Reservoir Computing for Macroeconomic Forecasting with Mixed Frequency Data

Giovanni Ballarin[1], Petros Dellaportas[2,3], Lyudmila Grigoryeva[4,5],
Marcel Hirt[8], Sophie van Huellen[6,7], Juan-Pablo Ortega[8]

November 2, 2022

Abstract

Macroeconomic forecasting has recently started embracing techniques that can deal with large-scale datasets and series with unequal release periods. The aim is to exploit the information contained in heterogeneous data sampled at different frequencies to improve forecasting exercises. Currently, MIxed-DAta Sampling (MIDAS) and Dynamic Factor Models (DFM) are the two main state-of-the-art approaches that allow modeling series with non-homogeneous frequencies. We introduce a new framework called the Multi-Frequency Echo State Network (MFESN), which originates from a relatively novel machine learning paradigm called reservoir computing (RC). Echo State Networks are recurrent neural networks with random weights and trainable readout. They are formulated as nonlinear state-space systems with random state coefficients where only the observation map is subject to estimation. This feature makes the estimation of MFESNs considerably more efficient than DFMs. In addition, the MFESN modeling framework allows to incorporate many series, as opposed to MIDAS models, which are prone to the curse of dimensionality. Our discussion encompasses hyperparameter tuning, penalization, and nonlinear multistep forecast computation. In passing, a new DFM aggregation scheme with Almon exponential structure is also presented, bridging MIDAS and dynamic factor models. All methods are compared in extensive multistep forecasting exercises targeting US GDP growth. We find that our ESN models achieve comparable or better performance than MIDAS and DFMs at a much lower computational cost.

Key Words: Reservoir Computing, Echo State Networks, forecasting, US output growth, GDP, mixed-frequency data, time series, Multi-Frequency Echo State Network, MIDAS, DFM

JEL: C53, C45, E17

[1] Department of Economics, University of Mannheim, L7, 3-5, Mannheim, 68131, Germany. Giovanni.Ballarin@gess.uni-mannheim.de

[2] Department of Statistical Science, UCL, Gower Str., London WC1E 6BT, UK. P.Dellaportas@ucl.ac.uk

[3] Department of Statistics, Athens University of Economics and Business, 10434 Athens, Greece. Petros@aueb.gr

[4] Faculty of Mathematics and Statistics, University of St. Gallen, Badanstrasse 6, CH-9000 St. Gallen, Switzerland. Lyudmila.Grigoryeva@unisg.ch

[5] Honorary Associate Professor, Department of Statistics, University of Warwick, Coventry CV4 7AL, UK. Lyudmila.Grigoryeva@warwick.ac.uk

[6] Global Development Institute (GDI), University of Manchester, Manchester M13 9PL, UK. Sophie.vanHuellen@manchester.ac.uk

[7] Department of Economics, SOAS University of London, WC1H 0XG, London, UK. Sv8@soas.ac.uk

[8] Division of Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, Singapore 637371. MarcelAndre.Hirt@ntu.edu.sg (some of the work done while at UCL), Juan-Pablo.Ortega@ntu.edu.sg

Contents

1 Introduction

The availability of timely and accurate forecasts of key macroeconomic variables is of crucial importance to economic policy makers, businesses, and the banking sector alike. Key macroeconomic variables, such as GDP growth, become available at low frequency with a considerable time lag and are subject to various rounds of revisions after their release. This is particularly problematic in a fast-changing and uncertain economic environment, as experienced during the Great Recesion of 2007-2008 (Hindrayanto et al. (2016)) and the recent pandemic (Buell et al. (2021), Huber et al. (2021)). However, a large number of the potentially predictive financial market (and other macroeconomic) indicators are available at a daily or even higher frequency (Andreou et al. (2013)). The desire to utilize such high-frequency data for macroeconomic forecasting has led to the exploration of techniques that can deal with large-scale datasets and series with unequal release periods (see Borio (2011, 2013), Morley (2015); we also refer the reader to Fuleky (2020) for more details regarding high-dimensional data and to Armesto et al. (2010) and Bańbura et al. (2013) for a review on mixed-frequency data).

We contribute to the existing literature by proposing a new macroeconomic forecasting framework that utilizes high-dimensional and mixed-frequency input data, the Multi-Frequency Echo State Network (MFESN). The MFESN originates from a machine learning paradigm called Reservoir Computing (RC). RC is a family of learning models that take advantage of the information processing capabilities of complex dynamical systems (see Maass et al. (2002), Legenstein and Maass (2007), Crutchfield et al. (2010), and Lukoševičius and Jaeger (2009), Tanaka et al. (2019) for reviews). Generally speaking, RC is a versatile class of recurrent neural network (RNN) models (see Salehinejad et al. (2017) for a detailed survey). Although conventional RNNs are well-suited for handling sequence data and dynamic problems, estimating their weights during the training phase is well-known to be inherently difficult (Pascanu et al. (2013), Doya (1992)). Reservoir networks stand out due to the fact that their inner weights can be *randomly generated* and *fixed*, and only the output (readout) map is subject to supervised training. Echo State Network (ESN) is one of the most popular instances of RC models with provable universality, generalization properties (see Grigoryeva and Ortega (2018a,b, 2019), Gonon et al. (2020a, 2022), Gonon and Ortega (2021), and references therein for more details), and excellent performance in forecasting and classification. While RNNs have been adopted for macroeconomic forecasting in a few instances (see, for example Paranhos (2021)), to the best of our knowledge, we are the first to explore easily-trainable reservoir models in this context. Our main contribution is three-fold. First, inspired by the remarkable empirical success of ESNs in prediction tasks, we propose the so-called Multi-Frequency Echo State Network (MFESN) framework, which allows multistep forecasting of the target variable at lower or the same frequencies as those of the input series. Second, we introduce two different approaches to predicting within the MFESN framework, namely *Single-Reservoir MFESN* (S-MFESN) and *Multi-Reservoir MFESN* (M-MFESN). S-MFESN is determined by modifying the ESN architecture to accommodate input and target variables of mixed frequencies. In M-MFESN, several Echo State Networks are adopted to handle input time series, each ESN corresponding to a group of input variables quoted at one given frequency. Finally, our third contribution consists in an extensive empirical comparative analysis of the forecasting capability of the proposed approaches in a concrete task of predicting the quarterly U.S. output growth. We inspect the forecasting capabilities of the MFESN framework compared to two well-established benchmarks widely used in the macroeconomic literature and among practitioners and show its empirical superiority in several thoroughly conducted forecasting exercises. Moreover, as a bi-product, we propose a new data aggregation scheme that allows bridging these two standard forecasting approaches, which is not available in the literature.

In our empirical study, we evaluate the multistep forecasting performance of the MFESN framework targeting quarterly U.S. output growth (Gross Domestic Product (GDP) growth) and utilizing a small- and medium-sized set of monthly and daily financial and macroeconomic variables. We compare the MFESN approach against two state-of-the-art methods, MIDAS and DFM, known for their ability to incorporate data of heterogeneous frequencies and utilize high-dimensional data inputs. The MIxed DAta Sampling (MIDAS) model developed in Ghysels et al. (2004, 2007) has been adopted widely for macroeconomic forecasting with mixed-frequency data (see for instance Clements and Galvão (2008, 2009), Ghysels and Wright (2009), Francis et al. (2011), Monteforte and Moretti (2012), Galvão and

Marcellino (2010), Galvão (2013), Andreou et al. (2013), Ghysels (2016), Jardet and Meunier (2020)). However, MIDAS is prone to curse-of-dimensionality problems and performs poorly when the set of predictors is of even moderate size (Clements and Galvão (2009), Kostrov (2021)) due to the optimization related issues. Recently, some attempts have been made in the literature to overcome these issues by employing variable selection techniques under some additional assumptions. For instance, Babii et al. (2022) proposes the MIDAS projection approach, which is more amenable to high-dimensional data environments under the assumption of sparsity. Even with these improvements, practical high-dimensional implementations of MIDAS remain challenging. This is in part caused by the ragged edges of the "raw" macroeconomic data, incomplete observations, and uneven sampling frequencies. The relative inflexibility of MIDAS regression lag specifications makes integrating daily and weekly data at true calendar frequencies (i.e. without interpolation or aggregation) very complex. State-space models effectively mitigate these issues. A strong state-of-the-art state-space competitor for our MFESN framework is the Dynamic Factor Model (DFM), which has been first introduced in Geweke (1977) and Sargent et al. (1977). DFMs have become the standard workhorse for macroeconomic nowcasting and prediction (for more details, we refer the reader to Stock and Watson (1996, 2002, 2016), Giannone et al. (2008), Bańbura and Rünstler (2011), Chauvet et al. (2015), Hindrayanto et al. (2016)). Conventional DFMs for data of multiple sampling frequencies are linear state-space models with a latent low-frequency process of interest and high-dimensional input time series. Although their linear structure lends itself to inference with likelihood-based methods and Kalman filtering, using DFMs in the high-dimensional setting is limited by the associated computational effort. For Gaussian state-space models, some of these issues are proposed to be handled with a more compact matrix representation as in Delle Monache and Petrella (2019). Still, in the particular settings of nowcasting and forecasting of GDP growth, the computational complexity is one of the main reasons why DFMs are rarely used with daily input series (see Bańbura et al. (2013) for a detailed review). We address these numerical difficulties using novel Python libraries for auto differentiation and using GPUs for parallel computing[1], which allow the estimation of DFMs even in instances of high-frequency input observations. Further, to adapt the DFM to mixed frequency tasks, we propose a new DFM aggregation scheme with Almon polynomial structure that bridges MIDAS and the DFM for our forecasting comparison. To our knowledge, we are the first to present this aggregation scheme which reduces the number of parameters subject to estimation.

In order to carry out a fair comparison of our MFESN framework with the state-of-the-art MIDAS and DFM models, we designed two model evaluation settings that differ regarding whether the financial crisis of 2008-2009 is included in the estimation period or not. In the first forecasting setting, all the competing models are estimated using the data from January 1st, 1990, until December 31st, 2007. Their performance in the forecasting into and after the financial crisis period is assessed. In the second evaluation setting, fitting is done with data largely encompassing the crisis period, again from January 1st, 1990 but now up to December 31st, 2011. In both cases, the forecasting (testing) period comprises the time period up to the COVID-19 pandemic events, namely the fourth quarter of 2019. Along with the two state-of-the-art DFM and MIDAS models, we use the unconditional mean of the sample as a baseline benchmark against the reservoir models. We find that our ESN-inspired models attain comparable or much better performance than DFMs at a much lower computational cost, even for a relatively long forecasting horizon of four quarters. Additionally, ESNs do not suffer from curse-of-dimensionality problems, which are known to be pervasive for MIDAS models and hence consistently outperform them in a number of forecasting exercises.

The remainder of the paper is structured as follows: Section 2 introduces the notation and terminology used throughout the paper. Section 3 presents the MFESN framework and discusses its implementations, including model estimation, hyperparameter tuning, penalization, and nonlinear multistep forecast evaluation. In this section, we dedicate our attention to proposing single-reservoir and multi-reservoir MFESN architectures and to spelling out their defining features. Section 4 discusses the specification of the benchmark models and introduces the new Almon polynomial aggregation scheme for the DFMs. Section 5 contains the empirical study, it describes the datasets used for the two model evalu-

[1] Our codes, forecasts, and simulations are available in the GitHub repository `RCEconModelling/Reservoir-Computing-for-Macroeconomic-Modelling`.

ation experiment designs and comparatively assesses one-step and multistep forecasting results. Section 6 concludes with the forecasting comparison and discusses future research avenues and applications.

2 Notation and Preliminaries

This section introduces the notation used throughout the paper. It also presents what we call *temporal notation*, which allows us to write multi-frequency models consistently and unambiguously, even when an arbitrary number of sampling frequencies is considered. We also introduce definitions for nowcasting and low- and high-frequency forecasting schemes.

2.1 Notation

We use the symbol $\mathbb{N}$ (respectively, $\mathbb{N}^+$) to denote the set of natural numbers with the zero element included (respectively, excluded). $\mathbb{Z}$ denotes the set of all integers, and $\mathbb{Z}_-$ (respectively, $\mathbb{Z}_+$) stands for the set of the negative (respectively, positive) integers with the zero element included. We abbreviate the set $[n] = \{1, \ldots, n\}$, with $n \in \mathbb{N}^+$.

Vector notation: A column vector is denoted by a bold lower case symbol like $\boldsymbol{r}$ and $\boldsymbol{r}^\top$ indicates its transpose. Given a vector $\boldsymbol{v} \in \mathbb{R}^n$, we denote its entries by v_i, with $i \in \{1, \ldots, n\}$; we also write $\boldsymbol{v} = (v_i)_{i \in \{1,\ldots,n\}}$. The symbols $\boldsymbol{i}_n, \boldsymbol{0}_n \in \mathbb{R}^n$ stand for the vectors of length n consisting of ones and of zeros, respectively. Additionally, given $n \in \mathbb{N}^+$, $\boldsymbol{e}_n^{(i)} \in \mathbb{R}^n$, $i \in \{1, \ldots, n\}$ denotes the canonical unit vector of length n determined by $\boldsymbol{e}_n^{(i)} = (\delta_{ij})_{j \in \{1,\ldots,n\}}$.

Matrix notation: We denote by $\mathbb{M}_{n,m}$ the space of real $n \times m$ matrices with $m, n \in \mathbb{N}^+$. When $n = m$, we use the symbols $\mathbb{M}_n$ and $\mathbb{D}_n$ to refer to the space of square and diagonal matrices of order n, respectively. Given a matrix $A \in \mathbb{M}_{n,m}$, we denote its components by A_{ij} and we write $A = (A_{ij})$, with $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots m\}$. The symbol $\mathbb{I}_n \in \mathbb{D}_n$ denotes the identity matrix and the symbol $\mathbb{O}_n$ stands for the zero matrix of dimension n.

Input and target stochastic processes: We fix a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ on which all random variables are defined. The input and target signals are modeled by discrete-time stochastic processes $\boldsymbol{z} = (\boldsymbol{z}_t)_{t \in \mathbb{Z}}$ and $\boldsymbol{y} = (\boldsymbol{y}_t)_{t \in \mathbb{Z}}$ taking values in $\mathbb{R}^K$ and $\mathbb{R}^J$, respectively. Moreover, we write $\boldsymbol{z}(\omega) = (\boldsymbol{z}_t(\omega))_{t \in \mathbb{Z}}$ and $\boldsymbol{y}(\omega) = (\boldsymbol{y}_t(\omega))_{t \in \mathbb{Z}}$ for each outcome $\omega \in \Omega$ to denote the realizations or sample paths of $\boldsymbol{z}$ and $\boldsymbol{y}$, respectively. Since $\boldsymbol{z}$ can be seen as a random sequence in $\mathbb{R}^K$, we write interchangeably $\boldsymbol{z} : \mathbb{Z} \times \Omega \longrightarrow \mathbb{R}^K$ and $\boldsymbol{z} : \Omega \longrightarrow (\mathbb{R}^K)^{\mathbb{Z}}$. The same applies to the analogous assignments involving $\boldsymbol{y}$.

Temporal notation: Let $(u_t)_{t \in I}$, $u_t \in \mathbb{R}$ be a (scalar) time series with I some index set (in this paper it will always be discrete). Time series $(u_t)_{t \in I}$ will be denoted just as (u_t) when the index set I is specified by the context. We write $u_{s_1:s_2} = (u_t)_{t \in \{s_1,\ldots,s_2\}}$ for integers $s_1 < s_2$ and time series (u_t). To define the concept of the sampling frequency, we must introduce an additional series, call it $(z_s)_{s \in J}$. The time index J is not the same as I. We assume that u_t is sampled at the coarsest rate; equivalently, it has the *lowest* sampling frequency, which we call in what follows the *reference frequency.* In practice, this means that in the same window of time, u_t will be observed at most as frequently as z_s. The case when the sampling frequency of z_s is strictly higher than that of u_t is of primary interest.

