A Neural Network Ensemble Approach for GDP Forecasting [*]

Luigi Longo[†], Massimo Riccaboni[‡], Armando Rungi[§]

March 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 mean with a Generalized Autoregressive Score (DFM-GAS). The analysis is based on a set of predictors encompassing a wide range of variables measured at different frequencies. The forecast exercise is aimed at evaluating the predictive ability of each model's component of the ensemble by considering variations in mean, potentially caused by recessions affecting the economy. Thus, we show how the combination of RNN and DFM-GAS improves forecasts of the US GDP growth rate in the aftermath of the 2008-09 global financial crisis. We find that a neural network ensemble markedly reduces the root mean squared error for the short-term forecast horizon.

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 of the Scuola Normale Superiore (Pisa) conference on "Machine Learning of Dynamic Processes and Time Series Analysis". We want to thank Marco Bee, Fulvio Corsi and Luca Trapin for their valuable comments.

[†] luigi.longo@imtlucca.it, IMT School for Advanced Studies, Lucca, Italy; corresponding author.

[‡] massimo.riccaboni@imtlucca.it, IMT School for Advanced Studies, Lucca, Italy.

[§] armando.rungi@imtlucca.it, IMT School for Advanced Studies, Lucca, Italy.

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 [Bańbura et al., 2013, Buono et al., 2017]. The seminal work by Giannone et al. [2008b] introduced a dynamic factor model (DFM) in order to nowcast the current and future GDP quarters based on a wide set of monthly indicators. On the other hand, 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 several 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]. The most recent application provides a promising implementation of machine learning algorithms within the context of economic predictions such as the nowcast of the US GDP by using a sparse-group LASSO by Babii et al. [2020]. Among different approaches, neural networks are those that most capture the attention of scholars thanks to their natural application within the context of time series[1]. 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 in 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 of periods of extraordinary economic recessions, such as both the 2008-2009 financial crisis and the recent Covid-19 crisis have shown. For this reason, most recent research works have sought to improve on prediction accuracy in times of recession by looking for alternative techniques. Among others, Foroni et al. [2020] used MIDAS regression and attempted 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, on the assumption that they are comparable. However, the necessity of more sophisticated specifications calls for efforts in developing augmented versions of the DFM. 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. [2020] used 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

[1] See for example Kaastra and Boyd [1996] for a neural network design to forecast financial time series.

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 the monitoring and nowcasting performance of the economic activity.

With this research background in mind, we contribute by proposing a combination of a more standard approach with machine learning that enables us to combine the benefits of both. Our ensemble package includes both a time-varying DFM and neural networks to verify how neural networks can 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 considered, especially for the short-term forecast horizon.

We argue that the variation in 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 2008-2009 crisis while implementing a Chow test to evaluate to what extent our model predicts during structural breaks. We compute one-quarter to four-quarter ahead forecasts to evaluate differences between the time-varying DFM and neural networks over different time horizons, as well as assessing when it is suitable to put them together in an ensemble model to forecast along the good and bad turns of the business cycle. As a matter of 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.

Indeed, the aim of the present work is to show the advantages of adopting an ensemble made up of neural network models combined with a time-varying dynamic factor model with a score specification (DFM-GAS) when dynamics in mean induced by potential structural breaks affect the business cycle. We consider 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]. Within this context, we argue that 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[2]. There are both theoretical and empirical reasons why neural network models are more

[2] For an accurate mathematical representation, we suggest Géron [2019].

appropriate for modelling non-linear macroeconomic series. Lapedes and Farber [1987] developed a simulation exercise to show that artificial neural networks accurately predict dynamic nonlinear systems, while Zhang et al. [1998] highlighted 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 [Poggio et al., 2017]. Empirically, neural networks have been used beyond finance to compare with standard models. Loermann and Maas [2019] found 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 [Joseph, 2019][3]. 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 2008-09 crisis and its 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. If the weights structure 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 our findings and discusses promising future developments.

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

[3] For this reason, we also include in the Appendix 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 Joseph [2019].

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 networks we use in this paper.

2.1 Score-Driven Dynamic factor model

This section illustrates the standard dynamic factor model accounting for time-variation in mean with a generalized autoregressive score approach (GAS). 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 in order to satisfy the weak stationarity assumption so that the general DFM specification is given by the following equations:

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

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

In equation (2.1), the monthly indicators are driven by two unobserved stationary stochastic processes that consist in the factor dynamics f_t (through Λ) and the random innovations ε_t. The factors are modelled as a stable VAR(p) process. Both ε_t and u_t are normal and the vector of idiosyncratic components ε_t is unrelated to u_t at all lags, i.e. $E\left[\varepsilon_t u'_{t-k}\right] = 0$ 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 Λ and f_t by Principal Component Analysis (PCA). The estimators $\hat{\Lambda}$ and $\hat{f}_t$ are obtained by solving the following minimization problem:

$$\min_{f_1,\dots,f_T,\Lambda} \frac{1}{NT} \sum_{t=1}^{T} \left(\bar{X}_t - \Lambda f_t\right)' \left(\bar{X}_t - \Lambda f_t\right) \qquad \text{s.t.} \qquad N^{-1}\Lambda'\Lambda = I_r \tag{2.3}$$

The variance-covariance matrix estimator for ε_t is given by:

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

The estimated vector $\hat{f}_t$ is the principal components 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 PCA and Kalman smoothing, a bridge equation is used to estimate parameters and forecast the dependent variable, which is GDP at a quarterly frequency [4]:

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

The h-step ahead forecast is computed as follows:

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

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. [2020]. 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.

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 θ. The information set at time t consists in $\{\alpha_t, F_t\}$ where:

$$\mathcal{F}_t = \left\{Y^{t-1}, F^{t-1}, X^t\right\}, \text{ for } t = 1, \ldots, n$$

We assume y_t to have the following observation density:

[4] Please note that, accordingly, factors are aggregated in order to represent quarterly quantities.

$$y_t \sim p\left(y_t \mid \alpha_t, \mathfrak{F}_t; \theta\right) \tag{2.7}$$

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

$$\alpha_{t+1} = \omega + \sum_{i=1}^{p} A_i s_{t-i+1} + \sum_{j=1}^{q} B_j \alpha_{t-j+1} \tag{2.8}$$

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

$$s_t = S_t \cdot \nabla_t, \quad \nabla_t = \frac{\partial \ln p\left(y_t \mid \alpha_t, \mathfrak{F}_t; \theta\right)}{\partial f_t}, \quad S_t = S\left(t, \alpha_t, \mathfrak{F}_t; \theta\right) \tag{2.9}$$

In this way, the time-varying vector is updated to the next period using the score function ∇_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:

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

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 α_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:

$$\alpha_t = \gamma s_{t-1} + \alpha_{t-1} \tag{2.11}$$

The original process should include a constant and a parameter for the lagged value of α_t. The intuition behind our approach is to estimate a random walk process without an intercept. Indeed, the parameter of α_{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 via maximum likelihood where the time-varying process is identified recursively in the normal log-likelihood equation [5].

[5] A grid-search algorithm is implemented to initialize the maximum likelihood estimation procedure. Results of a simulation exercise are available upon request.

2.2 Recurrent Neural Networks

Artificial neural networks are widely used machine learning algorithms that are mostly employed for prediction purposes. These models take a collection of numerical inputs multiplied with 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 perfectly designed for problems of nowcasting and forecasting, where a series of known inputs is used to predict an output variable that is generally unknown.

Specifically, we use 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 [6]:

$$\begin{aligned}\mathbf{Y}_t &= \phi\left(\mathbf{X}_t \cdot \mathbf{W}_x + \mathbf{Y}_{t-1} \cdot \mathbf{W}_y + \mathbf{b}\right) \\ &= \phi\left(\begin{bmatrix} \mathbf{x}_t & \mathbf{Y}_{t-1} \end{bmatrix} \cdot \mathbf{W} + \mathbf{b}\right) \text{ with } \mathbf{W} = \begin{bmatrix} \mathbf{W}_x \\ \mathbf{w}_y \end{bmatrix}\end{aligned}$$

- $\mathbf{Y_t}$ is a $m \times n_{neurons}$ 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 $n_{neurons}$ is the number of neurons;

- $\mathbf{X_t}$ is a $m \times n_{inputs}$ matrix containing the inputs for all observations;

[6] See Géron [2019] for a more detailed explanation.

- $\mathbf{W_x}$ is a $n_{inputs} \times n_{neurons}$;
- $\mathbf{W_y}$ is a $n_{neurons} \times n_{neurons}$ matrix containing the connection weights for the outputs of the previous time step;
- The weights' matrices $\mathbf{W_x}$ and $\mathbf{W_y}$ are often concatenated into a single weight matrix $\mathbf{W}$ of shape $(n_{inputs} + n_{neurons}) \times n_{neurons}$;
- $\mathbf{b}$ is a vector of size $n_{neurons}$ containing each neuron's bias term.

