglin, and Huan Liu. Causal interpretability for machine learning-problems, methods and evaluation. ACM SIGKDD Explorations Newsletter, 22(1):18–33, 2020.

Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. International Journal of Automation and Computing, 14(5):503–519, 2017.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ” why should i trust you?” explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pages 1135–1144, 2016.

Adam Richardson, Thomas van Florenstein Mulder, and Tuğrul Vehbi. Nowcasting gdp using machine-learning algorithms: A real-time assessment. International Journal of Forecasting, 2020.

Lloyd S Shapley. A value for n-person games. Contributions to the Theory of Games, 2 (28):307–317, 1953.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In International Conference on Machine Learning, pages 3145–3153. PMLR, 2017.
Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

Timo Terasvirta and Heather M Anderson. Characterizing nonlinearities in business cycles using smooth transition autoregressive models. *Journal of applied econometrics*, 7(S1):S119–S136, 1992.

Andrew Tiffin. Seeing in the dark: a machine-learning approach to nowcasting in lebanon. 2016.

Guoqiang Zhang, B Eddy Patuwo, and Michael Y Hu. Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, 14(1):35–62, 1998.

## Appendices

### A Data (quarterly prediction)

| Groups of variables         | Frequency | N. of variables |
|-----------------------------|-----------|-----------------|
| Labor                       | M         | 20              |
| Manufacturing               | M         | 7               |
| Monetary                    | M         | 4               |
| Reserves/federal surplus    | M         | 3               |
| Banking                     | M         | 5               |
| Capacity/ industrial        | M         | 23              |
| Housing                     | M         | 5               |
| Sales                       | M         | 11              |
| CPI/PPI                     | M         | 22              |
| Income and consumption      | M         | 10              |
| Interest rate and bonds     | M         | 21              |
| Trade                       | M         | 4               |
| Assets/liabilities          | M         | 8               |
| Other indicators            | M         | 6               |

Table 4: Monthly indicators for model in version 1.
Table 5: Higher-frequency indicators added to version 1 in order to generate version 2 of the model.

The full list of variables with FRED code and sources is available upon request.

B Neural networks training

RNN training requires the choice of many hyperparameters which have to balance complexity in order to optimize prediction and mitigating the risk of overfitting.

B.1 Data normalization

Data are normalized before being introduced in the neural network algorithm: this is to simplify the model to learn patterns behind the inputs which present different scaling. The min-max scaler is introduced and consists in the following transformation:

\[
\frac{x_i - \min(x)}{\max(x) - \min(x)}
\]

B.2 Activation function

The activation function used for the quarterly anaysis is the scaled exponential linear unit (SELU):

\[
SELU(x) = \lambda \begin{cases} 
x & \text{if } x > 0 \\
\alpha e^x - \alpha & \text{if } x \leq 0
\end{cases}
\]

where \( \alpha \approx 1.67326 \) and \( \lambda \approx 1.05070 \).

B.3 Stochastic gradient descent

SGD algorithm is used for the minimization of the error loss during training. Unsupervised learning in neural network is the first part of training and works by randomly assigning weights and then activating neurons through the activation function until an
output is computed for each observation. At this stage, MSE is computed in the training and validation set and gradient is computed through SGD. This will give the direction to move weights the next iteration. This process is repeated for a pre determined number of epochs. Adam optimizer is used as a particular kind of SGD algorithm which is based on 4 parameters:

- **alpha** is the learning rate, i.e. the speed at which weights are adjusted every iteration
- **beta1** is the exponential decay rate for the first moment
- **beta2** is the exponential decay rate for the second moment
- **epsilon** is a small number used to prevent any division for 0

Adam parametrization is fixed by default: $\text{alpha}=0.001$, $\text{beta1}=0.9$, $\text{beta2}=0.999$, $\text{epsilon}=10\times10^{-8}$. The model will be evaluated by computing loss (MSE) in training as well as validation set by using weights tuned in the training sample to emphasize the out-of-sample performance. Number of epochs is chosen such that loss is minimized in both training and validation set. Neural networks are computed by using keras (API of tensorflow) and connection to tensorboard provides plot of loss and accuracy during epochs of the network training.

### B.4 Numbers of neurons and layers

Numbers of neurons and layers determine the complexity of the network: a high number requires more computational power to calculate an output value from the combination of inputs chosen. In the parameters’ tuning different combinations of neurons and layers are implemented to see how the error function varies.

### B.5 Neural network structure

In this section we provide tensorboard output of the entire neural network structure. The graph includes the operative level of the network from bottom to the top. In the graph, inputs are fed into the network in a sequential order through the RNN cells.
Figure 7: graph at operative level of LSTM one quarter ahead prediction. Data are fed from bottom to the top in the LSTM cells.