We assume that all sampling happens at instants which are evenly spaced in time. Series other than the reference one and with higher sampling frequencies are given an additional time index, the *tempo index*, written $t, *|\kappa$, where κ is the *frequency multiplier.* Our tempo notation assumes that low- and high-frequency series are sampled with temporal *alignment*: this means that the reference time index t and the tempo index $*|\kappa$ have the following properties.

Definition 2.1 *A reference time index $t \in \mathbb{Z}_+$ and a tempo index $*|\kappa$ for a given high-frequency $\kappa \in \mathbb{N}^+$ are such that the following relations hold*

(i) $t, 0|\kappa \equiv t$

(ii) $t, \kappa|\kappa \equiv t + 1$

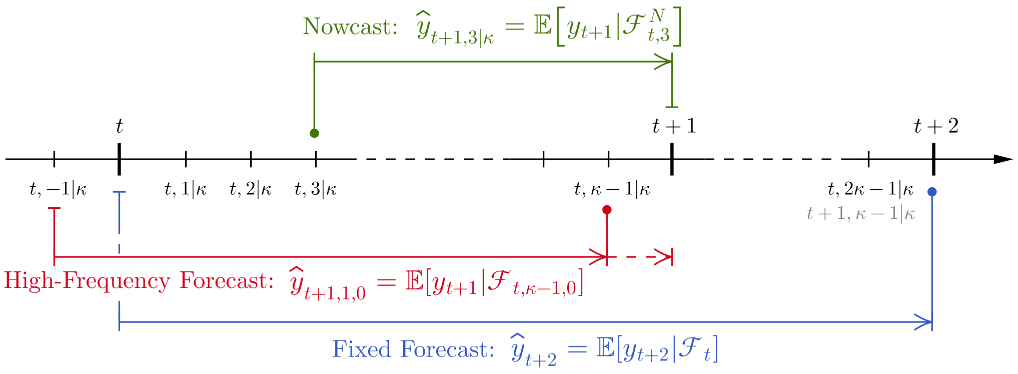

Nowcast: $\widehat{y}_{t+1,3|\kappa} = \mathbb{E}\left[y_{t+1}|\mathcal{F}^N_{t,3}\right]$
t
$t+1$
$t+2$
$t,-1|\kappa$
$t,1|\kappa$
$t,2|\kappa$
$t,3|\kappa$
$t,\kappa-1|\kappa$
$t,2\kappa-1|\kappa$
$t+1,\kappa-1|\kappa$
High-Frequency Forecast: $\widehat{y}_{t+1,1,0} = \mathbb{E}[y_{t+1}|\mathcal{F}_{t,\kappa-1,0}]$
Fixed Forecast: $\widehat{y}_{t+2} = \mathbb{E}[y_{t+2}|\mathcal{F}_t]$

Figure 1: Diagram illustrating the fixed forecasting, high-frequency forecasting, and nowcasting schemes in tempo notation. Arrows point to time indices of the forecast target, while dots indicate the high-frequency time placeholder for the constructed high-frequency forecasts.

(iii) $t, s|\kappa \equiv t + \lfloor s/\kappa \rfloor, (s \bmod \kappa)|\kappa \quad \text{for } \forall s \in \mathbb{N}$

(iv) $t, -s|\kappa \equiv (t-1) - \lfloor s/\kappa \rfloor, \kappa - (s \bmod \kappa)|\kappa \quad \text{for } \forall s \in \mathbb{N},$

where mod *is the modulo operation and for any $x \in \mathbb{R}$ the floor operator $\lfloor x \rfloor$ outputs the greatest $z \in \mathbb{N}$ such that $z \leq x$.*

We now give an example to clarify the use of the tempo notation.

Example 2.2 (Macroeconomic Mixed Frequency Data) Let (y_t) be the time series of quarterly sampled GDP growth and let (u_r) be the time series of industrial production (IP) available at monthly frequency. Notice that we use two different time indexes, t and r. We assume that both series are observed at the *end* of the relevant time period: GDP is released at the end of each quarter, and IP is released at the end of each month. Additionally, we assume that GDP is observed at the end of the last month of the quarter which coincides with the release of monthly IP. Formally, we may then write $t \equiv \lfloor r/3 \rfloor$ for all t and r, since there are exactly 3 months in each quarter. In our tempo notation we proceed in a reverse fashion and we instead anchor time to the lowest frequency index. The frequency multiplier is $\kappa = 3$, therefore $r \equiv t, s|3$ with $s \in \{1, 2, 3\}$.

Since in the tempo notation we can exchange "frequency" and "frequency multiplier", we will make no distinction between the two terms in what follows.

2.2 Nowcasting, Forecasting and Multicasting

In order to clarify the design of the forecasting experiments conducted in this paper, we now present the different types of prediction that we shall tackle.

For the sake of simplicity, let t denote time in the reference frequency of (y_t) and suppose that only a single regressor (z_r) of frequency κ is included in the forecasting model along with an autoregressive term. The notation can be readily extended to include multiple regressors. In this section, let $h \geq 0$ be a *low-frequency* prediction horizon for the low-frequency variable of interest: h always stands for the forecasting horizon counted from the last available low-frequency observation of the target series (y_t). Let $\ell \geq 0$ be a *high-frequency* horizon with respect to frequency κ. We shall consider the following different forecasting exercises, illustrated in Figure 1:

- **Fixed forecasting:** We call a forecast *fixed* when predictions for the target variable are constructed only at the end of the low-frequency periods, namely, $h \in \mathbb{N}$. We use this terminology because when constructing this forecast, neither its information set changes with different choices of h, nor it is updated by the information coming from the high-frequency regressors. The information set which is used at the time of h-steps ahead fixed forecasting at t is the σ-algebra defined as

$$\mathcal{F}_t = \sigma\left(\left\{y_t, y_{t-1}, y_{t-2}, \ldots, z_{t,0|\kappa}, z_{t,-1|\kappa}, z_{t,-2|\kappa}, \ldots\right\}\right) \tag{2.1}$$

and the h-steps ahead fixed forecast at t, when using the mean square error as a loss, is hence provided by the conditional expectation

$$\widehat{y}_{t+h} = \mathbb{E}\left[y_{t+h}|\mathcal{F}_t\right] \equiv \mathbb{E}\left[y_{t,0|\kappa}|\mathcal{F}_t\right]. \tag{2.2}$$

- **Nowcasting:** We call *nowcasting* the setup in which one constructs a high-frequency proxy for a yet-unobserved target which will be available at the end of the *current* low-frequency period. As such, we construct a nowcast only for horizons $0 < \ell \leq \kappa - 1$; notice that $\ell = \kappa$ yields a contemporaneous regression at $t+1$, while $\ell = 0$ falls into the category of fixed forecasting considered above, hence both these two cases are excluded. The σ-algebras that are used in order to construct nowcasts $\widehat{y}_{t+1,\ell|\kappa}$ are given by

$$\mathcal{F}^N_{t,\ell} = \sigma\left(\left\{y_t, y_{t-1}, y_{t-2}, \ldots, z_{t,\ell|\kappa}, z_{t,\ell-1|\kappa}, z_{t,\ell-2|\kappa}, \ldots\right\}\right).$$

  The ℓ-steps nowcast for the high-frequency proxy constructed at moments $t, \ell|\kappa$ of the current period for the low-frequency variable which becomes available at $t+1, 0|\kappa \equiv t+1$ is provided by the conditional expectation

$$\widehat{y}_{t+1,\ell|\kappa} = \mathbb{E}\left[y_{t+1}|\mathcal{F}^N_{t,\ell}\right].$$

- **High-frequency forecasting:** Unlike the fixed forecasting scheme where one makes forecasts of the target variable in its low (reference) frequency, one may also use high-frequency regressors to produce additional forecasts. For example, in the case of a target released at the end of each year and having monthly quoted covariates, the fixed forecasting scheme would correspond to constructing forecasts for the yearly sampled variable of interest always at the end of the last month of the year (December). At the same time, with all the information collected up to the end of December, there are other possibilities to construct forecasts. In particular, the forecaster could consider placing herself at the end of any other month of the year instead and construct predictions for the monthly proxy of the yearly variable for future years.

  It is obvious that in this scheme, one often artificially *reduces* the information set. Although not all the available information is exploited, this procedure has its own benefits: first, it renders high-frequency forecast instances; second, it allows taking into account misspecification due to a seasonal response of (y_t) to (z_r). This is especially important whenever multiple time series with different sampling frequencies are combined in one model and seasonality effects are either difficult to detect or impossible to avoid. In the context of macroeconomic forecasting, we refer the reader to Clements and Galvão (2008, 2009), Chen and Ghysels (2010) and Jardet and Meunier (2020) where these questions are carefully discussed.

  We now formally present the ***high-frequency forecasting*** scheme in our notation. Let the forecaster place herself at time t: she wishes to construct a high-frequency forecast for some $t, \ell|\kappa$ with $\ell \in \mathbb{N}$. The maximal information set available at t is clearly $\mathcal{F}_t$ as in (2.1). However, if she uses $\mathcal{F}_t$ then the forecast for $t, \ell|\kappa$ coincides with the fixed forecast and is given by (2.2) for any ℓ. Notice that the forecasts can be constructed using the *reduced* information sets instead, which for $0 < \ell \leq \kappa - 1$ are defined as

$$\mathcal{F}^H_{t,\ell} = \sigma\left(\left\{y_{t-1}, y_{t-2}, y_{t-3}, \ldots, z_{t,-\ell|\kappa}, z_{t,-\ell-1|\kappa}, z_{t,-\ell-2|\kappa}, \ldots\right\}\right),$$

  and for $\ell \geq \kappa$ and $h = \lfloor \ell/\kappa \rfloor$ can be written as

$$\begin{aligned}\mathcal{F}^H_{t,\ell} &= \sigma\left(\left\{y_{t+h-\lceil \ell/\kappa \rceil}, y_{t+h-1-\lceil \ell/\kappa \rceil}, \ldots, z_{t+h,-\ell|\kappa}, z_{t+h,-\ell-1|\kappa}, z_{t+h,-\ell-2|\kappa}, \ldots\right\}\right)\\ &= \sigma\left(\left\{y_{t-1}, y_{t-2}, y_{t-3}, \ldots, z_{t,-\ell \bmod \kappa|\kappa}, z_{t,-(\ell+1) \bmod \kappa|\kappa}, z_{t,-(\ell+2) \bmod \kappa|\kappa}, \ldots\right\}\right).\end{aligned} \tag{2.3}$$

  The high-frequency forecast information sets nest the fixed forecasting setup since $\mathcal{F}^H_{t,\ell} \equiv \mathcal{F}_t$ if $\ell = \kappa h$ for $h \in \mathbb{N}$.

- **Multicasting:** One always aims to construct one-step and multistep forecasts by using all the available information at a given point in time. It is, therefore, natural to compare models by constructing high-frequency nowcasts for the target variable to be released at the end of the current period and its high-frequency proxy forecasts for the next periods. To avoid confusion, we refer to this situation as *multicasting*. More explicitly, provided that the forecaster finds herself at time index $t, s|\kappa$ and is interested in all the forecasts up to some maximal low-frequency horizon $H \geq 1$, for each $1 \leq \ell \leq H\kappa$ the multicasting scheme yields the following combination:
  (a) *Nowcasting* when $0 < \ell \leq s$, with information sets $\mathcal{F}^N_{t,\ell}$.
  (b) Forecasting when $\ell > s$:
    (i) *Fixed forecasting* if ℓ satisfies $\ell \bmod \kappa = 0$, with information set $\mathcal{F}_t$.
    (ii) *High-frequency forecasting* if $\ell \bmod \kappa \neq 0$, with information sets $\mathcal{F}^H_{t,\ell}$.

3 Reservoir Models

In this section we introduce a reservoir computing (RC) model for time series forecasting. We focus on a family of RC systems called *Echo State Networks*, which have been succesfully applied to the forecasting deterministic dynamical systems Jaeger and Haas (2004), Pathak et al. (2017, 2018), Wikner et al. (2021), Arcomano et al. (2022) but have yet to be fully developed in the forecasting of stochastic time series in general, and in the mixed-frequencies context, in particular. In the following paragraphs, we explain the linear estimation of ESN model parameters, the hyperparameters tuning, the loss penalty selection, and how to carry out nonlinear forecasting. Finally, we propose a multi-frequency ESN modeling strategy, a generalization of the standard Echo State Network setup, that can be implemented with single unified or with multiple frequency-specific state-space equations.

3.1 Reservoir Models

Echo State Networks (ESNs) are nonlinear state-space models belonging to the Reservoir Computing (RC) family, which more broadly can be characterized as recurrent neural network models. The success of RC modeling in a range of scientific data analysis and forecasting applications has been driven by their ease of training and the possibility of implementing RC systems using high-performance dedicated hardware (see Tanaka et al. (2019) for an overview). The following nonlinear state-space system provides a general setup for reservoir computing models

$$\boldsymbol{x}_t = F(\boldsymbol{x}_{t-1}, \boldsymbol{z}_t), \tag{3.1}$$
$$\boldsymbol{y}_t = h_{\boldsymbol{\theta}}(\boldsymbol{x}_t), \tag{3.2}$$

for all $t \in \mathbb{Z}$, where the state space map $F : \mathbb{R}^N \times \mathbb{R}^K \to \mathbb{R}^N$, $N, K \in \mathbb{N}^+$ is called in our context the *reservoir map*, and $h_{\boldsymbol{\theta}} : \mathbb{R}^N \to \mathbb{R}^J$, $J \in \mathbb{N}^+$ is called the *readout* or the *observation map*. The sequences $(\boldsymbol{z}_t)_{t\in\mathbb{Z}}$, $\boldsymbol{z}_t \in \mathbb{R}^K$, and $(\boldsymbol{y}_t)_{t\in\mathbb{Z}}$, $\boldsymbol{y}_t \in \mathbb{R}^J$, stand for the *input* and the *output (target)* of the system, respectively, and $(\boldsymbol{x}_t)_{t\in\mathbb{Z}}$, $\boldsymbol{x}_t \in \mathbb{R}^N$, are the associated *reservoir states*. In what follows, we shall assume that the readout map $h_{\boldsymbol{\theta}}(\cdot)$ is parameterized by $\boldsymbol{\theta} \in \Theta$ and these parameters need to be estimated using a sample of observations. Moreover, in cases where both inputs and targets are stochastic processes, we consider the following observation equation:

$$\boldsymbol{y}_t = h_{\boldsymbol{\theta}}(\boldsymbol{x}_t) + \boldsymbol{\epsilon}_t, \tag{3.3}$$

where $(\boldsymbol{\epsilon}_t)_{t\in\mathbb{Z}}$ are J-dimensional independent zero-mean innovations with variance $\sigma^2_\epsilon \mathbb{I}_J$ that are also independent of $\boldsymbol{x}_t$ across all t. In the particular case of the ESN model, the reservoir map is chosen so that it resembles the neuron dynamics in a recurrent neural networks and the readout map is chosen to be an affine function. The ESN model is hence given by

$$\begin{aligned}\boldsymbol{x}_t &= \alpha \boldsymbol{x}_{t-1} + (1-\alpha)\sigma(A\boldsymbol{x}_{t-1} + C\boldsymbol{z}_t + \boldsymbol{\zeta}) \\ \boldsymbol{y}_t &= \boldsymbol{a} + W^\top \boldsymbol{x}_t + \boldsymbol{\epsilon}_t,\end{aligned}$$

where $A \in \mathbb{M}_N$ is the *reservoir matrix*, $C \in \mathbb{M}_{N,K}$ is the *input matrix*, $\boldsymbol{\zeta} \in \mathbb{R}^N$ is the *input shift*, $\alpha \in [0, 1)$ is the *leak rate* and $W \in \mathbb{M}_{N,J}$ are the *readout weights*. The map $\sigma : \mathbb{R} \to \mathbb{R}$ is a sigmoid function applied elementwise; we shall generally take σ as tanh. ESNs have been proven to have universal approximation properties for L^p-integrable stochastic processes (Gonon and Ortega (2019)), and estimation and generalization error bounds have been established in Gonon et al. (2020a, 2022).