In this model, $\mathbf{Y_t}$ is a function of $\mathbf{X_t}$ and $\mathbf{Y_{t-1}}$, which is a function of $\mathbf{X_{t-1}}$ and $\mathbf{Y_{t-2}}$ 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. This mechanism works because of the stochastic gradient descent algorithm: the gradient of the loss function gives the direction to move weights onto the next iteration. We use two different types of RNN: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)[7].

LSTM was introduced by Hochreiter and Schmidhuber [1997] and its main feature is the identification of a short-term and a long-term state. This 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[8]:

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

[7] Among the advantages of using RNNs, we can include the possibility to solve the problem of vanishing gradient, see Hochreiter [1998].

[8] 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 huge 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; iii) activation function; iv) optimizer for SGD; v) batch size.

3 Data

We use similar data to that employed in the seminal work by Giannone et al. [2008b] from the Federal Reserve Economic Database (FRED) for a total of 138 monthly predictors[9]. The final 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, we report a bird's eye view of all of these predictors.

[9] Note that Giannone et al. [2008b] lists about 200 macroeconomic variables. We were not able to use some predictors because they are not available from public data sources.

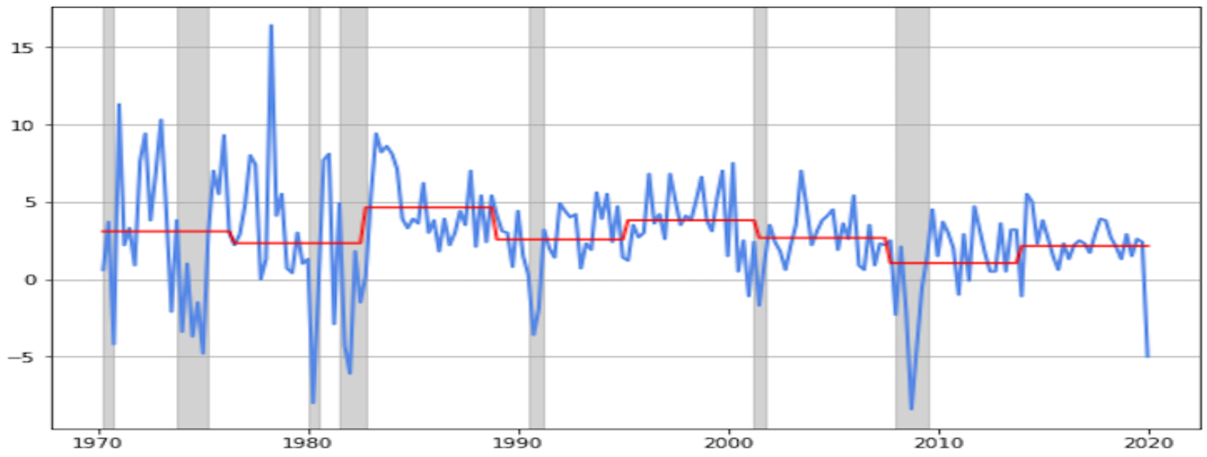

Figure 1: US Gross Domestic Product in the time window 1970Q1-2020Q1. The plotted blue line is the actual GDP value, whereas the red line represents the mean intervals of the process. Gray shaded areas are NBER recession periods.

Figure 1 shows the official US real GDP rate for the period 1970Q1-2020Q1 as well as NBER recession periods in the gray shaded areas[10]. The plotted red line indicates the intervals of the mean, which appears to vary considerably over time, especially in times of recession. From our perspective, the mean-variation is partially explained by the dynamics induced by the downturns along the business cycle.

4 Empirical analysis

In this section, we compare the predictive ability of different models for US GDP growth rates. The h-step prediction considers one quarter up to four quarters ahead forecasts performed by recurrent neural networks, dynamic factor models, and ensembles. For every prediction, the models are compared in terms of forecast errors. The section is organized as follows. We briefly discuss single models, and then we switch to coefficient estimation for the DFM, hyperparameters tuning for RNN, and we discuss how we select the models for the ensemble. We show performances of the models in terms of the root mean squared error (RMSE), and we evaluate how the DFM-GAS accuracy compared with the ensemble/RNN models varies in the out-of-sample window. 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. Finally, a Diebold-Mariano test by Diebold and Mariano [2002] is used to determine whether the ensemble and RNN forecasts are significantly different from that of the DFM-GAS.

[10] Official NBER recession dates found at: https://www.nber.org/research/data/us-business-cycle-expansions-and-contractions.

4.1 Dynamic factor model estimation

The factors are extracted from the panel of monthly indicators via principal component analysis and then 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 via least squares, while in the GAS setting the time-varying parameters are estimated with a maximum likelihood algorithm by means of numerical methods. The MLE algorithm operates in a univariate environment as long as 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. The simple GAS specification is chosen in order to keep the estimation straightforward and to possibly enhance the forecast performance of the model. The number of factor loadings is 2, while the number of shocks to the factors q is chosen through information criteria by Bai and Ng [2007]. The number of factor loadings is the same as that used by Giannone et al. [2008a]. 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 (approximately 30%). 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 well the models are able to capture the data generating process by using only the in-sample data. The same approach is used 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 in order to avoid overfitting. Here, we split the data into training, validation and test set. The training set consists in the first 70% of the observations (1970Q1-2005Q1) while the validation set is the last 20% of 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. The networks are all based on a 3-layer structure: the first layer consists in 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. The optimal number of epochs is chosen in a second step by plotting accuracy in the validation set of the networks with an optimal number of nodes (the plot is shown in the Appendix). Moreover, because of the stochastic nature of neural networks, we set a fixed 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 process. 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 the out-of-sample window. The ensemble process is therefore 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:

$$W^*_{M_1,T+n} = \frac{\frac{1}{MSE^*_{M_1,T+n}}}{\frac{1}{MSE^*_{M_1,T+n}} + \frac{1}{MSE^*_{M_2,T+n}}} \tag{4.1}$$

for $n = 2, 3, ..., N$, T is the last observation of the in-sample set, while $T + N$ is the last observation of the out-of-sample set. The mean squared error is defined as:

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

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. The choice of having two models in the ensemble may enable us to obtain better performances. In order 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:

$$W_{M_1,t} = \frac{\frac{1}{MSE_{M_1,t}}}{\frac{1}{MSE_{M_1,t}} + \frac{1}{MSE_{M_2,t}}} \tag{4.3}$$

Here the MSE uses only the observation at time t, therefore comparing two points: the prediction 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 and 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 both DFM and RNN. Within the DFM framework the dimensionality is controlled by the number of factors, while in the 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. The forecast accuracy is evaluated by the root mean squared error (RMSE) and results are reported for each of the models. In this way the overall performance of the ensemble learning is produced. Instead of merely comparing different models, we can use this exercise to see whether neural networks can help in the prediction of crisis periods for certain forecast horizons. Indeed, in the forecasting window there is an NBER recession period given by the 2008-09 financial crisis that induces a downturn in the GDP. We tested for potential breaks with a Chow test by implementing two unrestricted regressions:

$$y_t = \beta X_t + D_t * \gamma_0 \tag{4.4}$$

$$y_t = \beta X_t + D_t * (\gamma_0 + \gamma' X_t) \tag{4.5}$$

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 4.4 the dummy can be seen as a shift in the intercept, while in equation 4.5 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.

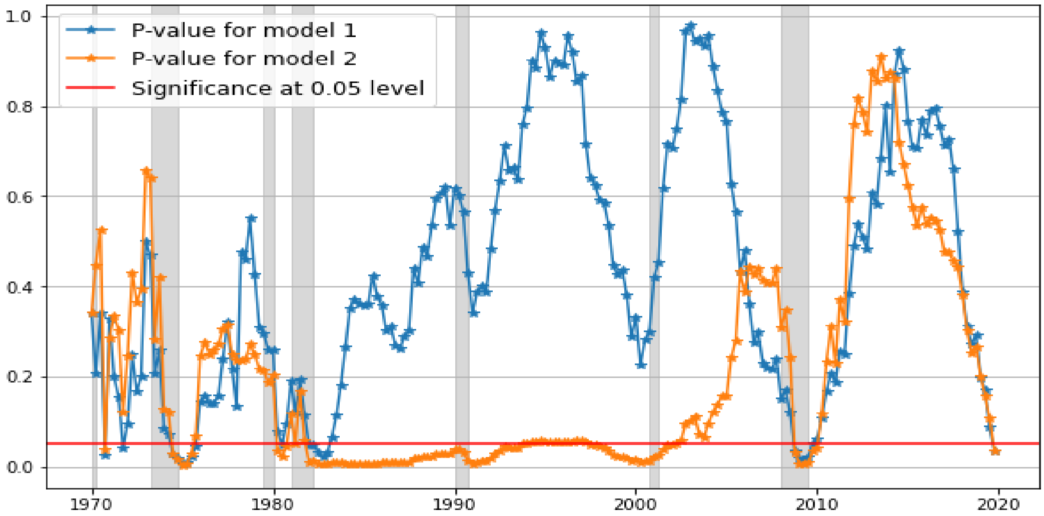

Figure 2: Plotted p-values for F-test of structural breaks. The horizontal red line considers a significance level of 0.05. **Model 1** and **Model 2** represent respectively equations 4.4 and 4.5.