In the graph above is shown the neural network structure for the LSTM one quarter ahead prediction. Expanding the first node "sequentially" gives the conceptual frame of the graph with the structure of layers:

Figure 8: Layers in the LSTM structure.
When using the GRU a similar structure is obtained with the only difference that inputs are fed into GRU cells.

![GRU Structure Diagram](image)

Figure 9: Layers in the GRU structure.

### C Chow test

\[ y_t = \beta X_t + D_t \ast \gamma_0 \]  \hspace{1cm} \text{(C.1)}

\[ y_t = \beta X_t + D_t \ast (\gamma_0 + \gamma' X_t) \]  \hspace{1cm} \text{(C.2)}

where the GDP is regressed on a set of covariates \( X_t \) (with one in the first place of the vector in order to enable the model to estimate an intercept) and a dummy assuming values of one after the presumed period of break. In equation C.1 the dummy can be seen as a shift in the intercept, while in equation C.2 it is also multiplied by the covariates. The covariates are chosen to be the five most correlated variables with the GDP process. We run regressions by assuming a potential break for every quarter in the out-of-sample window (shifting the dummy each observation) and then using an F-statistic to test whether the restricted model is significantly different with respect to the unrestricted one. The test is implemented over the whole sample.
Figure 10: Plotted p-values for F-test of structural breaks. The horizontal red line considers a significance level of 0.05. Model 1 and Model 2 represent respectively equations C.1 and C.2.

According to the test we implemented, structural breaks occur in the period of the 2008-09 financial crisis. Notice that model 1 assumes change only in the mean of the process and the test cannot reject the null hypothesis of no break for recession at the beginnings of 1990 and 2000. This suggests that NBER recession may imply a change in the model’s coefficients but not necessarily a shift in the mean value. Indeed, this happens whenever the recession turns to affect severely the economy (2008-09).
D DFM-GAS weights

Figure 11: Averaged weights for comparison between the DFM-GAS and the LSTM for $h = 1$ (top-left), $h = 2$ (top-right), $h = 3$ (bottom-left), $h = 4$ (bottom-right).

Figure 12: Averaged weights for comparison between the DFM-GAS and the GRU for $h = 1$ (top-left), $h = 2$ (top-right), $h = 3$ (bottom-left), $h = 4$ (bottom-right).
E Differences between models forecasts (sample pre COVID-19)

For the one-quarter-ahead forecast the difference in accuracy between the ensemble and the DFM-GAS is greater and this can be better evaluated by plotting the difference between the predictions of the two approaches.

The top panels of figure 13 show the plots of the difference in the predicted values between the DFM-GAS and the ensemble one-quarter and four-quarter-ahead predictions. These two plots help us in the evaluation of differences in accuracy regarding short and long-term forecast horizons. For $h = 1$, during and near the crisis, the difference has higher values and this means that the ensemble and DFM-GAS have an increasing divergence during these periods. The difference for $h = 4$ seems to be less on average, at least until 2015, and displays small differences during the crisis. In this sense, the marginal gain of using a more complex model, such as a neural network ensemble, seems to be worthwhile for the one-quarter-ahead forecast. For the four-quarter-ahead forecast the comparative advantage is smaller and this might suggest keeping the DFM-GAS in order to simplify the model estimation during the prediction exercise. In order to verify whether the difference is due to the neural network component of the ensemble, we plot the difference between the DFM-GAS and the neural network counterpart for the one- and four-quarter-ahead forecasts.
In the bottom panel of figure 13, we plot differences between the most accurate neural network and the DFM-GAS for the one- and four-quarter-ahead forecast. The large difference during the crisis in $h = 1$ confirms that the ensemble outperforms the DFM-GAS due to the LSTM forecast. For $h = 4$ the differences between the neural network and the DFM-GAS are fewer on average and especially during the crisis: this confirms that using the ensemble is preferable for a shorter-term forecast horizon such as $h = 1$.

In order to evaluate differences between the accuracy of the forecasts, we use a Diebold-Mariano test that compares the models in the out-of-sample window. With this procedure we test the null hypothesis of equality between the accuracy of two forecasts. We use the test to compare the RNN with the DFM-GAS in short-term ($h = 1$) and long-term ($h = 4$) forecast exercises, with a focus on the period of the 2008-09 crisis. Pairwise comparison is carried out between the ensemble and the DFM-GAS, as well as the RNN counterpart of the combined process used in the ensemble with the DFM-GAS. This exercise compares the forecast over the entire out-of-sample window as well as testing the hypothesis of equality in accuracy just by considering the period of the 2008-09 crisis.