Note that, since the observation equation is linear in $\boldsymbol{x}_t$, the weights W can be estimated via either a least-squares estimator or, more commonly, using a ridge regression of $\boldsymbol{y}_t$ on $\boldsymbol{x}_t$. To simplify notation, we will suppress the intercept $\boldsymbol{a}$ from the observation equation since its inclusion in the linear estimation of W is trivial.

The core advantage of an ESN model is that, unlike factor models or recurrent neural networks, the state equation features fixed, randomly sampled parameter matrices A, C, and $\boldsymbol{\zeta}$. Therefore, in ESNs, states are always directly observable given inputs, whereas in a factor model, one must infer latent factors, as well as estimate model parameters such as loading matrices. Also, unlike standard recurrent networks, there is no need to estimate all model parameters jointly via gradient descent algorithms, making model fitting a straightforward linear regression problem.

So far, we have left the nature of ESN inputs $(\boldsymbol{z}_t)$ and targets $(\boldsymbol{y}_t)$ undetermined. Since we are interested in forecasting, we can now be more precise about the timing of targets with respect to the inputs. We write

$$\boldsymbol{x}_t = \alpha \boldsymbol{x}_{t-1} + (1-\alpha)\sigma(A\boldsymbol{x}_{t-1} + C\boldsymbol{z}_t + \boldsymbol{\zeta}) \tag{3.4}$$

$$\boldsymbol{y}_{t+1} = W^\top \boldsymbol{x}_t + \boldsymbol{\epsilon}_{t+1}, \tag{3.5}$$

where $\boldsymbol{y}_{t+1}$ is the target one-step ahead observation of input $\boldsymbol{z}_t$. If $\boldsymbol{y}_t$ has J components, but only one series is of interest, the linear regression equation in (3.5) can be partitioned to involve only the jth component $y_{j,t+1}$, that is

$$y_{j,t+1} = W_j^\top \boldsymbol{x}_t + \epsilon_{j,t+1}.$$

Here, W_j is the jth column vector of W as defined in (3.5). As we shall see in Section 3.3, it is nonetheless often necessary to consider a multivariate regression equation, especially in multistep forecasting setups.

3.2 Estimation

Assume that a sample $(\boldsymbol{z}_t)_{1:T-1}$, $(\boldsymbol{y}_t)_{2:T}$ of inputs and targets is available. Given an initial state $\boldsymbol{x}_0$, states $\boldsymbol{x}_1$, …, $\boldsymbol{x}_T$ can be readily computed according to (3.4), and W in (3.5) can be estimated by ordinary linear regression. Recursive/online versions of the least squares estimator, $\widehat{W}^{\mathrm{RLS}}$, as presented in Young (2012), can be used although this is a methodology that we do not implement in this paper. Define $X = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{T-1})^\top \in \mathbb{M}_{T-1,N}$, $Y = (\boldsymbol{y}_2, \boldsymbol{y}_3, \ldots, \boldsymbol{y}_T)^\top \in \mathbb{M}_{T-1,J}$. The least squares estimator $\widehat{W}^{\mathrm{LS}}$ is given by

$$\widehat{W}^{\mathrm{LS}} = (X^\top X)^{-1} X^\top Y. \tag{3.6}$$

Even though the result of this estimation problem is obviously dependent on the initial state condition $\boldsymbol{x}_0$, the architecture of the ESN will be typically chosen (see Section 3.2.3) so that the socalled *fading memory property* holds Boyd and Chua (1985), a corollary of which is the *state forgetting property* (see, for instance, (Grigoryeva and Ortega, 2019, Theorem 27)) that states that the importance of the choice of $\boldsymbol{x}_0$ fades away as the time-length of the sample increases. Even though, $\widehat{W}^{\mathrm{LS}}$ is only properly estimated when $N < T$, it is not uncommon for ESN models to require very large state-spaces (in the order of 10^3-10^4 dimensions, see for example Pathak et al. (2017)). This makes desirable to apply echo state networks such that $N \geq T$. Given that the OLS estimator has no unique solution in this case, we consider the ridge regression for the estimation of W. The ridge estimator is given by

$$\widehat{W}^{\mathrm{R}}_\lambda = (X^\top X + \lambda((T-1)\,\mathbb{I}_N))^{-1} X^\top Y, \tag{3.7}$$

where $\lambda > 0$ is the penalty strength. When $\lambda \to 0$, the estimator $\widehat{W}^{\mathrm{R}}_\lambda$ converges to the minimum-norm least squares solution (Ishwaran and Rao (2014)). In applications, ridge regression is the most commonly used estimation method applied to ESNs, as it provides a straightforward regularization scheme both when $N < T$ and $N \geq T$. Moreover, the virtue of the ridge regression problem is the fact that the objective function is convex and hence, even in those cases when $\min\{N, T\}$ is large, it can be efficiently solved using stochastic gradient descent.

Notice that (3.4)-(3.5) is a nonlinear state-space model in the context of nonparametric estimation. Hence it is clear that a natural trade-off between bias and variance exists due to the choice of N. Since ESNs have a natural connection to random-weights neural networks (Cao et al. (2018)) and random projection regression (Maillard and Munos (2012)), one may consider comparing them to nonparametric sieve methods in an asymptotic framework. If the data were independently sampled, classical results on consistent linear sieve estimation would require that $N^2/T = o(1)$ (see for example Chen (2007)). Belloni et al. (2015) have proven that under mild conditions one may weaken rates to $N/T = o(1)$ up to log factors, and Chen and Christensen (2015) have generalized this result by showing that for dependent data $N(log(N))^2/T = o(1)$ is sufficient. For single-layers neural networks specifically, Chen and White (1999) demonstrated a rate of $N^{2d} log(N)/T = o(1)$ with $d > 1$ for ϕ-mixing and β-mixing data generating processes. Though there are only limited results on the statistical properties of reservoir models (see, for instance, Grigoryeva and Ortega (2018b), Gonon et al. (2022)), sieve rates seem to suggest that choosing $N = O(T)$ in echo state networks could lead to nontrivial forecasting bias owing itself to poor approximation properties. This comparison, however, is only qualitative, as it does not acknowledge the nonlinear state-space structure of ESNs.

The well-known bias-variance trade-off for out-of-sample generalization error also points to poor forecasting performance when a model is at the interpolation threshold. While ridge regression is not necessarily a solution to estimator inconsistency, it is commonly applied to address generalization concerns in statistical learning (see Hastie et al. (2009)). Recently the link between regularization and generalization has also been studied more accurately and Hastie et al. (2022) have shown that "ridgeless" (i.e. interpolation) solutions can be optimal in some scenarios. However, in our empirical evaluations in Section 5, cross-validation (CV) consistently selects non-negligible ridge penalties even for large models, implying that ridge penalization indeed plays an important role in the ESN forecasting performance – at least in applications involving macroeconomic-financial data with multiple sampling frequencies. As statistical inference issues are beyond the scope of this work, we consider the use of the ridge estimator $\widehat{W}^{\mathrm{R}}_{\lambda}$ as a means to control the generalization error in line with the ESN literature (we refer the reader to Hart et al. (2021) where some of these questions are considered for the case of dynamical systems).

3.2.1 Fixed, Expanding and Rolling Window Estimation

Model parameter stability is an important and well-studied question in linear time series analysis. Indeed, the identification and modeling of structural breaks play a key role in macroeconomic modeling. To account for this possibility, we compare multiple estimation setups which may account for possible changes in model parameters.

Suppose again that a sample $Y = (\boldsymbol{y}_2, \boldsymbol{y}_3, \ldots, \boldsymbol{y}_T)^\top \in \mathbb{M}_{T-1,J}$ of targets is available, an initial state $\boldsymbol{x}_0$ is given and regressors $Z = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{T-1})^\top \in \mathbb{M}_{T-1,K}$ are observed. Additionally, the researcher has available an out-of-sample dataset, $Y^\dagger = (\boldsymbol{y}_{T+1}, \boldsymbol{y}_{T+2}, \ldots, \boldsymbol{y}_{T+S})^\top \in \mathbb{M}_{S,J}$, $Z^\dagger = (\boldsymbol{z}_T, \boldsymbol{z}_{T+1}, \ldots, \boldsymbol{z}_{T+S-1})^\top \in \mathbb{M}_{S,K}$ for $S \geq 1$. We define the following estimation setups:

(i) **Fixed parameters**: An estimator $\widehat{W}$ is computed strictly over sample observations Y and Z; tuning parameters are also chosen with data available up to time T. Model parameters are kept fixed as the estimated model is applied to construct forecasts $\widehat{\boldsymbol{y}}_{T+1}, \widehat{\boldsymbol{y}}_{T+2}, \ldots, \widehat{\boldsymbol{y}}_{T+S}$ as out-of-sample regressors $\boldsymbol{z}_T, \boldsymbol{z}_{T+1}, \ldots, \boldsymbol{z}_{T+S-1}$ are added to the information set.

(ii) **Expanding window:** Given estimator $\widehat{W}$, for each out-of-sample time step $s = 1, \ldots, S$, define $\widehat{W}^{\mathrm{EW}}_s$ as the estimate computed by "expanding" the sample window up to time $T + s$, given by $Y^{\mathrm{EW}}_s := (\boldsymbol{y}_2, \boldsymbol{y}_3, \ldots, \boldsymbol{y}_T, \boldsymbol{y}_{T+1}, \ldots, \boldsymbol{y}_{T+s})^\top$ and $Z^{\mathrm{EW}}_s := (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{T-1}, \boldsymbol{z}_T, \ldots, \boldsymbol{z}_{T+s-1})^\top$. Tuning parameters can be re-estimated over windows Y^{EW}_s, Z^{EW}_s.

(iii) **Rolling window:** Unlike in the above expanding window case, in this setup the within-window sample size is kept fixed across windows – that is, the sample window "rolls" over the data –

by defining $\widehat{W}_s^{\mathrm{RW}}$ as the estimate over $Y_s^{\mathrm{RW}} := (\boldsymbol{y}_{2+s}, \boldsymbol{y}_{3+s}, \ldots, \boldsymbol{y}_{T+s-1}, \boldsymbol{y}_{T+s})^\top$ and $Z_s^{\mathrm{RW}} := (\boldsymbol{z}_{1+s}, \boldsymbol{z}_{2+s}, \ldots, \boldsymbol{z}_{T+s-2}, \boldsymbol{z}_{T+s-1})^\top$ for $s = 1, \ldots, S$. Parameters are re-estimated over windows Y_s^{RW}, Z_s^{RW}.

The fixed-parameter setup is the most rigid one. It builds upon the idea that the initial sample contains sufficient information for correct model estimation and forecasting and that the model parameters are constant. Its theoretical analysis is hence relatively easy since there is no need to discuss the stability of the penalty and the hyperparameters across sample windows. An expanding window setup is based on the belief that newly available data contains key information to produce forecasts and, therefore, must be continuously incorporated. In essence, forecasters do this when they re-estimate a model at each data release cycle. In the case of a rolling window estimation strategy, one can theoretically handle model changes. Although proper structural break modeling would require a consistent identification of breakpoints, rolling window estimation can potentially accommodate slow drifts in model parameters over time by directly discarding old data, unlike with an expanding window. We do not explore the selection of an optimal window size, which in rolling window estimation has been shown to improve forecasting performance (Inoue et al. (2017)).

3.2.2 Penalty Selection

Penalty selection via cross-validation with time series data has already been studied, and its validity has been established in Bergmeir et al. (2018). In our empirical study, we use a simple out-of-sample cross-validation (CV) strategy with ten folds over the training split of the data. The construction of the folds is sequential and involves properly accounting for the time dependence of the data. Because ESN models involve relatively large dimensional state-spaces, we want to split the data in a parsimonious way. To do this, we select a testing window of five target observations, and all preceding observations (at any frequency) are used to fit the model. Splitting proceeds from the end of the sample backward, and we always use one-step-ahead target observations to construct the loss. In expanding or rolling windows setups, we further re-validate the penalty via out-of-sample CV as the sample grows: this is important to ensure that, as the sample sizes grow, estimated ESN weights do not induce over-smoothing but rather reflect the information contained in the window. For additional details, we refer the reader to Appendix 7.1.

3.2.3 Hyperparameter Tuning

A fundamental result in the ESN literature is that the structure of the matrices $(A, C, \boldsymbol{\zeta})$ determines the dynamics of the states $(\boldsymbol{x}_t)$ and much work has been put into determining optimal specifications (see for example Rodan and Tino (2011), Goudarzi et al. (2016), Farkas et al. (2016), Grigoryeva et al. (2015, 2016), Gonon et al. (2020b)). In what follows, we call the tuple $(A, C, \boldsymbol{\zeta})$ the *parameters* of the ESN that we use to determine what we call the *hyperparameters* in the echo state model equations. Indeed, rewrite the state equation (3.4) as

$$\boldsymbol{x}_t = \alpha \boldsymbol{x}_{t-1} + (1-\alpha)\sigma(\rho \overline{A} \boldsymbol{x}_{t-1} + \gamma \overline{C} \boldsymbol{z}_t + \omega \overline{\boldsymbol{\zeta}}),$$

where $\overline{A} = A/|\lambda_1(A)|$ with $\lambda_1(A)$ denoting the largest eigenvalue of A, $\overline{C} = C/\|C\|$ and $\overline{\boldsymbol{\zeta}} = \boldsymbol{\zeta}/\|\boldsymbol{\zeta}\|$ are the normalized input matrix and normalized input shift, respectively. The choice of the norm $\|\cdot\|$ is inconsequential, so we will be using the 2-norm in our empirical study. The hyperparameter $\rho \in \mathbb{R}_+$ is called the *spectral radius* of the reservoir; $\gamma \in \mathbb{R}_+$ is the *input scaling*, and $\omega \in \mathbb{R}_+$ is the *shift scaling*. The choice of the hyperparameters defines the properties of the state map. In particular, it is standard to ensure the existence and the uniqueness of solutions of the state equation for semi-infinite inputs via the so-called *Echo State Property* (see Grigoryeva and Ortega (2018a,b, 2019) for the details) as well as the continuity of the resulting input/output map in order to obtain the so-called *Fading Memory Property*. We consider α as a hyperparameter since it is often tuned for each specific application of the ESN model. Notice that if $\rho = 0$ and $\alpha = 0$ the state equation reduces to $\boldsymbol{x}_t = \sigma(\gamma \overline{C} \boldsymbol{z}_t + \omega \overline{\boldsymbol{\zeta}})$, there are no autoregressive dynamics and the ESN turns into a nonlinear regression model with random coefficients (in other words, a feedforward neural network with random weights which is usually referred to as a *Extreme Learning Machine*, see Cao et al. (2018) for an extensive review).

We now propose a general scheme that allows us to jointly select hyperparameters $\boldsymbol{\varphi} := (\alpha, \rho, \gamma, \omega)$ for a model of the form (3.4)-(3.5). Our approach builds on the idea of leave-one-out cross-validation for time series models. Using a fixed, expanding, or rolling window over the training data, we can always compute the one-step forecasting errors committed by the ESN, given fixed normalized model matrices $(\overline{A}, \overline{C}, \overline{\boldsymbol{\zeta}})$ and a hyperparameter vector $\boldsymbol{\varphi}$. By choosing an appropriate loss function $\boldsymbol{\ell} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ we can thus compute the empirical ESN forecasting error

$$\mathcal{L}_T(\boldsymbol{\varphi}) := \sum_{t=T_0}^{T-1} \boldsymbol{\ell}(\boldsymbol{y}_{t+1}, \widehat{W}_t(\boldsymbol{\varphi})^\top \boldsymbol{x}_t),$$

where $\widehat{W}_t(\boldsymbol{\varphi})$ is the readout weights estimator involving data available up to time t and $1 < T_0 < T-1$ is the minimum number of observations used for fitting. Notice that if $\boldsymbol{\ell}(\boldsymbol{y}_{t+1}, \widehat{W}_t(\boldsymbol{\varphi})^\top \boldsymbol{x}_t) = \|\boldsymbol{y}_{t+1} - \widehat{W}_t(\boldsymbol{\varphi})^\top \boldsymbol{x}_t\|_2^2$ then $\mathcal{L}_T(\boldsymbol{\varphi})$ is simply the squared loss that is minimized in training (modulo a ridge penalty term). Here, however, the interest lies not in finding the matrix W, which minimizes $\mathcal{L}_T$, but rather the optimal hyperparameter vector

$$\boldsymbol{\varphi}^* \in \arg\min_{\boldsymbol{\varphi}} \mathcal{L}_T(\boldsymbol{\varphi}). \tag{3.8}$$

We highlight that to tune $\boldsymbol{\varphi}$ one may choose a norm, metric, or pseudo-metric $\boldsymbol{\ell}$ that is different from the one used in the estimation of the readout map W.