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).

Table 1 shows out-of-sample performances for all the models:

	RMSE			
	$h = 1$	$h = 2$	$h = 3$	$h = 4$
Dynamic Factor Model, DFM	2.9137	2.9941	3.0262	3.0017
t.v. Dynamic Factor Model, DFM-GAS	2.3191	2.5239	2.5787	2.4569
Long short-term Memory, LSTM	2.1585	2.2865	2.3972	2.3520
Gated Recurrent Unit, GRU	2.3196	2.2578	2.6628	2.3394
Ensemble model	2.0130	2.2578	2.3327	2.3094

Table 1: Forecast comparison 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 from factor models and one from neural networks: DFM-GAS & LSTM (h=1,3); DFM-GAS & GRU (h=2,4).

The forecast is reported for one up to four periods ahead. The maximum prediction horizon is therefore one year. Intuitively, a longer forecast horizon implies a higher prediction error by increasing forecast uncertainty: only current and past quarter information is available in order 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 and the forecast performance will be limited[11]. Table 1 reports forecast performances by comparing RMSE for every model. 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 in the first moment. Conversely, there is not a superior neural network, but the LSTM performs better for one quarter and three quarter ahead forecasts, whereas the GRU has a lower RMSE for two quarter ahead forecasts. Apart from $h = 3$, recurrent neural networks always outperform 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 between the factor models because it always outperforms its fixed-parameter counterpart and it will change the RNN depending on the selected time horizon. The ensemble model uses DFM-GAS and LSTM for $h = 1$ and $h = 3$, whereas the GRU is used for $h = 2$ and $h = 4$. The ensemble outperforms all the models for all the forecast horizons (for $h = 2$ it has the same performance as the GRU). Table 2.1 shows that differences between the ensemble and the DFM-GAS decrease with a longer forecast horizon. 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. 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 ($h = 1$).

4.5 Forecast comparison with weights and RMSE difference

In this section we compare forecast performances of the ensemble components, i.e. RNN and DFM-GAS, by using weights constructed as in equation 4.3 for the case of the US GDP one quarter ahead forecast. To this end, we evaluate the 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.

[11] This is generally true in theory: Giannone et al. [2004] found empirical results verifying that it is more difficult to forecast for longer horizons within the context of a macroeconomic system.

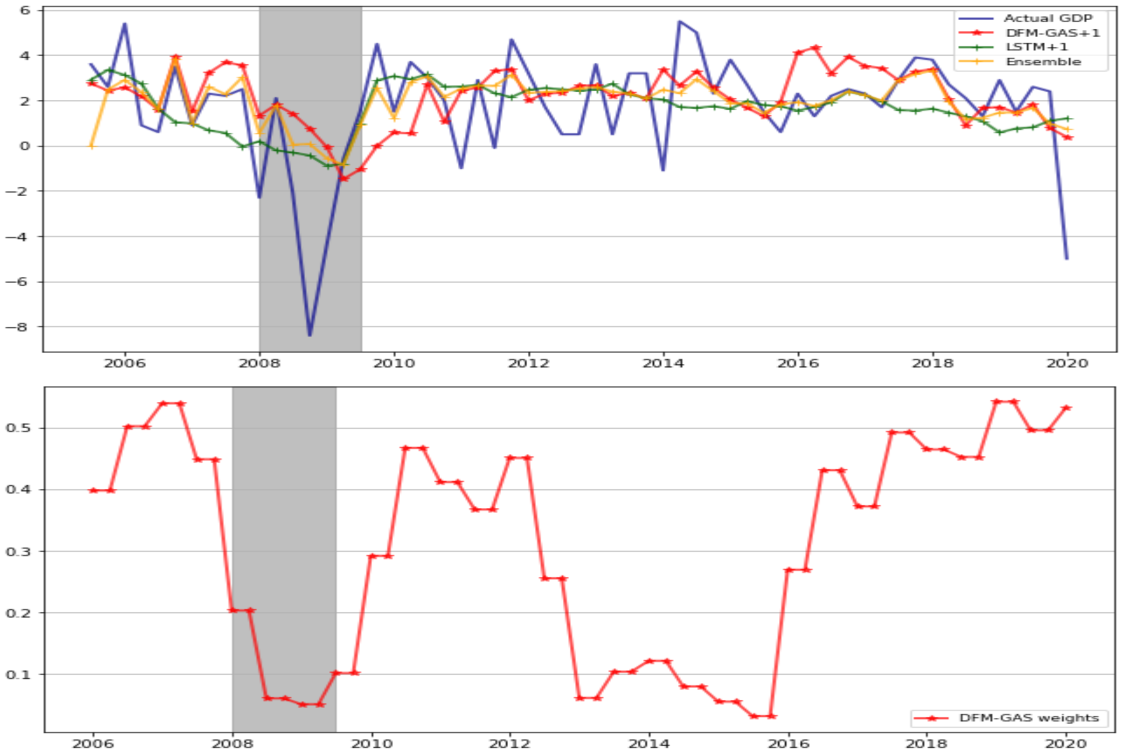

Figure 3: US GDP one quarter ahead forecast with DFM-GAS ($h = 1$), LSTM and ensemble (first panel); DFM-GAS weights constructed as in equation 4.3 comparing the DFM-GAS prediction with the ensemble prediction (second panel). Gray shaded area is the NBER recession period of the 2008-09 crisis.

Figure 3 shows real GDP growth rate with model predictions for the one-quarter-ahead forecast as well as the corresponding weights for the DFM-GAS, constructed as in equation 4.3 and pairwise averaged in order to mitigate noise. In this case the LSTM is the best performing 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. In the time window considered, 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. Weights for the DFM-GAS are plotted in the out-of-sample window to evaluate its contribution compared to that of the ensemble. When weights are low it means that the DFM-GAS is underperforming with respect to the ensemble for that set of observations. During the 2008-09 crisis the weights reach a minimum and it is clear that the ensemble is better at capturing the downturn and recovery of the economy. The weights of the DFM-GAS for every horizon are constructed to compare the prediction of the ensemble model with the actual GDP, as shown in figure 4.

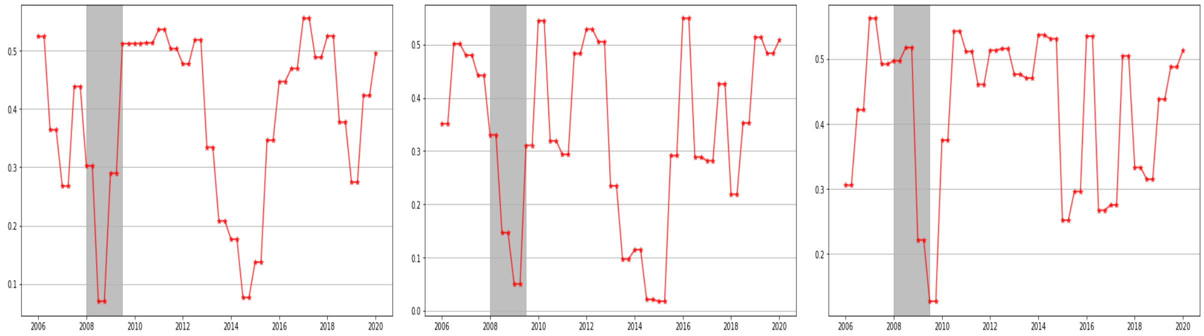

Figure 4: Averaged weights of the DFM-GAS in the ensemble model for $h = 2$ (left), $h = 3$ (middle), $h = 4$ (left).

Figure 4 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. From the weights in the Appendix, it is also clear that sometimes the DFM-GAS outperforms the RNN and therefore it is preferable to use the ensemble rather than simply making predictions based on the RNN.

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 5 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.