For one quarter ahead it is the LSTM while for four quarters ahead it is the GRU.
Table 6: Results for Diebold-Mariano test comparing ensemble, LSTM and GRU against the DFM-GAS for one quarter \((h = 1)\) and four quarter \((h = 4)\) ahead prediction. Column (1) represents the results for the comparison between ensemble and DFM-GAS. Column (2) represents the results for the comparison between ensemble and DFM-GAS during the crisis (first 25 observations of the out-of-sample window). Column (3) represents the results for the comparison between LSTM and DFM-GAS for \(h = 1\) and between GRU and DFM-GAS for \(h = 4\). Column (4) represents the results for the comparison between LSTM and DFM-GAS for \(h = 1\) and between GRU and DFM-GAS for \(h = 4\) during the crisis. Each cell contains DM test statistics with a different loss criterion, while p-values are shown in brackets: *** \(p < 0.01\), ** \(p < 0.05\), * \(p < 0.1\).

|                | one quarter ahead, \(h = 1\) |                  |                  |                  |
|----------------|-------------------------------|------------------|------------------|------------------|
|                | (1)                           | (2)              | (3)              | (4)              |
|                | ensemble                      | ensemble (crisis)| LSTM             | LSTM (crisis)    |
| **MSE \((h = 1)\)** | 3.3748***                    | 3.6812***        | 1.0367           | 2.7912***        |
|                | (0.0013)                      | (0.0051)         | (0.3042)         | (0.0210)         |
| **MAD \((h = 1)\)**  | 4.6279***                    | 4.1607***        | 0.4797           | 2.2051*          |
|                | (2.18e-05)                    | (0.0024)         | (0.6332)         | (0.0549)         |

|                | one year ahead, \(h = 4\) |                  |                  |                  |
|----------------|-----------------------------|------------------|------------------|------------------|
|                | (1)                         | (2)              | (3)              | (4)              |
|                | ensemble                    | ensemble (crisis)| GRU              | GRU (crisis)     |
| **MSE \((h = 4)\)** | 3.3251***                  | 1.1478           | 1.3952           | 0.5870           |
|                | (0.0015)                    | (0.2806)         | (0.1683)         | (0.5716)         |
| **MAD \((h = 4)\)**  | 4.3702***                  | 0.9839           | 1.6178           | -0.4841          |
|                | (5.32e-05)                  | (0.3508)         | (0.1111)         | (0.6399)         |

The Diebold-Mariano test compares different forecasts in the out-of-sample window, testing the null hypothesis of equality in accuracy. In table 6 the results for the Diebold-Mariano test are illustrated for different loss functions: mean squared error (MSE) and mean absolute deviation (MAD). The test is carried out for the entire out-of-sample window (columns 1 and 3) and considering only the observations characterized by the 2008-09 crisis (columns 2 and 4). Short-term \(h = 1\) and long-term \(h = 4\) forecasts are considered. This clarifies the general differences between the ensemble and neural networks with respect to the DFM-GAS, focusing on the crisis period and evaluating whether the inequalities are more pronounced in the short-term. Columns (1) and (3) compare, respectively, an ensemble and an LSTM with a DFM-GAS one-quarter and four-quarter-ahead forecast for the entire out-of-sample window. Columns (2) and (4) consist in the comparison of an ensemble and an LSTM with a DFM-GAS one-quarter and four-quarter-ahead forecast, considering only crisis periods. Each cell contains DM test statistics as well as p-values in brackets. The ensemble forecast is significantly different and more accurate compared to the DFM-GAS for \(h = 1\). The null hypothesis of forecast equality between ensemble and DFM-GAS is still rejected for \(h = 4\) when all the sample
is considered, but this is not the case for the period of crisis, where the test cannot reject the null (this result could be anticipated by figure 13). Divergences in accuracy are not so evident between single neural networks and DFM-GAS.

E.1 Factor dynamics

With the factor analysis from the DFM we can evaluate how the break given by the crisis could alter the factors dynamics and their co-movements. In the previous section we saw how the predictors markedly change their role in predicting the state of the economy, also noticing that some predictors become significant especially in periods of crisis to anticipate the future GDP release. In order to be more precise for interpreting factors we estimate them with a 0 days horizon, giving nowcasted estimates of the GDP.

![Figure 14: Nowcasted common factors within the sample without Covid.](image)

For the sample without Covid (figure 14) the time-varying mean and the factors well capture the GDP dynamics. By looking at the factor series it is visible that during recession factor volatility increases and the drop in mean of the GDP is matched by a low factors’ estimates. The variation in the economic indicators is well captured by the PCA at a 0 days horizon.
Indeed the DFM works in the nowcast at 0 days horizon without particular divergences between standard DFM (estimated projecting GDP on the factors via OLS) and DFM-GAS: the change in the model’s features allow the factor dynamics to fit the actual GDP. At this stage we can combine interpretation results from Shapley and factors: the higher volatility of factors can be linked to those features of the model whose Shapley values experienced a large variation during the crisis. In other words, with the factors we understand there is a change in how the predictors relate to the estimate of the GDP and with the Shapley we get insights on what predictors are mostly affected by the variation in the economic cycle.