We present the entire hyperparameter optimization routine in Algorithm 1. Note that step (i) might entail re-normalizing inputs and targets at each window t. This setup is general and allows applying any global optimization routine to minimize $\mathcal{L}_T(\boldsymbol{\varphi})$. We construct the loss $\mathcal{L}_T(\boldsymbol{\varphi}_j)$ sequentially, that is by summing squared residuals of the model estimated in step (i) of Algorithm 1 when $\boldsymbol{\ell}$ is a qudratic loss. One can program $\mathcal{L}_T(\boldsymbol{\varphi}_j)$ via Tensoflow so that the gradient can be evaluated by backpropagation in Algorithm 1 (iii). Since there is no guarantee that the objective function is convex or even everywhere smooth, we suggest applying optimizers known to explore the search space efficiently. We emphasize at this point that the lack of convexity guarantees for the loss functions is much more consequential for the other models that we present later on and that we shall be using as benchmarks, in particular for the MIDAS model. Indeed, as we discuss in Appendix 7.5.1, even though explicit gradient calculations might help the optimization for the MIDAS models (see Kostrov (2021) for a detailed discussion), multiple suboptimal minima for MIDAS can not be ruled out.

The parameter space is bounded, since $\boldsymbol{\varphi} \in [0,1) \times [0,\overline{\rho}] \times [0,\overline{\gamma}] \times [0,\overline{\omega}]$ and, normalizing both data and state-space matrices, we suggest that upper bounds $\overline{\rho}$, $\overline{\gamma}$, and $\overline{\omega}$ can be safely set to be less than or equal to 10.

One issue with the state formulation in (3.4) and thus with the hyperparameter optimization routine just described, is that $\boldsymbol{\varphi} = (\alpha, \rho, \gamma, \omega)$ can not always be point identified. For example, if we simplify the state equation to be linear and let $\alpha = \omega = 0$, it is obvious that the ESN model is system isomorphic Grigoryeva and Ortega (2021) to $\boldsymbol{x}_t^* = d\rho\overline{A}\boldsymbol{x}_{t-1}^* + d\gamma\overline{C}\boldsymbol{z}_t$, $\boldsymbol{y}_t = d^{-1}W\boldsymbol{x}_t^* + \boldsymbol{\epsilon}_t$ for all $d \neq 0$. This issue also arises in nonlinear models, for example when $\sigma \equiv \tanh$ and γ is sufficiently small. Parameter identification in nonlinear models has been extensively studied in semi- and nonparametric cross-sectional regressions. For instance, it is known that in certain setups, point identification requires a proper normalization to be imposed. The interested reader can refer to Section 6.3 of Horowitz (2009) for a discussion in a similar vein regarding nonparametric transformation models. Since we often set $\omega = 0$, hyperparameter identification can be a significant issue when attempting model tuning. When $\omega = 0$, a simple but valuable reparametrization is given by

$$\boldsymbol{x}_t = \alpha\boldsymbol{x}_{t-1} + (1-\alpha)\sigma\left(\psi\overline{A}\boldsymbol{x}_{t-1} + \overline{C}\boldsymbol{z}_t\right),$$

where $\psi = \rho/\gamma$. This prescription allows decoupling ρ and γ at the cost of the constant input scaling, which may be undesirable whenever one wants to attenuate the nonlinearity induced by the sigmoid

Algorithm 1: Hyperparameter tuning

```
Data: Sample $\boldsymbol{y}_{2:T} = \{\boldsymbol{y}_2, \boldsymbol{y}_3, \ldots, \boldsymbol{y}_T\}$, $\boldsymbol{z}_{1:T-1} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{T-1}\}$, initial state $\boldsymbol{x}_0$, initial
    guess $\boldsymbol{\varphi}_0$, convergence criterion Crit, maximal algorithm iterations MaxIter. If ridge
    regression is used to estimate $W$, fixed regularization strength $\lambda > 0$.
Result: $\boldsymbol{\varphi}^*$
Fix $T$ and determine the model fit windows for $t = T_0, \ldots, T-1$. Choose whether the ESN
  model is estimated with fixed or rolling window;
$j = 0$;
while (not Crit) and ($j <$ MaxIter) do
    (i) Given $\boldsymbol{\varphi}_j$, estimate coefficient matrices
          $$\left(\widehat{W}_t(\boldsymbol{\varphi}_j)\right)_{T_0:T-1},$$
     where possibly $\widehat{W}_t(\boldsymbol{\varphi}_j)$ does not depend on $t$, e.g. in the fixed estimation setup;
    (ii) Compute
          $$\mathcal{L}_T(\boldsymbol{\varphi}_j) := \sum_{t=T_0}^{T-1} \boldsymbol{\ell}\left(\boldsymbol{y}_{t+1}, \widehat{W}_t(\boldsymbol{\varphi}_j)^\top \boldsymbol{x}_t\right),$$
     the total one-step-ahead forecasting loss;
    (iii) Update $\boldsymbol{\varphi}_{j+1} \leftarrow \boldsymbol{\varphi}_j$ with an appropriate rule (for example,the gradient descent of $\mathcal{L}_T$ in
     the direction of $\boldsymbol{\varphi}_j$; in our applications, we use variants L-BFGS-B and pattern search);
    (iv) $j \leftarrow j + 1$, update Crit;
```

map without also reducing the spectral radius.[2] It is immediate to modify the optimization scheme to deal with the case $\widetilde{\boldsymbol{\varphi}} = (\alpha, \psi)$. In the sequel, we assume that the ESN models are estimated using the approaches proposed in this subsection and use the conventional ESN specification as in (3.4)-(3.5) to discuss the forecasting strategy.

3.3 ESN Forecasting

We are interested in using ESN models to construct conditional forecasts of target variables. Given that the conditional mean is the best mean square error estimator for h-step-ahead target $\boldsymbol{y}_{t+h}$, $h \geq 1$, our main focus is approximating

$$\widehat{\boldsymbol{y}}_{t+h|t} := \mathbb{E}\left[\boldsymbol{y}_{t+h} | \boldsymbol{x}_{0:t}, \boldsymbol{z}_{0:t}\right]. \tag{3.9}$$

The case $h = 1$ is trivial, because the ESN model is estimated by regressing $\boldsymbol{y}_{t+1}$ on state $\boldsymbol{x}_t$, and thus we can set $\widetilde{\boldsymbol{y}}_{t+1|t} = \widehat{W}_t(\boldsymbol{\phi}^*)^\top \boldsymbol{x}_t$. However, when $h > 1$ the nonlinear state dynamics precludes a direct computation of the conditional mean. This is in contrast to linear models like VARMAs or DFMs, where the assumption of linearity implies that conditional expectations reduce to simple matrix-vector operations. In particular, linear models are such that the variance (and any other higher-order moments) of the noise term do not impact the conditional mean forecast.

Let $p_\theta(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{z}_t)$ and $g_\theta(\boldsymbol{y}_{t+1} | \boldsymbol{x}_t)$ be the state transition and observation densities, respectively. Then, for $h > 1$,

$$\widehat{\boldsymbol{y}}_{t+h|t} = \int \boldsymbol{y}_{t+h}\, g_\theta(\boldsymbol{y}_{t+h} | \boldsymbol{x}_{t+h-1}) \prod_{j=1}^{h-1} p_\theta(\boldsymbol{x}_{t+j} | \boldsymbol{x}_{t+j-1}, \boldsymbol{z}_{t+j}) \nu(\boldsymbol{z}_{t+j} | \boldsymbol{x}_{t+j-1}) \mathrm{d}\boldsymbol{z}_{t+j} \mathrm{d}\boldsymbol{x}_{t+j} \mathrm{d}\boldsymbol{y}_{t+h}, \tag{3.10}$$

[2] One can fix $\overline{C}$ to have a different scaling before optimizing the hyperparameter ψ. However, this amounts to one more *ex ante* model tuning step.

where $\nu(\boldsymbol{z}_{t+j}|\boldsymbol{x}_{t+j-1})$ is the conditional density of inputs. Here, we introduce the additional assumption that $\boldsymbol{x}_{t+j-1}$ is sufficient to condition on past states and inputs, that is

$$\nu(\boldsymbol{z}_{t+j}|\boldsymbol{x}_{t+j-1}) \equiv \nu(\boldsymbol{z}_{t+j}|\boldsymbol{x}_{0:t+j-1}, \boldsymbol{z}_{0:t+j-1}).$$

Some elements in the expectation integral are not directly available. Specifically, while an ESN explicitly models both $p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{z}_t)$ and $g_\theta(\boldsymbol{y}_{t+1}|\boldsymbol{x}_t)$, the density $\nu(\boldsymbol{z}_{t+j}|\boldsymbol{x}_{t+j-1})$ is unavailable.

In the remaining part of this subsection, we present two approaches to forecasting that are available for the target variable. In our empirical study, we exclusively work with the first one, which is more suitable for forecasting in cases in which the input and target variables are sampled at different frequencies, and shows superior performance compared to the other standard benchmarks, as we demonstrate later in Section 5.

Forecasting of target time series via iterative forecasting of inputs. We are interested in constructing forecasts of target variables which are, in general, not the same as the model inputs. We propose a method that resolves the issue of the intractability of (3.10) and that capitalizes on the available results that use ESNs for the forecasting of dynamical systems. More explicitely, we introduce an equation in addition to the ESN specification that allows us to sidestep the modeling of the density ν directly, thus making feasible the computation of $\widehat{\boldsymbol{y}}_{t+h|t}$ even when $h > 1$. Consider the model

$$\boldsymbol{x}_t = \alpha \boldsymbol{x}_{t-1} + (1-\alpha)\sigma(A\boldsymbol{x}_{t-1} + C\boldsymbol{z}_t + \boldsymbol{\zeta}) \tag{3.11}$$

$$\boldsymbol{z}_{t+1} = \mathcal{W}^\top \boldsymbol{x}_t + \boldsymbol{u}_{t+1}, \tag{3.12}$$

where we use the symbol $\mathcal{W}$ for the output matrix in order to separate this case from the general ESN equations and where $(\boldsymbol{u}_t)_{t\in\mathbb{Z}}$ are K-dimensional independent zero-mean innovations with variance $\sigma_u^2 \mathbb{I}_K$ that are independent of $\boldsymbol{x}_t$ across all t.

In this case, the reservoir map $F(\boldsymbol{x}_{t-1}, \boldsymbol{z}_t)$ in (3.1) is determined by (3.11). This setup allows for re-feeding the forecasted variables back into the state equation as inputs, providing the following state recursion:

$$\boldsymbol{x}_t = F(\boldsymbol{x}_{t-1}, \mathcal{W}^\top \boldsymbol{x}_{t-1} + \boldsymbol{u}_t) =: G_\theta(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t),$$

where the subscript θ denotes the dependence on the model coefficients. In the reservoir computing literature, regimes where the ESN state equation is iteratively fed with the model outputs are called "autonomous" (Gonon et al., 2020b). They are widely and successfully utilized for the prediction of deterministic dynamical systems. Indeed, in those instances, provided that the $\widehat{\mathcal{W}}$ estimate is available from data, the $h > 1$ steps autonomous state iteration is given by

$$F_\theta^*(\boldsymbol{x}_t) := \alpha \boldsymbol{x}_t + (1-\alpha)\sigma((A + C\widehat{\mathcal{W}}^\top)\boldsymbol{x}_t + \boldsymbol{\zeta}) \tag{3.13}$$

and

$$\boldsymbol{x}_{t+h} = \underbrace{F_\theta^* \circ F_\theta^* \circ \cdots \circ F_\theta^*}_{h \text{ times}}(\boldsymbol{x}_t). \tag{3.14}$$

Hence one can directly obtain the h-steps ahead predictions of the input time series as $\boldsymbol{z}_{t+h} = \widehat{\mathcal{W}}^\top \boldsymbol{x}_{t+h-1}$.

In the case of stochastic target variables, we notice that for the conditional forecast of the states, it holds that

$$\int \boldsymbol{x}_t \, p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{z}_t)\nu(\boldsymbol{z}_t|\boldsymbol{x}_{t-1})\mathrm{d}\boldsymbol{z}_t = \int G_\theta(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t)\phi(\boldsymbol{u}_t)\mathrm{d}\boldsymbol{u}_t, \tag{3.15}$$

where the density ϕ is unavailable. Additionally, even under Gaussianity assumptions, which are standard in the filtering literature, namely $\boldsymbol{u}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_{\boldsymbol{u}})$, the presence of the nonlinear map G_θ makes the computation of the forecasts of $\boldsymbol{z}_{t+h}$ a non-straightforward exercise. Nevertheless, this forecast construction can be readily used when one is interested exclusively in predicting the time series $\boldsymbol{z}_t$.

Whenever the final goal of the exercise is forecasting some other explained variable $\boldsymbol{y}_{t+h}$ h-steps ahead, additional issues arise. In this case, one needs to compute the conditional expectation in (3.10) which is intractable even under Gaussian assumptions on the innovations. One option is to apply particle filtering techniques such as bootstrap sampling or sequential importance sampling (SIS) to evaluate the expectation (Doucet et al., 2001). We emphasize that the state-space dimension is usually chosen to be large, and hence implementing the standard filtering or more sophisticated techniques requires some care.

Our approach is to avoid dealing with the nonlinear densities involved in (3.10) with the help of (3.15) and, instead, to reduce the computation of the conditional expectation $\widehat{\boldsymbol{y}}_{t+h|t}$ to a composition of functions. By the linearity of the observation equation (3.5) and the assumption of independence in the zero-mean noise $\boldsymbol{\epsilon}_{t+h}$ we write

$$\widehat{\boldsymbol{y}}_{t+h|t} = \int W^\top \boldsymbol{x}_{t+h-1} \prod_{j=1}^{h-1} p_\theta(\boldsymbol{x}_{t+j}|\boldsymbol{x}_{t+j-1}, \boldsymbol{z}_{t+j}) \nu(\boldsymbol{z}_{t+j}|\boldsymbol{x}_{t+j-1}) \mathrm{d}\boldsymbol{x}_{t+j} \mathrm{d}\boldsymbol{z}_{t+j}$$

and use the approximation

$$\widetilde{\boldsymbol{y}}_{t+h} = W^\top \underbrace{F_\theta^* \circ F_\theta^* \circ \cdots \circ F_\theta^*}_{h-1 \text{ times}} (\boldsymbol{x}_t), \tag{3.16}$$

which originates from

$$\int G_\theta(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t)\phi(\boldsymbol{u}_t)\mathrm{d}\boldsymbol{u}_t \approx G_\theta(\boldsymbol{x}_{t-1}, \mathbb{E}[\boldsymbol{u}_t]) = F(\boldsymbol{x}_{t-1}, \mathcal{W}^\top \boldsymbol{x}_{t-1}) \equiv F_\theta^*(\boldsymbol{x}_{t-1}), \tag{3.17}$$

where $\boldsymbol{u}_t$ is assumed to be zero-mean. The validity of (3.17) itself requires implicit assumptions on the nature of the distribution of $\boldsymbol{u}_t$, but here we want to keep the analysis of $\widehat{\boldsymbol{y}}_{t+h|t}$ to a minimum, and just use the insights from the dynamical systems ESN literature. We are hence not delving deeper into alternative approaches to estimate forecasts or, more generally, to compute conditional expectations of ESN models with stochastic inputs.