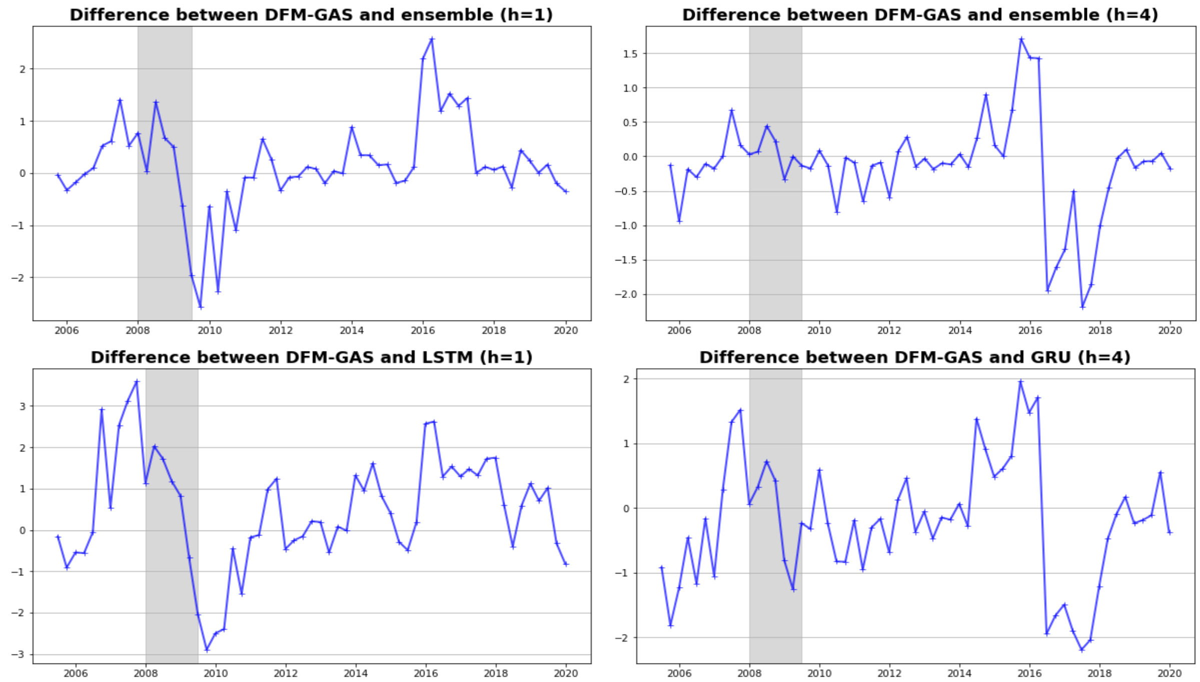

Figure 5: Difference between DFM-GAS and ensemble one quarter ahead prediction (top left) and between DFM-GAS and ensemble four quarters ahead prediction (top right). Difference between DFM-GAS and LSTM one quarter ahead prediction (bottom left) and between DFM-GAS and GRU four quarter ahead prediction (bottom right). LSTM is the best performing RNN one quarter ahead, while GRU is the best performing RNN one year ahead.

In the bottom panel of figure 5, we plot differences between the most accurate neural network [12] 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 $h = 1$.

4.6 Diebold-Mariano test for prediction comparison

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

[12] For one quarter ahead it is the LSTM while for four quarters ahead it is the GRU.

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.

	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)

Table 2: 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 represents DM test statistics with a different loss criterion, while p-values are shown in brackets: $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$.

The Diebold-Mariano test compares different forecasts in the out-of-sample window, testing the null hypothesis of equality in accuracy. In table 2 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 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 5). Divergences in accuracy are not so evident between single neural networks and DFM-GAS.

5 Conclusions

In this paper we contribute to the development of an integrated (ensemble) approach to predict future GDP releases, to take advantage of the availability of big data regarding the current state of the economy. Although they have been broadly discussed in previous literature, no clear consensus has yet been reached on the way to integrate machine learning techniques with traditional methods in order to improve predictions. 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. It is clear that during a recession the GDP variable experiences a drop in value and this causes a break in the mean of the underlying process, which makes it more difficult to predict the new data. Indeed, when the economy experiences a structural break, the data generating process may change because it is affected by shocks that have influences on the dimensions of the economy. The main issue is that during the crisis the analyst is neither aware of the intensity of the shocks nor of its impact on the business cycle, be it permanent or temporary. In this case, inferences about the new data generating process can be drawn immediately after a recovery has started. We argue that every econometric model making strong assumptions about the underlying GDP specification suffers out-of-sample when structural breaks occur, even when accounting for shifts and parameter variation. Within this context, we propose a combination, using both a time-varying DFM as well as 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 marked gap for shorter forecast horizons. The marginal gains of using the ensemble decreases when longer forecast horizons are considered. Indeed, the difference in accuracy between the neural network ensemble and a single DFM-GAS is low for the one year forecast, as confirmed in the RMSE difference analysis and by a Diebold-Mariano test. We believe the reason for the huge gap with short-term forecasts is the fact that out-of-sample structural breaks imply a higher level of complexity, which is better handled with non parametric models such as neural networks. This is not the case for long-term forecasts. Our future work will focus on the application of our approach to assessing how suitable it is in predicting the present Covid-19 economic crisis.

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. *Large Shocks and New Data (August 8, 2020)*, 2020.

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.

Marta Bańbura, 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.

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.

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.

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

James Durbin and Siem Jan Koopman. *Time series analysis by state space methods.* Oxford university press, 2012.

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.

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.

A Joseph. Parametric inference with universal function approximators. 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.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 4765–4774. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf`.

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.

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

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)