Direct forecasting of target time series. Alternatively to forecasting target variables with the iteratively predicted input time series, one could estimate $\widehat{\boldsymbol{y}}_{t+h|t}$ based exclusively on linear projection arguments. Indeed, it is possible to estimate the h-step-ahead regression

$$\boldsymbol{y}_{t+h} = W_h \boldsymbol{x}_t + B_h \boldsymbol{v}_{t+h:t} + \boldsymbol{e}_{t+h}$$

for $h \geq 1$, where $\boldsymbol{v}_{t+h:t}$ is a vector of additional control variables (regressors) and $\boldsymbol{e}_{t+h}$ is a noise vector independent of states and of $\boldsymbol{v}_{t+h:t}$. In general, $\boldsymbol{v}_{t+h:t}$ can consist, for example, of specific lags of (a subset of) regressors, or even other series observed at the same frequency as the target. For clarity, we refer to this approach as direct (one- or multistep) forecasting. The regression coefficients W_h and B_h can be estimated via least squares or ridge regression in a similar fashion as the usual ESN readout weights are. In the simple case where $\boldsymbol{v}_{t+h:t}$ is zeros, we get the forecast

$$\widetilde{\boldsymbol{y}}_{t+h|t} = \widehat{W}_h \boldsymbol{x}_t. \tag{3.18}$$

The validity of this direct forecast approximation builds upon an assumption of linearity of $\mathbb{E}_\theta\left[\boldsymbol{y}_{t+h}|\boldsymbol{x}_t\right]$. The direct forecast approximation is more restrictive than the iterative one in (3.16). We emphasize that (3.17) produces a local approximation at each step $1 \leq j \leq h$, while all $\widetilde{\boldsymbol{y}}_{t+h|t}$ in (3.18) are derived via linearizing at $\boldsymbol{x}_t$. The literature on various models with the underlying linearity assumption is rich. Although one might compare the performance of the direct multistep forecasting ESN strategy to a plethora of existing linear models, we do not find it interesting to constrain intrinsically nonlinear ESN models to a linear regime. For instance, one could explore the Local Projections (LP) estimation of Jordà (2005), which builds on similar ideas. However, in this paper, we aim to assess the added value offered by the nonlinear state map. Therefore, in the following sections and in our empirical study, we adopt only the strategy based on the iterative ESN forecasting of the input series.

3.4 Multi-Frequency Echo State Models

In this subsection, we construct a broad class of ESN models that can accommodate input and target time series sampled at distinct sampling frequencies. We call this family of reservoir models the *Multi-Frequency Echo State Networks* (MFESNs). We extend the prediction strategy discussed in Section 3.3 to the case of input series with mixed frequencies for each of the presented MFESNs. In particular, we show that whenever the forecasted variable of interest is observed at most as frequently as other covariates-series, the proposed reservoir models allow accurate multistep prediction that is superior to the existing benchmarks. The state-space structure of MFESNs is naturally amenable to multi-frequency settings, which is also a virtue of (dynamic) factor models popular in the macroeconomics literature. We emphasize that, unlike other state-space prescriptions, the tuples $(A, C, \boldsymbol{\zeta})$ of matrix parameters of the state maps of ESN-based models are randomly generated and hence do not require estimation. Provided that often the available data is scarce while the dimensions of state spaces need to be sufficiently large to account for the complex dynamics of multiple input time series, the fact that the inner weights of ESNs as recurrent neural networks are randomly generated allows combating numerous computational issues. This offers additional advantages of our proposed approach compared to other benchmark competitors as we see later.

We present two groups of MFESN architectures. The first family is based on a single echo state network architecture and we call these models *Single-Reservoir Multi-Frequency Echo State Networks* (S-MFESNs). The second group allows for as many state equations as the number of distinct sampling frequencies are present in the data *Multi-Reservoir Multi-Frequency Echo State Networks* (M-MFESNs). We defer the empirical comparative analysis of these groups of models to Section 5 where use them in a US GDP multistep forecasting exercise.

3.4.1 Single-Reservoir MFESN

We propose several variants of the single reservoir ESN architectures applicable in the multi-frequency setting. To that goal, we formulate a single state equation with a time index that runs at the highest sampling frequency. We start by recalling that in the temporal notation we introduced in Section 2.1 in Definition 2.1, we let t be the reference frequency, which shall also be the frequency of the target variable. All other frequencies including the highest among all considered time series will be measured using this reference frequency.

Consider L collections of different time series. We assume that each l-th group, $l \in [L]$, consists of n_l time series that are sampled at a common frequency κ_l and contain observations $(\boldsymbol{z}^{(l)}_{t,s|\kappa_l})_{t,s}$ with $\boldsymbol{z}^{(l)}_{t,s|\kappa_l} \in \mathbb{R}^{n_l}$ for all $t \in \mathbb{Z}$ and $s \in \{0, \ldots, \kappa_l - 1\}$. Let $\kappa_{\max} = \max_l \kappa_l$ be the highest sampling frequency among L time series groups and let $q_l := \kappa_{\max}/\kappa_l$ indicate how slower is the κ_l sampling frequency with respect to the highest one. We stack together and repeat the observations in a way that is consistent with the high-frequency index and define

$$\boldsymbol{z}_{t,s|\kappa_{\max}} := \left(\boldsymbol{z}^{(1)\top}_{t-1,\kappa_1-1+\lfloor (s+1)/q_1 \rfloor|\kappa_1}, \boldsymbol{z}^{(2)\top}_{t-1,\kappa_2-1+\lfloor (s+1)/q_2 \rfloor|\kappa_2}, \ldots, \boldsymbol{z}^{(L)\top}_{t-1,\kappa_L-1+\lfloor (s+1)/q_L \rfloor|\kappa_L} \right)^{\top} \in \mathbb{R}^{\sum_{l=1}^{L} n_l},$$

Now it is possible to write a single high-frequency ESN as

$$\boldsymbol{x}_{t,s|\kappa_{\max}} = \alpha \boldsymbol{x}_{t,s-1|\kappa_{\max}} + (1-\alpha)\sigma(A\boldsymbol{x}_{t,s-1|\kappa_{\max}} + C\boldsymbol{z}_{t,s|\kappa_{\max}} + \boldsymbol{\zeta}), \tag{3.19}$$

$$\boldsymbol{z}_{t,s+1|\kappa_{\max}} = \mathcal{W}^{\top} \boldsymbol{x}_{t,s|\kappa_{\max}} + \boldsymbol{u}_{t,s+1|\kappa_{\max}}, \tag{3.20}$$

where $\mathcal{W} \in \mathbb{M}_{N,\sum_{l=1}^{L} n_l}$.

We term this class of MFESN models the *Single-Reservoir Multi-Frequency ESNs* (S-MFESNs).

Aligned and stacked S-MFESN. Multiple choices of the observation equation prescription are possible depending on the learning task of interest. Provided that we aim at forecasting target variables, the following two architecture choices seem the most natural:

- An *aligned* S-MFESN assumes that the most recent state – with respect to the reference time index t – is sufficient to model the process dynamics. More precisely, in order to obtain target $\boldsymbol{y}_{t+1} \in \mathbb{R}^J$, the state equation of S-MFESN is iterated $\kappa_{\max}$ times until the state $\boldsymbol{x}_{t-1,\kappa_{\max}|\kappa_{\max}} = \boldsymbol{x}_{t,0|\kappa_{\max}}$ is observed and used in the following observation equation

  $$\boldsymbol{y}_{t+1} = W^\top \boldsymbol{x}_{t,0|\kappa_{\max}} + \boldsymbol{\epsilon}_{t+1}, \tag{3.21}$$

  where $W \in \mathbb{M}_{N,J}$.

  Let $\widehat{W}$ and $\widehat{\mathcal{W}}$ be the estimates of the readout matrices based on a sample of length T. Then the high-frequency autonomous state transition map is given by

  $$F_{\kappa_{\max}}(\boldsymbol{x}_{t,s-1|\kappa_{\max}}) := \alpha \boldsymbol{x}_{t,s-1|\kappa_{\max}} + (1-\alpha)\sigma\left((A + C\widehat{\mathcal{W}}^\top)\boldsymbol{x}_{t,s-1|\kappa_{\max}} + \boldsymbol{\zeta}\right), \tag{3.22}$$

  which, composed with itself exactly $\kappa_{\max}$ times, yields the target-frequency-aligned autonomous state transition map

  $$F(\boldsymbol{x}_{t,0|\kappa_{\max}}) := \underbrace{F_{\kappa_{\max}} \circ F_{\kappa_{\max}} \cdots \circ F_{\kappa_{\max}}}_{\kappa_{\max} \text{ times}}(\boldsymbol{x}_{t,0|\kappa_{\max}}). \tag{3.23}$$

  Finally, from (3.16) the h-steps ahead forecasts can be computed as

  $$\widetilde{y}_{T+h|T} = \widehat{W}^\top \big(\underbrace{F^{(\mathtt{m},\mathtt{d})} \circ F^{(\mathtt{m},\mathtt{d})} \circ \cdots \circ F^{(\mathtt{m},\mathtt{d})}}_{h-1 \text{ times}} (\boldsymbol{x}^{(\mathtt{m},\mathtt{d})}_{T,0|72})\big). \tag{3.24}$$

- A *stacked* S-MFESN extends the aligned S-MFESN by taking into account $p \in \mathbb{N}$ high-frequency state lags in the observation equation yielding

  $$\boldsymbol{y}_{t+1} = W^\top \begin{pmatrix} \boldsymbol{x}_{t,0|\kappa_{\max}} \\ \vdots \\ \boldsymbol{x}_{t,-p|\kappa_{\max}} \end{pmatrix} + \boldsymbol{\epsilon}_{t+1},$$

  where $W \in \mathbb{M}_{N(p+1),J}$. This method offers more flexibility though the number of parameters that need to be estimated grows linearly with p.

In this paper we are not interested in model selection and hence apply only the alignment approach in our practical exercises.

Example 3.1 (Forecasting with aligned S-MFESN) Suppose we want to use an aligned S-MFESN model to forecast a quarterly one-dimensional target (y_t) using $n_{(\mathtt{m})}$ monthly and $n_{(\mathtt{d})}$ daily series, $(\boldsymbol{z}^{(\mathtt{m})}_{t,s|\kappa_1})$ and $(\boldsymbol{z}^{(\mathtt{d})}_{t,s|\kappa_2})$, respectively. We adopt the assumption that daily data is released 24 days over each calendar month and hence $\kappa_1 = 3$, $\kappa_2 = 72$ and $\kappa_{\max} = 72$, while $q_1 = 24$ and $q_2 = 1$. We want to repeat the most recent monthly observation, as the unified monthly-daily state equation is run at high frequency.

Let $t, *|72$ be the temporal index with quarterly reference. The input vector for the S-MFESN state equation is given by

$$\boldsymbol{z}^{(\mathtt{m},\mathtt{d})}_{t,s|72} := \left({\boldsymbol{z}^{(\mathtt{m})}_{t-1,2+\lfloor (s+1)/24 \rfloor|3}}^\top, {\boldsymbol{z}^{(\mathtt{d})}_{t,s|72}}^\top \right)^\top \in \mathbb{R}^{n_{(\mathtt{m})}+n_{(\mathtt{d})}}.$$

Let the dimension of the state space is N. Then the complete model is written as

$$\boldsymbol{x}^{(\mathtt{m},\mathtt{d})}_{t,s|72} = \alpha \boldsymbol{x}^{(\mathtt{m},\mathtt{d})}_{t,s-1|72} + (1-\alpha)\sigma(A\boldsymbol{x}^{(\mathtt{m},\mathtt{d})}_{t,s-1|72} + C\boldsymbol{z}^{(\mathtt{m},\mathtt{d})}_{t,s|72} + \boldsymbol{\zeta}), \tag{3.25}$$

$$\boldsymbol{z}^{(\mathtt{m},\mathtt{d})}_{t,s+1|72} = \mathcal{W}^\top \boldsymbol{x}^{(\mathtt{m},\mathtt{d})}_{t,s|72} + \boldsymbol{u}_{t,s+1|72}, \tag{3.26}$$

$$y_{t+1} = W^\top \boldsymbol{x}^{(\mathtt{m},\mathtt{d})}_{t} + \boldsymbol{\epsilon}_{t+1}, \tag{3.27}$$

where we notice that (3.25) and (3.26) are run in their own temporal index s and only whenever the state $\boldsymbol{x}_{t-1,\kappa_{\max}|\kappa_{\max}} = \boldsymbol{x}_{t,0|\kappa_{\max}}$ is observed it is then used in the observation equation. Here, the coefficient matrices $\mathcal{W} \in \mathbb{M}_{N,n_{(\mathtt{m})}+n_{(\mathtt{d})}}$ in (3.26) and $W \in \mathbb{R}^N$ in (3.27) can be easily estimated via ridge regression.

We assume, as usual, that the sample available for estimation ends at the reference time index T. From (3.22) the high-frequency autonomous state transition map is given by

$$F_{72}^{(\mathtt{m},\mathtt{d})}(\boldsymbol{x}_{t,s-1|72}^{(\mathtt{m},\mathtt{d})}) := \alpha \boldsymbol{x}_{t,s-1|72}^{(\mathtt{m},\mathtt{d})} + (1-\alpha)\sigma\left((A + C\widehat{\mathcal{W}}^\top)\boldsymbol{x}_{t,s-1|72}^{(\mathtt{m},\mathtt{d})} + \boldsymbol{\zeta}\right),$$

which, composed with itself exactly 72 times, by (3.23) yields the target-frequency-aligned autonomous state transition map

$$F^{(\mathtt{m},\mathtt{d})}(\boldsymbol{x}_{t,0|72}^{(\mathtt{m},\mathtt{d})}) := \underbrace{F_{72}^{(\mathtt{m},\mathtt{d})} \circ F_{72}^{(\mathtt{m},\mathtt{d})} \cdots \circ F_{72}^{(\mathtt{m},\mathtt{d})}}_{72 \text{ times}}(\boldsymbol{x}_{t,0|72}^{(\mathtt{m},\mathtt{d})}).$$

By applying $F^{(\mathtt{m},\mathtt{d})}$ to state $\boldsymbol{x}_{t,0|72}^{(\mathtt{m},\mathtt{d})}$ we iterate the S-MFESN forward in time to provide an estimate for $\boldsymbol{x}_{t+1,0|72}^{(\mathtt{m},\mathtt{d})}$, which can then be linearly projected using $\widehat{W}$ to yield a forecast for y_{t+2}. For the target variable, as well as forecasts, we do not use our temporal notation for the sake of compactness and clarity of exposition. Finally, the multistep forecasts for any $h \geq 1$ can be computed using (3.24) as

$$\widetilde{y}_{T+h|T} = \widehat{W}^\top\big(\underbrace{F^{(\mathtt{m},\mathtt{d})} \circ F^{(\mathtt{m},\mathtt{d})} \circ \cdots \circ F^{(\mathtt{m},\mathtt{d})}}_{h-1 \text{ times}}(\boldsymbol{x}_{T,0|72}^{(\mathtt{m},\mathtt{d})})\big).$$

This example provides an explicit illustration of our forecasting strategy in the case of the application to quarterly GDP forecasting using monthly and daily series.

3.4.2 Multi-Reservoir MFESN

Constructing a MFESN with a single reservoir is not necessarily the most effective modeling strategy. Having more than one reservoir allows a more flexible design of states for different subsets of input variables. For example, suppose quarterly and monthly data are used as regressors. In that case, it could be beneficial to handle series of different frequencies, each with their own reservoir, since economic time series observed with the same frequency have similar time dynamics. Our presentation is general enough to accommodate other types of the partitioning of series into the corresponding reservoir models. We leave it to future research to test other approaches based, for instance, on markets or data types as done in van Huellen et al. (2020).

Assume again L series with input observations $(\boldsymbol{z}_{t,s|\kappa_l}^{(l)})_{t,s}$, $l \in [L]$, with $\boldsymbol{z}_{t,s|\kappa_l}^{(l)} \in \mathbb{R}^{n_l}$ for all $t \in \mathbb{Z}$ and $s \in \{0, \ldots, \kappa_l - 1\}$ sampled at common frequencies $\{\kappa_1, \ldots, \kappa_L\}$, respectively. For each of the L input series we define the corresponding *ESN* model as

$$\boldsymbol{x}_{t,s|\kappa_l}^{(l)} = \alpha_l \boldsymbol{x}_{t,s-1|\kappa_l}^{(l)} + (1-\alpha_l)\sigma(A_l \boldsymbol{x}_{t,s-1|\kappa_l}^{(l)} + C_l \boldsymbol{z}_{t,s|\kappa_l}^{(l)} + \boldsymbol{\zeta}_l), \tag{3.28}$$

$$\boldsymbol{z}_{t,s+1|\kappa_l}^{(l)} = \mathcal{W}_l^\top \boldsymbol{x}_{t,s|\kappa_l}^{(l)} + \boldsymbol{u}_{t,s+1|\kappa_l}^{(l)}, \quad l \in [L], \tag{3.29}$$

with $\mathcal{W}_l \in \mathbb{M}_{N_l,n_l}$ with N_l the dimension of the state space. Notice that the time index s is different for each l according to our temporal notation introduced in Definition 2.1 and each state equation runs at its own frequency. The dimensions $\{N_1, N_2, \ldots, N_L\}$ of the state space can be chosen for the L reservoir models individually. Additionally, multiple reservoirs have the associated hyperparameter tuples $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_L\}$ to be tuned. This requires some care whenever one wants to optimize all hyperparameters jointly. Since there are L reservoir state equations, we call this class of MFESN models *Multi-Reservoir Multi-Frequency ESN* (M-MFESN).

Similar to our approach for S-MFESN, the state equations are iterated each κ_l times until the states $\boldsymbol{x}_{t-1,\kappa_l|\kappa_l}^{(l)} = \boldsymbol{x}_{t,0|\kappa_l}^{(l)}$ are obtained and we can formulate the *aligned* M-MFESN observation equation as

$$\boldsymbol{y}_{t+1} = W^\top \begin{pmatrix} \boldsymbol{x}_{t,0|\kappa_1}^{(1)} \\ \vdots \\ \boldsymbol{x}_{t,0|\kappa_L}^{(L)} \end{pmatrix} + \boldsymbol{\epsilon}_{t+1}, \tag{3.30}$$

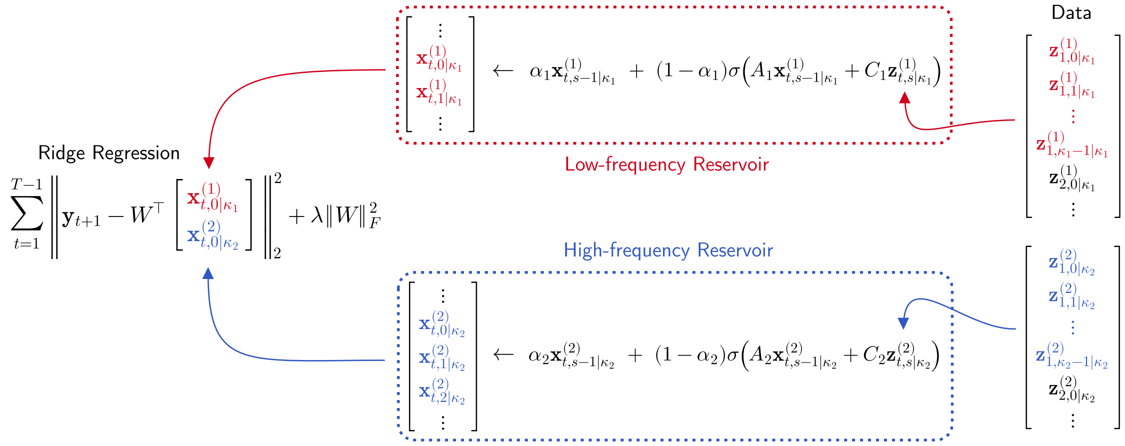

Data
Ridge Regression
$\sum_{t=1}^{T-1} \left\| \mathbf{y}_{t+1} - W^\top \begin{bmatrix} \mathbf{x}^{(1)}_{t,0|\kappa_1} \\ \mathbf{x}^{(2)}_{t,0|\kappa_2} \end{bmatrix} \right\|_2^2 + \lambda \|W\|_F^2$
$\leftarrow \alpha_1 \mathbf{x}^{(1)}_{t,s-1|\kappa_1} + (1-\alpha_1)\sigma\big(A_1 \mathbf{x}^{(1)}_{t,s-1|\kappa_1} + C_1 \mathbf{z}^{(1)}_{t,s|\kappa_1}\big)$
Low-frequency Reservoir
High-frequency Reservoir
$\leftarrow \alpha_2 \mathbf{x}^{(2)}_{t,s-1|\kappa_2} + (1-\alpha_2)\sigma\big(A_2 \mathbf{x}^{(2)}_{t,s-1|\kappa_2} + C_2 \mathbf{z}^{(2)}_{t,s|\kappa_2}\big)$
$\mathbf{x}^{(1)}_{t,0|\kappa_1}$, $\mathbf{x}^{(1)}_{t,1|\kappa_1}$
$\mathbf{x}^{(2)}_{t,0|\kappa_2}$, $\mathbf{x}^{(2)}_{t,1|\kappa_2}$, $\mathbf{x}^{(2)}_{t,2|\kappa_2}$
$\mathbf{z}^{(1)}_{1,0|\kappa_1}$, $\mathbf{z}^{(1)}_{1,1|\kappa_1}$, $\mathbf{z}^{(1)}_{1,\kappa_1-1|\kappa_1}$, $\mathbf{z}^{(1)}_{2,0|\kappa_1}$
$\mathbf{z}^{(2)}_{1,0|\kappa_2}$, $\mathbf{z}^{(2)}_{1,1|\kappa_2}$, $\mathbf{z}^{(2)}_{1,\kappa_2-1|\kappa_2}$, $\mathbf{z}^{(2)}_{2,0|\kappa_2}$

Figure 2: Scheme of a Multi-Reservoir MFESN (M-MFESN) model combining input data sampled at two frequencies with state alignment and estimation for one-step ahead forecasting of the target series.

where $W \in \mathbb{M}_{\sum_{l=1}^{L} N_l, J}$. The schematic diagram of the aligned M-MFESN for the case of two frequencies of regressor time series is shown in Figure 2.

Let $\widehat{W}$ and $\widehat{\mathcal{W}}_l$, $l \in [L]$ be the estimates of the readout matrices based on the T-long sample of observations. Then for any $l \in [L]$ the κ_l-frequency autonomous state transition map is given by

$$F^{(l)}_{\kappa_l}(\boldsymbol{x}^{(l)}_{t,s-1|\kappa_l}) := \alpha_l \boldsymbol{x}^{(l)}_{t,s-1|\kappa_l} + (1-\alpha)\sigma\left((A_l + C_l \widehat{\mathcal{W}}_l^\top)\boldsymbol{x}^{(l)}_{t,s-1|\kappa_l} + \boldsymbol{\zeta}_l\right). \tag{3.31}$$

The target-frequency-aligned autonomous state transition map associated to each frequency l is hence defined as

$$F^{(l)}(\boldsymbol{x}_{t,0|\kappa_{\max}}) := \underbrace{F^{(l)}_{\kappa_l} \circ F^{(l)}_{\kappa_l} \cdots \circ F^{(l)}_{\kappa_l}}_{\kappa_l \text{ times}}(\boldsymbol{x}^{(l)}_{t,0|\kappa_l}). \tag{3.32}$$

Finally, from (3.16) the h-steps ahead forecasts can be computed as

$$\widetilde{y}_{T+h|T} = \widehat{W}^\top \begin{pmatrix} \underbrace{F^{(1)} \circ F^{(1)} \circ \cdots \circ F^{(1)}}_{h-1 \text{ times}}(\boldsymbol{x}^{(1)}_{T,0|\kappa_1}) \\ \vdots \\ \underbrace{F^{(L)} \circ F^{(L)} \circ \cdots \circ F^{(L)}}_{h-1 \text{ times}}(\boldsymbol{x}^{(L)}_{T,0|\kappa_L}) \end{pmatrix}. \tag{3.33}$$

Similar to the stacked S-MFESN architecture, defining the *stacked* M-MFESN is straightforward, even though the notation is burdensome. We emphasize that for a stacked M-MFESN model, it is also necessary to choose L state lags, $\{p_1, \dots, p_L\}$, one for each state equation. The proliferation of model hyperparameters due to the additional lag specification required by a stacked M-MFESN can seriously complicate tuning for this class of models, especially when many distinct frequencies are modeled. In our empirical study, we use only the aligned M-MFESN architecture even though we see the stacked M-MFESNs as an interesting avenue for further research.

Example 3.2 (Aligned M-MFESN Forecasting) Similar to Example 3.1, we aim to forecast a quarterly target with monthly and daily series, but this time we use a M-MFESN model. We have to define two independent state equations, one for monthly and one for daily series; in the observation equations, two states must be aligned temporally and stacked to form the full set of regressors. The data consists again of quarterly (y_t), $n_{(\mathtt{m})}$ monthly series $(\boldsymbol{z}^{(\mathtt{m})}_{t,s|3})$ and $n_{(\mathtt{d})}$ daily series $(\boldsymbol{z}^{(\mathtt{d})}_{t,s|72})$.

The aligned M-MFESN model with two reservoirs of dimensions $N_{(\mathtt{m})}$ and $N_{(\mathtt{d})}$, respectively, is given by

$$\boldsymbol{x}^{(\mathtt{m})}_{t,s|3} = \alpha_1 \boldsymbol{x}^{(\mathtt{m})}_{t,s-1|3} + (1-\alpha_1)\sigma(A_1 \boldsymbol{x}^{(\mathtt{m})}_{t,s-1|3} + C_1 \boldsymbol{z}^{(\mathtt{m})}_{t,s|3} + \boldsymbol{\zeta}_1), \tag{3.34}$$

$$\boldsymbol{z}^{(\mathtt{m})}_{t,s+1|3} = \mathcal{W}^{\top}_{(\mathtt{m})} \boldsymbol{x}^{(\mathtt{m})}_{t,s|3} + \boldsymbol{u}^{(\mathtt{m})}_{t,s+1|3}, \tag{3.35}$$

$$\boldsymbol{x}^{(\mathtt{d})}_{t,s|72} = \alpha_2 \boldsymbol{x}^{(\mathtt{d})}_{t,s-1|72} + (1-\alpha_2)\sigma(A_2 \boldsymbol{x}^{(\mathtt{d})}_{t,s-1|72} + C_2 \boldsymbol{z}^{(\mathtt{d})}_{t,s|72} + \boldsymbol{\zeta}_2), \tag{3.36}$$

$$\boldsymbol{z}^{(\mathtt{d})}_{t,s+1|72} = \mathcal{W}^{\top}_{(\mathtt{d})} \boldsymbol{x}^{(\mathtt{d})}_{t,s|72} + \boldsymbol{u}^{(\mathtt{d})}_{t,s+1|72}, \tag{3.37}$$

$$y_{t+1} = W^{\top} \begin{pmatrix} \boldsymbol{x}^{(\mathtt{m})}_{t,0|3} \\ \boldsymbol{x}^{(\mathtt{d})}_{t,0|72} \end{pmatrix} + \boldsymbol{\epsilon}_{t+1}, \tag{3.38}$$

where $\mathcal{W}_{(\mathtt{m})} \in \mathbb{M}_{N_{(\mathtt{m})},n_{(\mathtt{m})}}$, $\mathcal{W}_{(\mathtt{d})} \in \mathbb{M}_{N_{(\mathtt{d})},n_{(\mathtt{d})}}$ and $W \in \mathbb{R}^{N_{(\mathtt{m})}+N_{(\mathtt{d})}}$. Here, the monthly reservoir ($\boldsymbol{x}^{(\mathtt{m})}_{t,s|3}$) has the temporal index of frequency 3, while the daily reservoir ($\boldsymbol{x}^{(\mathtt{d})}_{t,s|72}$) of 72; the high-frequency index s is different for the two models. Notice that in an M-MFESN model it is necessary to introduce L additional observation equations for the states, one for each reservoir/frequency (in this case these observation equations are (3.35) and (3.37)). Notice that the state equations are iterated each κ_l times to collect the states to be aligned in the observation equation (3.38). Again, the sample-based estimates of coefficient matrices $\widehat{\mathcal{W}}_{(\mathtt{m})}$, $\widehat{\mathcal{W}}_{(\mathtt{d})}$ and $\widehat{W}$ in (3.35), (3.36), and in (3.38), respectively, can be obtained via the ridge regression.

Exactly as in Example 3.1, using (3.31) we can introduce high-frequency autonomous state maps $F^{(\mathtt{m})}_3$ and $F^{(\mathtt{d})}_{72}$ as

$$F^{(\mathtt{m})}_3(\boldsymbol{x}^{(\mathtt{m})}_{t,s-1|3}) := \alpha_1 \boldsymbol{x}^{(\mathtt{m})}_{t,s-1|3} + (1-\alpha_1)\sigma\left((A_1 + C_1 \widehat{\mathcal{W}}^{\top}_{(\mathtt{m})})\boldsymbol{x}^{(\mathtt{m})}_{t,s-1|3} + \boldsymbol{\zeta}_1\right),$$

$$F^{(\mathtt{d})}_{72}(\boldsymbol{x}^{(\mathtt{d})}_{t,s-1|72}) := \alpha_2 \boldsymbol{x}^{(\mathtt{d})}_{t,s-1|72} + (1-\alpha_2)\sigma\left((A_2 + C_2 \widehat{\mathcal{W}}^{\top}_{(\mathtt{d})})\boldsymbol{x}^{(\mathtt{d})}_{t,s-1|72} + \boldsymbol{\zeta}_2\right),$$

as well as their target-frequency aligned counterparts $F^{(\mathtt{m})}$ and $F^{(\mathtt{d})}$, respectively, by (3.32) given by

$$F^{(\mathtt{m})}(\boldsymbol{x}^{(\mathtt{m},\mathtt{d})}_{t,0|3}) := \underbrace{F^{(\mathtt{m})}_3 \circ F^{(\mathtt{m})}_3 \cdots \circ F^{(\mathtt{m})}_3}_{3 \text{ times}}(\boldsymbol{x}^{(\mathtt{m},\mathtt{d})}_{t,0|3}),$$

$$F^{(\mathtt{d})}(\boldsymbol{x}^{(\mathtt{m},\mathtt{d})}_{t,0|72}) := \underbrace{F^{(\mathtt{d})}_{72} \circ F^{(\mathtt{d})}_{72} \cdots \circ F^{(\mathtt{d})}_{72}}_{72 \text{ times}}(\boldsymbol{x}^{(\mathtt{d})}_{t,0|72}).$$

The h-step ahead forecasts can be computed using the approximation in (3.33) as

$$\widetilde{y}_{T+h|T} = \widehat{W}^{\top} \left(\begin{array}{c} \underbrace{F^{(\mathtt{m})} \circ F^{(\mathtt{m})} \circ \cdots \circ F^{(\mathtt{m})}}_{h-1 \text{ times}}(\boldsymbol{x}^{(\mathtt{m})}_{T,0|3}) \\ \underbrace{F^{(\mathtt{d})} \circ F^{(\mathtt{d})} \circ \cdots \circ F^{(\mathtt{d})}}_{h-1 \text{ times}}(\boldsymbol{x}^{(\mathtt{d})}_{T,0|72}) \end{array} \right).$$

In this case, it is important to note that while both $F^{(\mathtt{m})}$ and $F^{(\mathtt{d})}$ are composed $h-1$ times at step h, the underlying number of autonomous reservoir iterations is different for the monthly and daily reservoirs, namely 3 and 72, and depends on their own frequencies. This also suggests that one should take into account the different time dynamics when, for example, tuning M-MFESN hyperparameters $\boldsymbol{\varphi}^{(\mathtt{m})}$ and $\boldsymbol{\varphi}^{(\mathtt{d})}$.