ID	Source	FRED code	Description
1	U.S. Bureau of Labor Statistics	MANEMP	All employees, manufacturing
2	U.S. Census Bureau	DGORDER	Manufacturers' New Orders: Durable Goods
3	U.S. Census Bureau	NEWORDER	Manufacturers' New Orders: Nondefense Capital Goods Excluding Aircraft
4	U.S. Census Bureau	AMTMNO	Value of Manufacturers' New Orders for All Manufacturing Industries
5	U.S. Census Bureau	AMTUNO	Value of Manufacturers' New Orders for All Manufacturing Industries with Unfilled Orders
6	U.S. Census Bureau	AMNMNO	Value of Manufacturers' New Orders for Nondurable Goods Industries
7	U.S. Census Bureau	AMTMUO	Value of Manufacturers' Unfilled Orders for All Manufacturing Industries
8	Board of Governors of the Federal Reserve System (US)	DTBSPCKM	Commercial Paper Outstanding
9	National Bureau of Economic Research	M02275USM398NNBR	Public Residential Buildings, Value of New Construction Put in Place for United States
10	Board of Governors of the Federal Reserve System (US)	BOGMBASE	Monetary Base; Total
11	Board of Governors of the Federal Reserve System (US)	TOTRESNS	Total Reserves of Depository Institutions
12	Board of Governors of the Federal Reserve System (US)	BORROW	Total Borrowings of Depository Institutions from the Federal Reserve
13	Board of Governors of the Federal Reserve System (US)	M1SL	M1 Money Stock
14	Board of Governors of the Federal Reserve System (US)	M2SL	M2 Money Stock
15	Organization for Economic Co-operation and Development	MABMM301USM189S	M3 for the United States
16	Board of Governors of the Federal Reserve System (US)	BUSLOANS	Commercial and Industrial Loans, All Commercial Banks
17	Board of Governors of the Federal Reserve System (US)	USGSEC	Treasury and Agency Securities, All Commercial Banks
18	Board of Governors of the Federal Reserve System (US)	INVEST	Securities in Bank Credit, All Commercial Banks
19	Board of Governors of the Federal Reserve System (US)	REALLN	Real Estate Loans, All Commercial Banks
20	Board of Governors of the Federal Reserve System (US)	CONSUMER	Consumer Loans, All Commercial Banks
21	U.S. Bureau of Labor Statistics	UNRATE	Unemployment Rate
22	U.S. Bureau of Labor Statistics	CIVPART	Labor Force Participation Rate
23	U.S. Bureau of Labor Statistics	UEMPLT5	Number Unemployed for Less Than 5 Weeks
24	U.S. Bureau of Labor Statistics	UEMP5TO14	Number Unemployed for 5-14 Weeks
25	U.S. Bureau of Labor Statistics	UEMP15T26	Number Unemployed for 15-26 Weeks
26	U.S. Bureau of Labor Statistics	UEMP15OV	Number Unemployed for 15 Weeks & Over
27	U.S. Bureau of Labor Statistics	PAYEMS	All Employees, Total Nonfarm
28	Automatic Data Processing, Inc.	NPPTTL	Total Nonfarm Private Payroll Employment
29	U.S. Bureau of Labor Statistics	AWHNONAG	Average Weekly Hours of Production and Nonsupervisory Employees, Total Private
30	U.S. Bureau of Labor Statistics	AWHMAN	Average Weekly Hours of Production and Nonsupervisory Employees, Manufacturing
31	U.S. Bureau of Labor Statistics	AWOTMAN	Average Weekly Overtime Hours of Production and Nonsupervisory Employees, Manufacturing
32	U.S. Bureau of Labor Statistics	AHETPI	Average Hourly Earnings of Production and Nonsupervisory Employees, Total Private
33	U.S. Bureau of Labor Statistics	CES2000000003	Average Hourly Earnings of All Employees, Construction
34	U.S. Bureau of Labor Statistics	CES3000000008	Average Hourly Earnings of Production and Nonsupervisory Employees, Manufacturing
35	U.S. Bureau of Labor Statistics	CES4300000008	Average Hourly Earnings of Production and Nonsupervisory Employees, Transportation and Warehousing
36	U.S. Bureau of Labor Statistics	CES4200000008	Average Hourly Earnings of Production and Nonsupervisory Employees, Retail Trade
37	U.S. Bureau of Labor Statistics	CES6000000008	Average Hourly Earnings of Production and Nonsupervisory Employees, Professional and Business Services
38	U.S. Bureau of Labor Statistics	CES6500000008	Average Hourly Earnings of Production and Nonsupervisory Employees, Education and Health Services
39	U.S. Bureau of Labor Statistics	TERMFCLVRNCNS	Average Hourly Earnings of Production and Nonsupervisory Employees, Education and Health Services
40	Board of Governors of the Federal Reserve System (US)	DTCTLVNANM	Average Amount Financed for New Car Loans at Auto Finance Companies
41	U.S. Census Bureau	RSAFS	Advance Retail Sales: Retail and Food Services, Total
42	U.S. Department of the Treasury. Fiscal Service	MTSDS133FMS	Federal Surplus or Deficit [-]
43	National Bureau of Economic Research	M07047USM144NNBR	Merchandise Trade Balance: Excess of Total Exports Over General Imports for United States
44	Board of Governors of the Federal Reserve System (US)	IPB50001N	Industrial Production: Total index
45	Board of Governors of the Federal Reserve System (US)	IPB50001N	Industrial Production: Total index
46	Board of Governors of the Federal Reserve System (US)	IPFPNSS	Industrial Production: Final Products and Nonindustrial Supplies
47	Board of Governors of the Federal Reserve System (US)	IPB50002N	Industrial Production: Final products
48	Board of Governors of the Federal Reserve System (US)	IPCONGD	Industrial Production: Consumer Goods
49	Board of Governors of the Federal Reserve System (US)	IPDCONGD	Industrial Production: Durable Consumer Goods
50	Board of Governors of the Federal Reserve System (US)	IPNCONGD	Industrial Production: Nondurable Consumer Goods
51	Board of Governors of the Federal Reserve System (US)	IPBUSEQ	Industrial Production: Business Equipment
52	Board of Governors of the Federal Reserve System (US)	IPMAT	Industrial Production: Materials
53	Board of Governors of the Federal Reserve System (US)	IPMAN	Industrial Production: Manufacturing (NAICS)
54	Board of Governors of the Federal Reserve System (US)	IPNMAN	Industrial Production: Nondurable Manufacturing (NAICS)
55	Board of Governors of the Federal Reserve System (US)	IPDMAN	Industrial Production: Durable Manufacturing (NAICS)
56	Board of Governors of the Federal Reserve System (US)	IPMINE	Industrial Production: Mining
57	Board of Governors of the Federal Reserve System (US)	IPG2211A2N	Industrial Production: Electric and gas utilities
58	Board of Governors of the Federal Reserve System (US)	IPB50089S	Industrial Production: Energy Materials: Energy, total
59	Board of Governors of the Federal Reserve System (US)	IPX5001ES	Industrial Production: Non-energy, total
60	Board of Governors of the Federal Reserve System (US)	IPG3361T3S	Industrial Production: Durable manufacturing: Motor vehicles and parts
61	Board of Governors of the Federal Reserve System (US)	IPHITEK2S	Industrial Production: Computers, communications equipment, and semiconductors
62	Board of Governors of the Federal Reserve System (US)	TCU	Capacity Utilization: Total Industry
63	Board of Governors of the Federal Reserve System (US)	MCUMFN	Capacity Utilization: Manufacturing (NAICS)
64	Board of Governors of the Federal Reserve System (US)	CAPUTLGMFDS	Capacity Utilization: Durable manufacturing
65	Board of Governors of the Federal Reserve System (US)	CAPUTLGMFNS	Capacity Utilization: Nondurable manufacturing
66	Board of Governors of the Federal Reserve System (US)	CAPUTLG21S	Capacity Utilization: Mining
67	Board of Governors of the Federal Reserve System (US)	CAPUTLHITEK2S	Capacity Utilization: Computers, communications equipment, and semiconductors
68	U.S. Census Bureau	AUTHNOTTSA	New Privately-Owned Housing Units Authorized, but Not Started: Total
69	National Bureau of Economic Research	M12002USM511NNBR	Index of General Business Activity for United States
70	U.S. Bureau of Labor Statistics	PPIFGS	Producer Price Index by Commodity for Finished Goods
71	U.S. Bureau of Labor Statistics	PPILFE	Producer Price Index by Commodity for Finished Goods Less Food and Energy
72	U.S. Bureau of Labor Statistics	PPIFCG	Producer Price Index by Commodity for Finished Consumer Goods
73	U.S. Bureau of Labor Statistics	PPIITM	Producer Price Index by Commodity Intermediate Materials: Supplies and Components
74	U.S. Bureau of Labor Statistics	WPUSOP1000	Producer Price Index by Commodity for Stage of Processing: Crude Materials
75	U.S. Bureau of Labor Statistics	PPIFLF	Producer Price Index by Commodity for Finished Goods Excluding Foods
76	U.S. Bureau of Labor Statistics	PPILFE	Producer Price Index by Commodity for Finished Goods Less Food and Energy
77	U.S. Bureau of Labor Statistics	WPSSOP1600	Producer Price Index by Commodity for Stage of Processing: Crude Materials Less Energy
78	U.S. Bureau of Labor Statistics	WPSSOP1500	Producer Price Index by Commodity for Stage of Processing: Crude Nonfood Materials Less Energy
79	U.S. Bureau of Labor Statistics	CPIAUCSL	Consumer Price Index for All Urban Consumers: All Items in U.S. City Average
80	U.S. Bureau of Labor Statistics	CPIFABSL	Consumer Price Index for All Urban Consumers: Food and Beverages in U.S. City Average
81	U.S. Bureau of Labor Statistics	CPIHOSSL	Consumer Price Index for All Urban Consumers: Housing in U.S. City Average
82	U.S. Bureau of Labor Statistics	CPIAPPSL	Consumer Price Index for All Urban Consumers: Apparel in U.S. City Average
83	U.S. Bureau of Labor Statistics	CPITRNSL	Consumer Price Index for All Urban Consumers: Transportation in U.S. City Average
84	U.S. Bureau of Labor Statistics	CPIMEDSL	Consumer Price Index for All Urban Consumers: Medical Care in U.S. City Average
85	U.S. Bureau of Labor Statistics	CUSR0000SAC	Consumer Price Index for All Urban Consumers: Commodities in U.S. City Average
86	National Bureau of Economic Research	M04186USM350NNBR	Consumer Price Index, Durable Commodities for United States
87	U.S. Bureau of Labor Statistics	CUSR0000SAS	Consumer Price Index for All Urban Consumers: Services in U.S. City Average
88	U.S. Bureau of Labor Statistics	CPIULFSL	Consumer Price Index for All Urban Consumers: All Items Less Food in U.S. City Average
89	U.S. Bureau of Labor Statistics	CPILFESL	Consumer Price Index for All Urban Consumers: All Items Less Food and Energy in U.S. City Average
90	U.S. Bureau of Labor Statistics	CUSR0000SA0L2	Consumer Price Index for All Urban Consumers: All Items Less Shelter in U.S. City Average