4 Benchmark Models

This section gives an overview of two popular methods that can accommodate time series with different sampling frequencies: The MIDAS (MIxed DAta Sampling) framework and the mixed-frequency dynamic factor model (DFM). A new aggregation scheme with Almon exponential structure is proposed to bridge MIDAS and DFM for our forecasting comparison. Both models serve as a baseline for forecasting comparison against our proposed MFESN model.

4.1 MIDAS

A state-of-the-art methodology for incorporating data of heterogeneous frequencies into one model is the MIDAS framework developed in Ghysels et al. (2004, 2007). Here we present MIDAS in its dynamic form, which allows the inclusion of target series autoregressive lags. We use our temporal notation given in Definition 2.1 throughout.

If the MIDAS model contains only one explanatory variable (x_r) with frequency multiplier κ, then it can be written as

$$y_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i} + \beta \sum_{k=0}^{K} \varphi(\boldsymbol{\theta}, k) x_{t,-k|\kappa} + \epsilon_t, \tag{4.1}$$

where α_0 is a constant term, $\{\alpha_i\}_{i=1}^{p}$ are the autoregressive parameters, β is a scaling parameter, $\{\varphi(\boldsymbol{\theta}, k)\}_{k=0}^{K}$ are the MIDAS weights given as a parametric function of lag k and underlying parameter vector $\boldsymbol{\theta} \in \mathbb{R}^q$, and (ϵ_t) is a martingale difference process relative to the filtration $\{\mathcal{F}_t\}$ generated by $\{y_{t-1-j}, x_{t-j}, \ldots, x_{t-j,-K|\kappa}, \epsilon_{t-1-j} \mid j \geq 0\}$ and such that $\mathbb{E}[\epsilon_t^2] = \sigma_\epsilon^2 < \infty$.

The MIDAS weighting scheme is the core innovation of the model. It borrows parsimony from distributed lag models in the sense that, even if K is large, the vector $\boldsymbol{\theta} \in \mathbb{R}^q$ is usually restricted to contain only a handful of parameters. This greatly reduces the number of coefficients that need to be estimated, and a nonlinear least-squares estimator $\widehat{\boldsymbol{\theta}}$ can be readily implemented. There are alternative formulations of the MIDAS framework where $\varphi(\boldsymbol{\theta}, k) = \theta_k$ so that the above reduces to a full linear model, the so-called unrestricted MIDAS or U-MIDAS (Foroni and Marcellino, 2011). We follow the literature and use the most commonly applied weighting scheme that is based on the exponential Almon weighting polynomial map $\varphi : \mathbb{R}^q \times \mathbb{N}^+ \longrightarrow \mathbb{R}^+$ (see Almon (1965) for more details). In particular, for the case of $q = 2$, the two-parameter Almon weighting polynomial is given by

$$\varphi(\boldsymbol{\theta}, k) = \varphi((\theta_1, \theta_2), k) = \exp(\theta_1 k + \theta_2 k^2), \quad k \in \mathbb{N}^+. \tag{4.2}$$

Since Almon weights need not sum up to a given constant for different values of θ_1 and θ_2, it is often common to consider the normalized Almon scheme

$$\overline{\varphi}(\boldsymbol{\theta}, k) = \frac{\exp(\theta_1 k + \theta_2 k^2)}{\sum_{k=0}^{K} \exp(\theta_1 k + \theta_2 k^2)}, \tag{4.3}$$

which together with (4.1) allows to treat β as a rescaling constant.

Let us now consider a more general model suitable for situations where time series of different frequencies are available and must be integrated into the MIDAS equation. Let $\{x_{r_1}^{(1)}, x_{r_2}^{(2)}, \ldots, x_{r_L}^{(L)}\}$ be the set of L regressors with their corresponding frequencies $\{\kappa_1, \kappa_2, \ldots, \kappa_L\}$, respectively. We also require that $\kappa_l \in \mathbb{N}^+$, $l \in [L]$. Additionally, it happens frequently in practice that κ_l, $l \in [L]$ takes values from a small set of integers. For example, in the case of yearly, quarterly, and monthly data $\kappa_l \in \{1, 4, 12\}$ even though L could be very large (often, hundreds or thousands of series might be of interest). The MIDAS model explaining low-frequency target variable y_t with L regressors $\{x_{r_l}^{(l)}\}_{l=1}^{L}$ can be written as follows

$$y_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{l=1}^{L} \beta_l \sum_{k_l=0}^{K_l} \varphi(\boldsymbol{\theta}_l, k_l) x_{t,-k_l|\kappa_l}^{(l)} + \epsilon_t, \tag{4.4}$$

where the martingale difference process (ϵ_t) is relative to the filtration generated by sets as in (4.1), modified to include all the L regressors considered.

The MIDAS framework produces forecasts of the chosen target variable at the low frequency of the target. Yet, due to the MIDAS multi-frequency structure, *nowcasting* is also a straightforward exercise:

if, for example, the high-frequency regressor is a single series (x_r) with frequency multiplier κ, one can construct exactly κ regression equations – one for each high-frequency release within a low-frequency period – and use these to produce high-frequency nowcasts of the target. In fact, due to the convenience of the MIDAS model, it is easy to define high-frequency regression specifications to study high-frequency forecasts and multicasts.

In practice, implementing (4.4) demands some care. From a computational point of view, as long as the relevant regression matrices can be constructed, estimation amounts to a nonlinear least-squares problem, which can be readily solved. In Appendix 7.2 and Appendix 7.5.1 we discuss the technical aspects of our MIDAS implementation in more detail. One of the important issues of the MIDAS estimation is the non-convexity of the nonlinear least squares loss as a function of parameters. Often, a practitioner may obtain different estimation results depending on initialization and, more importantly, those that lead to a different quality of forecasts. Another crucial disadvantage of the MIDAS specification is that practical implementations can be very challenging. This is caused mainly by the ragged edges of the "raw" macroeconomic data, incomplete observations, and uneven sampling frequencies. The relative inflexibility of MIDAS regression lag specifications makes integrating daily and weekly data at true calendar frequencies (i.e. without interpolation or aggregation) very complex. State-space models effectively mitigate these issues. Therefore we continue our exposition by presenting a state-space benchmark that is very popular in the macroeconomic forecasting literature, the so-called dynamic factor model (DFM).

4.2 Mixed-frequency DFM

Macroeconomic modeling based on dynamic factor models (DFMs) has been popular since their introduction in Geweke (1977) and Sargent et al. (1977). The proposition of DFMs is that a low-dimensional latent factor $(\boldsymbol{f}_t)_{t\in\mathbb{Z}}$, $\boldsymbol{f}_t \in \mathbb{R}^d$, drives a high-dimensional observable stochastic process $(\boldsymbol{y}_t)_{t\in\mathbb{Z}}$, $\boldsymbol{y}_t \in \mathbb{R}^n$. We consider a time-inhomogeneous state-space model with dynamics

$$\boldsymbol{f}_{t+1}|\boldsymbol{f}_{1:t}, \boldsymbol{y}_{1:t} \sim h_{t+1,\theta}(\cdot|\boldsymbol{f}_t) \tag{4.5}$$
$$\boldsymbol{y}_{t+1}|\boldsymbol{f}_{1:t+1}, \boldsymbol{y}_{0:t} \sim g_{t+1,\theta}(\cdot|\boldsymbol{f}_{t+1}) \tag{4.6}$$

for some time-dependent state transition kernels $h_{t,\theta}$ and observation densities $g_{t,\theta}$ and some parameter θ in a parameter space Θ. A common example in the literature (see Watson and Engle (1983) for more details) is linear Gaussian factor models with time-inhomogeneous state transitions that can be represented as

$$\boldsymbol{f}_{t+1} = A_\theta \boldsymbol{f}_t + R_\theta \boldsymbol{u}_t \tag{4.7}$$
$$\boldsymbol{y}_{t+1} = \Lambda_{t+1,\theta} \boldsymbol{f}_{t+1} + S_{t+1,\theta} \boldsymbol{w}_{t+1} \tag{4.8}$$

with state transition matrix $A_\theta \in \mathbb{M}_d$, time-dependent factor loading matrices $\Lambda_t \in \mathbb{M}_{n,d}$, $\boldsymbol{u}_t$ and $\boldsymbol{w}_t$ are independent Gaussian vectors with zero mean and identity covariance matrix of dimension p and n, respectively, and matrices R_θ and $S_{t,\theta}$ of appropriate dimensions. It is often assumed that the dimension p of the state noise vector $\boldsymbol{u}_t$ is smaller than the latent state space dimension d, which implies that $R_\theta R_\theta^\top$ is rank deficient, such as for AR(p) factor dynamics (Stock and Watson, 2016, Forni et al., 2005, Doz et al., 2011) so that $d = kp$ for some $k \in \mathbb{N}^+$,

$$A_\theta = \begin{pmatrix} A_\theta^{(1)} & A_\theta^{(2)} & \cdots & A_\theta^{(p-1)} & A_\theta^{(p)} \\ \mathbb{I}_k & \mathbb{O}_k & \cdots & \mathbb{O}_k & \mathbb{O}_k \\ \mathbb{O}_k & \mathbb{I}_k & \cdots & \mathbb{O}_k & \mathbb{O}_k \\ \vdots & & \ddots & & \vdots \\ \mathbb{O}_k & \mathbb{O}_k & \cdots & \mathbb{I}_k & \mathbb{O}_k \end{pmatrix}, \quad \Lambda_{t,\theta} = \begin{pmatrix} \Lambda_{t,\theta}^{(1)} & \Lambda_{t,\theta}^{(2)} & \cdots & \Lambda_{t,\theta}^{(p)} \end{pmatrix} \tag{4.9}$$

with $A_\theta^{(j)} \in \mathbb{M}_k$ and $\Lambda_{t,\theta}^{(j)} \in \mathbb{M}_{n,k}$. Setting $\boldsymbol{f}_t = (\boldsymbol{v}_t^\top, \boldsymbol{v}_{t-1}^\top, \ldots, \boldsymbol{v}_{t-p+1}^\top)^\top$ implies that $(\boldsymbol{v}_t)$ is a k-dimensional AR(p) process and it is commonly assumed that $\Lambda_{t,\theta}^{(j)} = \mathbb{O}_{n,k}$ for $j > 1$. Let the initial state

$\boldsymbol{f}_0$ be distributed according to ν. The joint density of the latent path $\boldsymbol{f}_{0:T}$ and observations $\boldsymbol{y}_{0:T}$ is then

$$p_{\theta,\nu}(\boldsymbol{f}_{0:T}, \boldsymbol{y}_{0:T}) = \nu(\boldsymbol{f}_0) g_{0,\theta}(\boldsymbol{y}_0|\boldsymbol{f}_0) \prod_{t=1}^{T} h_{t,\theta}(\boldsymbol{f}_t|\boldsymbol{f}_{t-1}) g_{t,\theta}(\boldsymbol{y}_t|\boldsymbol{f}_t),$$

while the marginal likelihood of $\boldsymbol{y}_{0:T}$ is $p_{\theta,\nu}(\boldsymbol{y}_{0:T}) = \int p_{\theta,\nu}(\boldsymbol{f}_{0:T}, \boldsymbol{y}_{0:T}) \mathrm{d}\boldsymbol{f}_{0:T}$. Popular procedures for learning the static parameters $\theta \in \Theta$ are based on gradient descent of the negative log-likelihood function $\ell_T \colon \Theta \to \mathbb{R}, \theta \mapsto -\log p_{\theta,\nu}(\boldsymbol{y}_{0:T})$ or on the Expectation Maximization (EM) algorithm introduced in Dempster et al. (1977). We consider here gradient descent algorithms[3] based on a sequence of step sizes $\gamma_k > 0$, that update the model parameters based on iterations of the form

$$\theta_{k+1} = \theta_k - \gamma_{k+1} \nabla_\theta \ell_T(\theta)|_{\theta=\theta_k},$$

for $k \in \mathbb{N}^+$.

Assuming a linear Gaussian setting where the transition density of the latent factor process is given by (4.9) to yield an AR(p) process $(\boldsymbol{v}_t)_{t\in\mathbb{Z}}$, $\boldsymbol{v}_t = (v_{1,t}, \dots, v_{k,t})$, there remains some flexibility as to how the linear mappings[4] $\mathsf{Agg}_{\boldsymbol{\theta}} \colon \mathbb{M}_{k,p} \to \mathbb{R}^n$, $(\boldsymbol{v}_{t-p+1} \dots, \boldsymbol{v}_t) \mapsto (\Lambda_{t,\boldsymbol{\theta}} \boldsymbol{f}_t)_i$ are chosen for each dimension $i \in [n]$. We call this linear mapping $\mathsf{Agg}_{\boldsymbol{\theta}}$ an *aggregation function* and consider specific examples below that yield different models for the observation matrices $\Lambda_{t,\boldsymbol{\theta}}$.

Example 4.1 (Stock aggregation) For $i \in [n]$, let $\boldsymbol{\beta}_i = (\beta_{i1}, \dots, \beta_{ik}) \in \mathbb{R}^k$ and consider

$$\mathsf{Agg}^{\mathrm{S}}_{\boldsymbol{\theta}}(\boldsymbol{v}_{t-p+1} \dots, \boldsymbol{v}_t)_i = \sum_{m=1}^{k} \beta_{im} v_{m,t},$$

with $\boldsymbol{\theta} = \boldsymbol{\beta}_i$.

Example 4.2 (Almon-Lag aggregation) For $i \in [n]$, let $\boldsymbol{\beta}_i \in \mathbb{R}^k$, $\boldsymbol{\psi}_i \in \mathbb{R}^{2k}$ and consider

$$\mathsf{Agg}^{\mathrm{AL}}_{\boldsymbol{\theta}}(\boldsymbol{v}_{t-p+1} \dots, \boldsymbol{v}_t)_i = \sum_{m=1}^{k} \beta_{im} \sum_{\ell=0}^{p-1} \overline{\varphi}(\psi_{im}, \ell) v_{m,t-\ell},$$

with $\boldsymbol{\theta} = (\boldsymbol{\beta}_i, \boldsymbol{\psi}_i, \boldsymbol{\beta}_i, \boldsymbol{\psi}_i)$ and Almon-Lag weights $\overline{\varphi}$ given in (4.3).

Example 4.3 (Trigonometric aggregation) For $i \in [n]$, let $\boldsymbol{\beta}_i \in \mathbb{R}^k$, and for $K \in \mathbb{N}$, let $\boldsymbol{\lambda} \in \mathbb{R}^K_+$, $\boldsymbol{\omega} \in [0,1]^K$, $\boldsymbol{\gamma} \in [-\pi, \pi]^K$ and $\tau \in \mathbb{R}_+$. Define

$$\mathsf{Agg}^{\sin}_{\boldsymbol{\theta}}(\boldsymbol{v}_{t-p+1} \dots, \boldsymbol{v}_t)_i = \sum_{m=1}^{k} \beta_{im} \sum_{\ell=0}^{p-1} \overline{a}_p(\boldsymbol{\lambda}, \boldsymbol{\omega}, \boldsymbol{\gamma}, \tau, \ell) v_{m,t-\ell},$$

with $\boldsymbol{\theta} = (\boldsymbol{\beta}_i, \boldsymbol{\lambda}, \boldsymbol{\omega}, \boldsymbol{\gamma}, \tau)$ and

$$\overline{a}_p(\boldsymbol{\lambda}, \boldsymbol{\omega}, \boldsymbol{\gamma}, \tau, \ell) = \frac{\exp\left(\frac{1}{\tau} \sum_{j=1}^{K} \lambda_j^2 \cos(2\pi\omega_j \ell + \gamma_j)\right)}{\sum_{\ell'=0}^{p-1} \exp\left(\frac{1}{\tau} \sum_{j=1}^{K} \lambda_j^2 \cos(2\pi\omega_j \ell' + \gamma_j)\right)}.$$

[3] Consistency of the maximum log-likelihood estimate for the dynamics (4.7)-(4.8) in the time-homogeneous case has been established for instance in Douc et al. (2011) under regularity assumptions, including for instance the full-rank of the noise covariance matrix S_θ, of the controllability matrix $C_\theta = \left(R_\theta | A_\theta R_\theta | \cdots | A_\theta^{d-1} R_\theta\right)$, and of the observability matrix $O_\theta = \left(\Lambda_\theta^\top | (\Lambda_\theta A_\theta)^\top | \cdots | (\Lambda_\theta A_\theta^{d-1})^\top\right)^\top$. It is also possible to consider an online learning setting using a recursive decomposition of the score function as in LeGland and Mevel (1997). For general latent state dynamics (4.5) and observation densities (4.6) that can be non-linear with non-Gaussian noise, particle filtering algorithms are often utilized that make use of particle approximations in gradient-descent or EM learning approaches, see for instance Kantas et al. (2015).