91	U.S. Bureau of Labor Statistics	CUSR0000SA0L5	Consumer Price Index for All Urban Consumers: All Items Less Medical Care in U.S. City Average
92	National Bureau of Economic Research	M0602AUSM144SNBR	Manufacturing and Trade Sales, Total for United States
93	U.S. Census Bureau	RSXFS	Advance Retail Sales: Retail (Excluding Food Services)
94	U.S. Census Bureau	TOTBUSSMSA	Total Business Sales
95	U.S. Census Bureau	MRTSSM44X72USS	Retail Sales: Retail and Food Services, Total
96	U.S. Census Bureau	BUSINV	Total Business Inventories
97	U.S. Census Bureau	MNFCTRSMSA	Manufacturers Sales
98	U.S. Census Bureau	MNFCTRIMSA	Manufacturers Inventories
99	U.S. Census Bureau	WHLSLRIMSA	Merchant Wholesalers Inventories
100	Federal Reserve Bank of St. Louis	INVCMRMTSPL	Real Manufacturing and Trade Inventories
101	U.S. Census Bureau	RETAILIMSA	Retailers Inventories
102	U.S. Bureau of Economic Analysis	DSPIC96	Real Disposable Personal Income
103	U.S. Bureau of Economic Analysis	PCE	Personal Consumption Expenditures
104	U.S. Bureau of Economic Analysis	PCEDG	Personal Consumption Expenditures: Durable Goods
105	U.S. Bureau of Economic Analysis	PCEND	Personal Consumption Expenditures: Nondurable Goods
106	U.S. Bureau of Economic Analysis	PCES	Personal Consumption Expenditures: Services
107	U.S. Bureau of Economic Analysis	PCEPI	Personal Consumption Expenditures: Chain-type Price Index
108	U.S. Bureau of Economic Analysis	PCEPILFE	Personal Consumption Expenditures Excluding Food and Energy (Chain-Type Price Index)
109	U.S. Bureau of Economic Analysis	DDURRG3M086SBEA	Personal consumption expenditures: Durable goods (chain-type price index)
110	U.S. Bureau of Economic Analysis	DNDGRG3M086SBEA	Personal consumption expenditures: Nondurable goods (chain-type price index)
111	U.S. Bureau of Economic Analysis	DSERRG3M086SBEA	Personal consumption expenditures: Services (chain-type price index)
112	U.S. Census Bureau	HSN1F	New One Family Houses Sold: United States
113	U.S. Census Bureau	MSACSR	Monthly Supply of Houses in the United States
114	U.S. Census Bureau	HNFSUSNSA	New One Family Houses for Sale in the United States
115	Federal Reserve Bank of Chicago	CFMMI	Chicago Fed Midwest Manufacturing Index
116	University of Michigan	UMCSENT	Chicago Fed Midwest Manufacturing Index
117	U.S. Department of the Treasury	HQMCB30YRP	30-Year High Quality Market (HQM) Corporate Bond Par Yield
118	Board of Governors of the Federal Reserve System (US)	FEDFUNDS	Effective Federal Funds Rate
119	Board of Governors of the Federal Reserve System (US)	TB3MS	3-Month Treasury Bill: Secondary Market Rate
120	Board of Governors of the Federal Reserve System (US)	TB6MS	6-Month Treasury Bill: Secondary Market Rate
121	Board of Governors of the Federal Reserve System (US)	GS1	1-Year Treasury Constant Maturity Rate
122	Board of Governors of the Federal Reserve System (US)	GS5	5-Year Treasury Constant Maturity Rate
123	Board of Governors of the Federal Reserve System (US)	GS7	7-Year Treasury Constant Maturity Rate
124	Board of Governors of the Federal Reserve System (US)	GS10	10-Year Treasury Constant Maturity Rate
125	Moody's	AAA	Moody's Seasoned Aaa Corporate Bond Yield
126	Moody's	BAA	Moody's Seasoned Baa Corporate Bond Yield
127	Bank of England	NEFXRUKA	Nominal Effective Exchange Rate index
128	Organization for Economic Co-operation and Development	CCEUSP01USM651N	Euro to National Currency Spot Exchange Rate for the United States
129	Organization for Economic Co-operation and Development	CCUSSP01JPM650N	US Dollar to National Currency Spot Exchange Rate for Japan
130	Board of Governors of the Federal Reserve System (US)	EXSZUS	Switzerland / U.S. Foreign Exchange Rate
131	Board of Governors of the Federal Reserve System (US)	EXJPUS	Japan / U.S. Foreign Exchange Rate
132	Board of Governors of the Federal Reserve System (US)	EXUSUK	U.S. / U.K. Foreign Exchange Rate
133	Board of Governors of the Federal Reserve System (US)	EXCAUS	Canada / U.S. Foreign Exchange Rate
134	Board of Governors of the Federal Reserve System (US)	BOGZ1FL073164003Q	Interest Rates and Price Indexes; NYSE Composite Index, Level
135	S&P Dow Jones Indices LLC	CSUSHPISA	S&P/Case-Shiller U.S. National Home Price Index
136	Board of Governors of the Federal Reserve System (US)	GS10	10-Year Treasury Constant Maturity Rate
137	Board of Governors of the Federal Reserve System (US)	TB3MS	3-Month Treasury Bill: Secondary Market Rate
138	Organization for Economic Co-operation and Development	CHNXTEXVA01NCMLM	International Trade: Exports: Value (goods): Total for China
139	Organization for Economic Co-operation and Development	EA19XTEXVA01CXMLM	International Trade: Exports: Value (goods): Total for the Euro Area
140	Organization for Economic Co-operation and Development	JPNXTEXVA01CXMLM	International Trade: Exports: Value (goods): Total for Japan
141	U.S. Census Bureau	JPNXTEXVA01CXMLM	Housing Starts: Total: New Privately Owned Housing Units Started

Table 3: Variables for quarterly analysis measured at monthly frequency.

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: **alpha**=0.001 , **beta1**=0.9 , **beta2**=0.999 , **epsilon**=10e-08. 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.

In the figure below is shown a plot of loss (MSE) and accuracy for every number of epochs in the training of the LSTM in the one quarter ahead prediction.

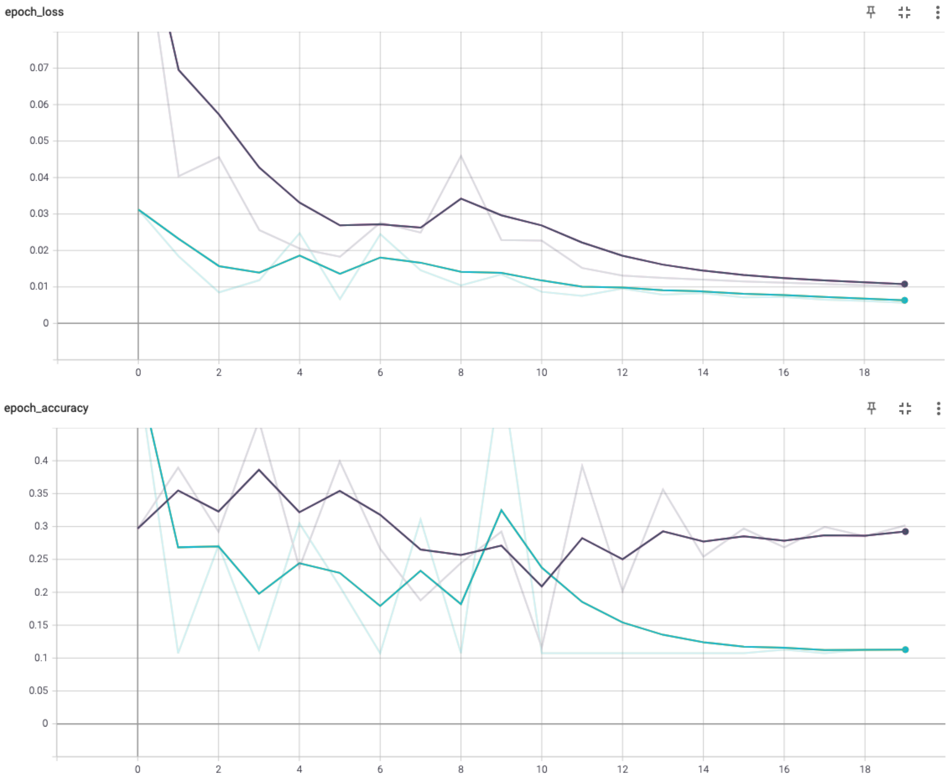

Figure 6: Loss in terms of MSE (y-axis) for different number of epochs (x-axis) during neural network training (panel above). Accuracy (y-axis) for different number of epochs (x-axis) during neural network training (panel below). purple line is the train set while blue line is the validation set.

The optimal number of epochs should minimize loss in both training and validation set, reaching balance between in-sample and out-of-sample properties for the model.

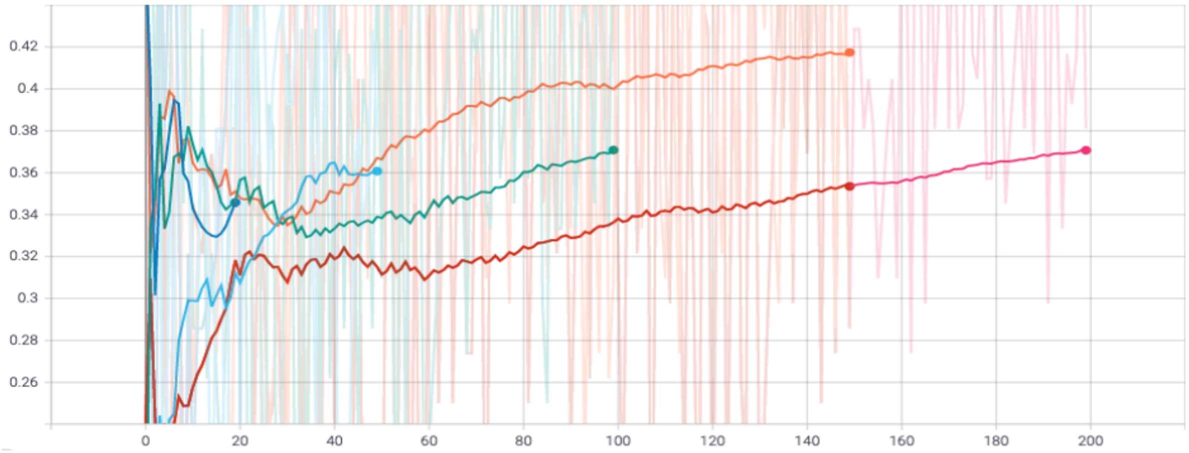

Figure 7: Plotted smoothed accuracy for models with different epochs in the tuning exercise of the LSTM model.

Figure 7 reports the smoothed accuracy for models with different epochs in the tuning exercise of the LSTM model. In this case a structure with 150 epochs gives the highest accuracy in the validation set. The same procedure is implemented to tune the GRU network and used to forecast in the test set for the horizons considered.

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.

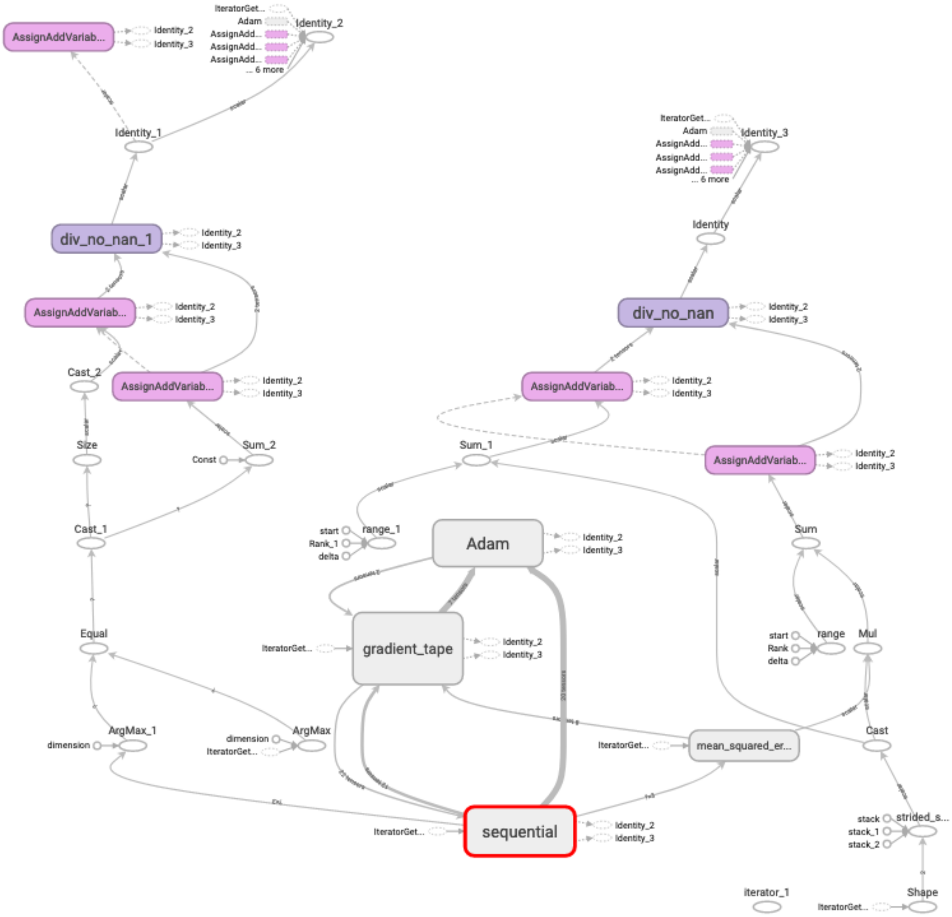

Figure 8: graph at operative level of LSTM one quarter ahead prediction. Data are fed from bottom to the top in the LSTM cells.

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:

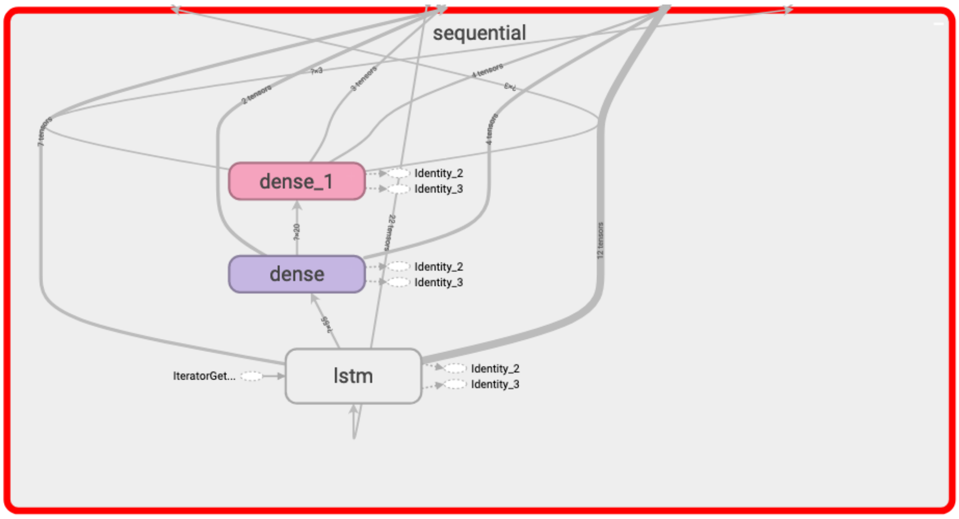

Figure 9: Layers in the LSTM structure.

When using the GRU a similar structure is obtained with the only difference that inputs are fed into GRU cells.

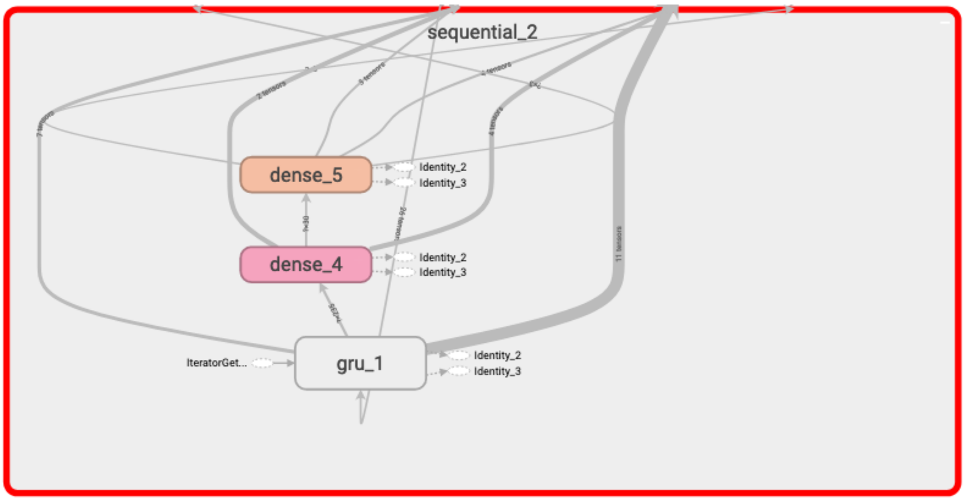

Figure 10: Layers in the GRU structure.

C DFM-GAS weights

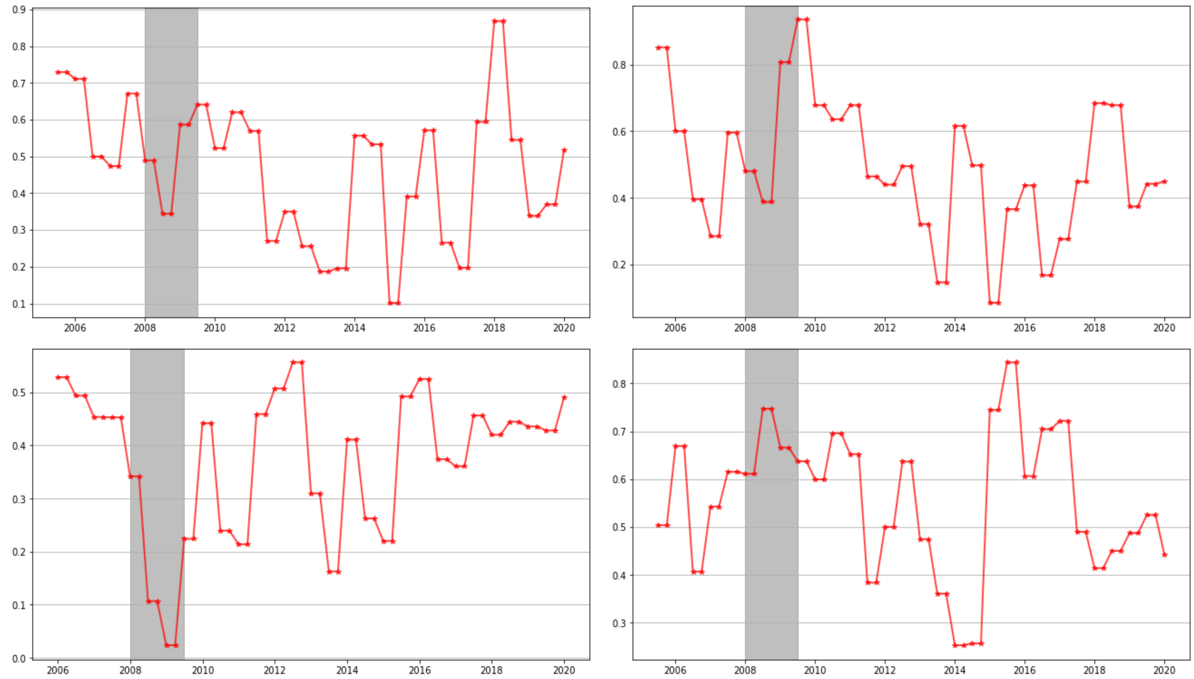

Figure 11: Averaged weights for comparison between the DFM-GAS and the LSTM for $h = 1$ (top-left), $h = 2$ (top-right), $h = 3$ (bottom-left), $h = 4$ (bottom-right).

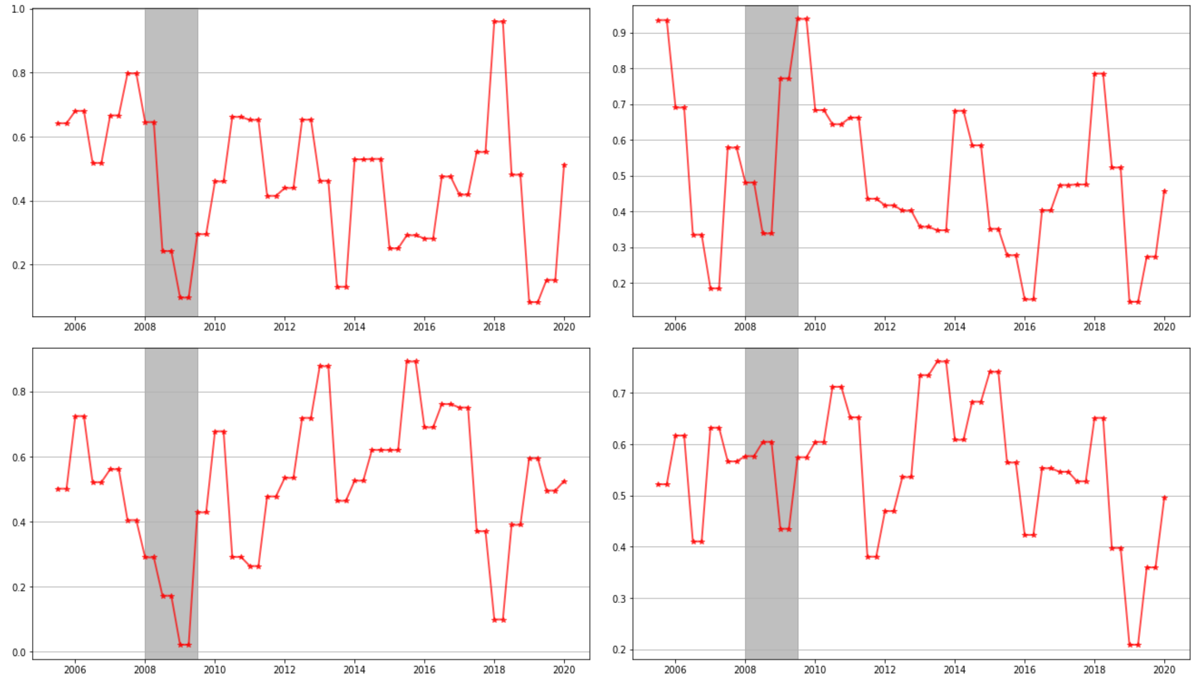

Figure 12: Averaged weights for comparison between the DFM-GAS and the GRU for $h = 1$ (top-left), $h = 2$ (top-right), $h = 3$ (bottom-left), $h = 4$ (bottom-right).

D Shapley Coefficients

In many machine learning applications it is difficult to interpret model's coefficients in order to evaluate inputs' contribution to the final prediction. The reason of such difficulty in interpreting models' parameters arises whenever the models are characterized by high level of complexity. To address the problem we develop an algorithm able to compute shapley values which can be seen as coefficients that quantify the individual inputs' contribution to the total prediction of the model. The algorithm is based on a python package computing SHAP (SHapley Additive exPlanations) from Lundberg and Lee [2017]. SHAP assigns each feature an importance value for a particular prediction. This concept is widely used in game theory and can be interpreted as the average marginal contribution of the input to the prediction, i.e. the extent to which the prediction is affected by including a particular input to the set of total predictors.

Inputs:	**(1)**	**(2)**
Industrial Production: Materials	-0.0234^{***} (0.004)	-0.0051 (0.008)
Commercial Paper Outstanding	-0.0213^{***} (0.013)	-0.0061 (0.007)
Industrial Production: Manufacturing: Durable Goods: Computers, Communications Equipment, and Semiconductors	-0.0210 (0.009)	-0.0035 (0.009)
Average Hourly Earnings of Production and Nonsupervisory Employees, Manufacturing	-0.0210^{***} (0.003)	0.0081 (0.008)
Producer Price Index by Commodity for Stage of Processing: Crude Nonfood Materials Less Energy (DISCONTINUED)	-0.0144^{**} (0.004)	-0.0056 (0.007)
Interest Rates and Price Indexes; NYSE Composite Index, Level	-0.0144^{***} (0.013)	0.0062 (0.008)
International Trade: Exports: Value (goods): Total for the Euro Area	-0.0139 (0.036)	-0.0052 (0.010)
Personal Consumption Expenditures	-0.0130^{***} (0.002)	0.0031 (0.008)
Commercial and Industrial Loans, All Commercial Banks	-0.0130 (0.003)	-0.0102 (0.007)
Consumer Price Index for All Urban Consumers: All Items Less Medical Care in U.S. City Average	-0.0112^{***} (0.002)	-0.0067 (0.008)
Average Hourly Earnings of Production and Nonsupervisory Employees, Education and Health Services	-0.0109^{***} (0.004)	-0.0024 (0.008)

$^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$

Manufacturers' New Orders: Durable Goods	-0.0104	-0.0007
	(0.012)	(0.007)
Average Hourly Earnings of Production and Nonsupervisory Employees, Professional and Business Services	-0.0104^{**}	0.0106
	(0.005)	(0.008)
Consumer Price Index for All Urban Consumers: Housing in U.S. City Average	0.0109^{***}	-0.0095
	(0.003)	(0.008)
Manufacturers Sales	0.0133^{***}	0.0001
	(0.012)	(0.010)
Personal Consumption Expenditures: Durable Goods	0.0134^{***}	-0.0066
	(0.008)	(0.008)
Industrial Production: Equipment: Business Equipment	0.0134^{**}	$-3.7532e-08$
	(0.010)	(0.009)
Consumer Price Index for All Urban Consumers: Commodities in U.S. City Average	0.0137^{***}	-0.0129
	(0.003)	(0.007)
Total Business Sales	0.0157^{*}	-0.0011
	(0.009)	(0.010)
Real Estate Loans, All Commercial Banks	0.0190^{***}	-0.0070
	(0.005)	(0.010)
S&P/Case Shiller U.S. National Home Price Index	0.0190	0.0084
	(0.011)	(0.008)
Manufacturers' New Orders: Manufacturing with Unfilled Orders	0.0193^{***}.	-0.0023
	(0.014)	(0.008)
Average Hourly Earnings of All Employees, Construction	0.0219^{***}	-0.0069
	(0.003)	(0.011)

$^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$

Table 4: Column (1): Shapley coefficients of the 23 most influential variables for LSTM one quarter ahead prediction. Column (2): Shapley coefficients for GRU one quarter ahead prediction. Critical values for inference are computed by bootstrap.

In table 4 Shapley coefficients are shown for the 23 most relevant macroeconomic indicators used for the one quarter ahead prediction of the LSTM. Relevance of the indicators is given by the size of the computed coefficient. Inference is based on confidence intervals constructed by bootstrapping: Shapley values are computed many times and the random source is given by the stochastic assignment of initial weights to the Neural Networks training. The most influential indicators come from a huge variety of macroeconomic sectors including employment, industrial production, real estates and financial index. Shapley coefficients for the most influential variables of the LSTM are computed also for a GRU neural network one quarter ahead prediction: significance is no longer provided and the variables relevant for an LSTM does not seem to be relevant also for a GRU.