[4] The Markovian representation (4.5)-(4.6), that is, the companion form, is based on the autoregresssive order p, however, one can set $A_{\boldsymbol{\theta}}^{(\ell)} = \mathbb{O}_k$ for some $\ell \leq p$.

This aggregation scheme is motivated by self-attention models (we refer the reader to Bahdanau et al. (2014), Vaswani et al. (2017) for more details), but to retain linearity only considers a relative positional encoding with a Toeplitz structure. Observe that the aggregation parameters are shared across all n dimensions in contrast to the Almon lag scheme in Example 4.2.

Some authors (see for example Mariano and Murasawa (2003), Bańbura and Modugno (2014)) have imposed different restrictions on the form of the emission matrix or aggregation function, particularly for one-dimensional mixed-frequency factor models of quarterly GDP growth rates and monthly covariates, which are motivated by approximations of growth rates. We do not pursue this additional restriction in this work.

The static parameters $\boldsymbol{\theta}$ can be estimated using gradient ascent of the log-likelihood computed using Kalman filtering. For alternative estimation approaches using EM that could be extended to this setting, we refer the reader to Bańbura and Modugno (2014). Bai et al. (2013) discusses connections between mixed-frequency factor models and MIDAS regression. Nonlinear or non-Gaussian dynamic factor models in a mixed frequency setting have been considered in Gagliardini et al. (2017), Leippold and Yang (2019) that rely on particle filtering methods in conjunction with backward simulation algorithms as in Godsill et al. (2004), while Schorfheide et al. (2018) consider a Bayesian approach using particle MCMC (see Andrieu et al. (2010)). Such approaches can become computationally expensive and are not considered for benchmarking purposes.

While previous mixed-frequency DFMs (see Mariano and Murasawa (2003), Bańbura and Modugno (2014) for a more thorough discussion) often consider time series which are sampled at two frequencies, we introduce here a flexible mixed-frequency DFM that describes $L \in \mathbb{N}$ collections of distinct time series sampled at frequencies $\{\kappa_1, \ldots, \kappa_L\}$ and each consisting of $\{n_1, \ldots, n_L\}$ series, respectively. In the same setting as in Subsection 3.4, each group of n_l, $l \in [L]$, time series sampled at frequency κ_l contains observations $(\boldsymbol{y}^{(l)}_{t,s|\kappa_l})$ with $\boldsymbol{y}^{(l)}_{t,s|\kappa_l} \in \mathbb{R}^{n_l}$ for all $t \in \mathbb{Z}$ and $s \in \{0, \ldots, \kappa_l - 1\}$. Let $\kappa_{\max} = \max_l \kappa_l$. Suppose that the latent factor dynamics are updated at the highest sampling frequency based on the linear transition

$$\boldsymbol{f}_{t,s+1|\kappa_{\max}} = A_{\boldsymbol{\theta}} \boldsymbol{f}_{t,s|\kappa_{\max}} + R_{\boldsymbol{\theta}} \boldsymbol{u}_{t,s+1|\kappa_{\max}}, \tag{4.10}$$

where

$$\boldsymbol{f}_{t,s|\kappa_{\max}} = \left(\boldsymbol{v}^{\top}_{t,s|\kappa_{\max}}, \ldots, \boldsymbol{v}^{\top}_{t,s-p+1|\kappa_{\max}} \right)^{\top},$$

with $A_{\boldsymbol{\theta}}$ given in (4.9) for the special case where $A^{(\ell)}_{\boldsymbol{\theta}} = \mathbb{O}_k$ for $\ell \geq 2$, $p = \kappa_{\max}$ and

$$A^{(1)}_{\boldsymbol{\theta}} = \bar{A} \frac{\rho}{\max\left\{\rho, |\lambda_1(\bar{A})|\right\}}$$

with parameters $\rho \in (0,1)$, $\bar{A} \in \mathbb{M}_k$ and with $\lambda_1(\bar{A})$ denoting the largest eigenvalue of $\bar{A}$. In the simplified scenario of first order autoregressive dynamics, we parameterize $R_{\boldsymbol{\theta}} \in \mathbb{M}_k$ to be positive definite and diagonal and $\boldsymbol{u}_{t,s+1|\kappa_{\max}}$ are a sequence of IID k-dimensional standard Gaussian variables.

Notice that Kalman filtering formulas yield the first moment

$$\widehat{\boldsymbol{f}}_{t,s|\kappa_{\max}} = \mathbb{E}\left[\boldsymbol{f}_{t,s|\kappa} \middle| \boldsymbol{y}_{1,0|\kappa_{\max}}, \ldots, \boldsymbol{y}_{t,s|\kappa_{\max}} \right]$$

recursively online, see for example Appendix 7.3 for details in the general time-inhomogeneous case. Due to the linearity in (4.10), for any $h \in \mathbb{N}$,

$$\widehat{\boldsymbol{f}}_{t,s+h|\kappa_{\max}} = \mathbb{E}\left[\boldsymbol{f}_{t,s+h|\kappa_{\max}} \middle| \boldsymbol{y}_{1,0|\kappa_{\max}}, \ldots, \boldsymbol{y}_{t,s|\kappa_{\max}} \right] = A^{h}_{\boldsymbol{\theta}} \widehat{\boldsymbol{f}}_{t,s|\kappa_{\max}}.$$

Furthermore, from the linearity of the emission model, we obtain the forecasts for any $s, h \in \mathbb{N}$ such that $s, h \bmod \kappa_l = 0$,

$$\mathbb{E}\left[\boldsymbol{y}^{(l)}_{t,s+h|\kappa_{\max}} \middle| \boldsymbol{y}_{1,0|\kappa_{\max}}, \ldots, \boldsymbol{y}_{t,s|\kappa_{\max}} \right] = \mathsf{Agg}_{\boldsymbol{\theta}^{(l)}} \left(\widehat{\boldsymbol{f}}_{t,s+h|\kappa_{\max}} \right). \tag{4.11}$$

We observe that there is a single latent factor process that describes the observations at all frequencies, in contrast, for instance, to hierarchical Hidden Markov Models (HMM) (Hihi and Bengio, 1995) where the latent variables evolve a-priori at different time-scales or like with some of the ESN models developed in this paper.

It is possible to write the above model as a general time-inhomogeneous state-space system (4.5)-(4.6) by suitable parameterizing the time-dependencies in the emission matrices. We provide more details on implementing our mixed frequency DFM in Appendix 7.3. The standard Kalman filtering recursions utilized therein for parameter estimations have a cubic complexity in the dimension d or n of the Markovian factor process $\boldsymbol{f}$ or the observation process $\boldsymbol{y}$, respectively, at every time step. The marginal log-likelihood[5] is optimized based on stochastic gradient methods with adaptive step sizes (Kingma and Ba, 2014) and is generally not a concave function of the parameter values.

Example 4.4 (Quarterly-Monthly-Daily DFM Model) We consider $n_{(\mathtt{d})}$ daily time series over 24 days per calendar month that are averaged over 6 days and denoted as $\boldsymbol{y}^{(\mathtt{d})}$. Furthermore, we consider $n_{(\mathtt{m})}$ monthly $\boldsymbol{y}^{(\mathtt{m})}$ as well as $n_{(\mathtt{q})}$ quarterly time series $\boldsymbol{y}^{(\mathtt{q})}$. Notice that $\kappa_{\max} = 72/6 = 12$. The latent factor process of dimension k is updated every 6 days in line with the 6-month average of the daily data, and assumed to have the AR(1) dynamics

$$\boldsymbol{f}_{t,s+1|12} = A^{(1)} \boldsymbol{f}_{t,s|12} + R\boldsymbol{u}^{(\mathtt{d})}_{t,s+1|12}$$

for any $s, t \in \mathbb{N}$, $A^{(1)} \in \mathbb{M}_{k,k}$ and $R \in \mathbb{M}_k$. The averaged daily data is described by

$$\boldsymbol{y}^{(\mathtt{d})}_{t,s|12} = \beta^{(\mathtt{d})} \boldsymbol{f}_{t,s|12} + S^{(\mathtt{d})} \boldsymbol{u}_{t,s|12}$$

for $\beta^{(\mathtt{d})} \in \mathbb{M}_{n_{(\mathtt{d})},k}$, $S^{(\mathtt{d})} \in \mathbb{M}_{n_{(\mathtt{d})}}$ and IID $n_{(\mathtt{d})}$-dimensional standard Gaussian variables $\boldsymbol{u}^{(\mathtt{d})}_{t,s|12}$. The monthly data in the stock aggregation scheme are modeled as

$$\boldsymbol{y}^{(\mathtt{m})}_{t,s|12} = \beta^{(\mathtt{m})} \boldsymbol{f}_{t,s|12} + S^{(\mathtt{m})} \boldsymbol{u}^{(\mathtt{m})}_{t,s|12}$$

for $s \bmod 4 = 0$ with $\beta^{(\mathtt{m})} \in \mathbb{M}_{n_{(\mathtt{m})},k}$, $S^{(\mathtt{m})} \in \mathbb{M}_{n_{(\mathtt{m})}}$ and IID $n_{(\mathtt{m})}$-dimensional standard Gaussian variables $\boldsymbol{u}^{(\mathtt{m})}_{t,s|12}$, whilst an Almon aggregation scheme yields the model

$$\boldsymbol{y}^{(\mathtt{m})}_{t,s|12} = \beta^{(\mathtt{m})} \sum_{\ell=0}^{4} \overline{\boldsymbol{\varphi}}(\boldsymbol{\psi}^{(\mathtt{m})}, \ell) \odot \boldsymbol{f}_{t,s-\ell|12} + S^{(\mathtt{m})} \boldsymbol{u}^{(\mathtt{m})}_{t,s|12}$$

for $s \bmod 4 = 0$ with $\beta^{(\mathtt{m})} \in \mathbb{M}_{n_{(\mathtt{m})},k}$, $S^{(\mathtt{m})} \in \mathbb{M}_{n_{(\mathtt{m})}}$, IID $n_{(\mathtt{m})}$-dimensional standard Gaussian variables $\boldsymbol{u}^{(\mathtt{m})}_{t,s|12}$ and $\overline{\boldsymbol{\varphi}}(\boldsymbol{\psi}^{(\mathtt{m})}, \ell) = \left(\overline{\varphi}({\psi^{(\mathtt{m})}}_1, \ell), \ldots, \overline{\varphi}({\psi^{(\mathtt{m})}}_k, \ell)\right)^\top \in \mathbb{R}^k$. The symbol $\odot$ stands for the Hadamard or componentwise matrix product. The quarterly components are described analogously as

$$\boldsymbol{y}^{(\mathtt{q})}_{t,0|12} = \beta^{(\mathtt{q})} \boldsymbol{f}_{t,0|12} + S^{(\mathtt{q})} \boldsymbol{u}^{(\mathtt{q})}_{t,0|12}$$

for a stock aggregation scheme, while the Almon scheme writes as

$$\boldsymbol{y}^{(\mathtt{q})}_{t,0|12} = \beta^{(\mathtt{q})} \sum_{\ell=0}^{12} \overline{\boldsymbol{\varphi}}(\boldsymbol{\psi}^{(\mathtt{q})}, \ell) \odot \boldsymbol{f}_{t,-\ell|12} + S^{(\mathtt{q})} \boldsymbol{u}^{(\mathtt{q})}_{t,0|12}$$

with $\beta^{(\mathtt{q})} \in \mathbb{M}_{n_{(\mathtt{q})},k}$, $S^{(\mathtt{m})} \in \mathbb{M}_{n_{(\mathtt{q})}}$, IID $n_{(\mathtt{q})}$-dimensional standard Gaussian variables $\boldsymbol{u}^{(\mathtt{m})}_{t,0|12}$ and $\overline{\boldsymbol{\varphi}}(\boldsymbol{\psi}^{(\mathtt{q})}, \ell) = \left(\overline{\varphi}({\psi^{(\mathtt{q})}}_1, \ell), \ldots, \overline{\varphi}({\psi^{(\mathtt{q})}}_k, \ell)\right)^\top \in \mathbb{R}^k$.

[5] We compute gradients of the marginal log-likelihood using a Kalman filter implementation for a time-inhomogeneous linear Gaussian state space model in TensorFlow Probability (Dillon et al., 2017).

5 Empirical Study

In this section, we compare the forecasting performance of our proposed MFESN models to that of MIDAS and DFMs. We use a combination of macroeconomic and financial data sampled at low and high-frequency intervals, respectively. Our empirical exercises encompass several setups, with a small and a medium-sized set of regressors, fitting models with data before and after the 2007-08 crisis, and with different estimation windows.

5.1 Data

Two sets of predictors of different sizes are compiled: 'Small-MD' with 9 predictors and 'Medium-MD' with 33 predictors in monthly and daily frequency. The reference frequency is quarterly: this is the frequency at which the target variable, US GDP growth, is available. Seasonally adjusted quarterly and monthly data is obtained from the Federal Reserve Bank of St. Louis Monthly (FRED-MD) and Quarterly (FRED-QD) Databases for Macroeconomic Research; see McCracken and Ng (2016, 2020) for detail. Daily data is obtained from Refinitiv Datastream, a subscription-based data service. All data is the last revised vintage data. The macroeconomic target and predictors, their transformations, and availability are listed in Table 5.1.

The selection of predictors follows the seminal work by Stock and Watson (1996, 2006) in which the FRED-MD and FRED-QD data are proposed. Variations of their dataset have been used profusely in the literature; e.g. see Boivin and Ng (2005), Marcellino et al. (2006), Hatzius et al. (2010). Indicators from ten macroeconomic and financial categories are considered: (1) output and income, (2) labor market, (3) housing, (4) orders and inventories, (5) price indices, (6) money and credit, (7) interest rates, (8) exchange rates, (9) equity, and (10) derivatives. The latter five categories represent financial market conditions and are sourced at daily frequency. The exception are interest rates, which move relatively slowly and enter as monthly aggregates, available in the FRED-MD data. A subset of predictors is selected for the 'Small-MD' dataset by choosing variables that have been identified as leading indicators in the empirical literature; e.g. Ingenito and Trehan (1996), Clements and Galvão (2008), Andreou et al. (2013), Marsilli (2014), Ferrara et al. (2014), Carriero et al. (2019), Jardet and Meunier (2020). Data availability is an additional criterion, and predictors unavailable before 1990 are not considered.[6]

We follow instructions by McCracken and Ng (2016, 2020) on pre-processing macroeconomic predictors before they are used as input for forecasting. These are mainly differenced for detrending. We further transform financial predictors to capture market disequilibrium and volatility. Disequilibrium indicators, such as interest rate spreads, have been found to be more relevant for macroeconomic prediction than routine changes captured by differencing; e.g. see Borio and Lowe (2002), Gramlich et al. (2010), Qin et al. (2022). In addition to disequilibrium indicators, realized stock market volatility has been found to improve macroeconomic predictions; e.g. Chauvet et al. (2015). In the absence of intraday trading data from the 1990s onward, which prevents us from utilising conventional daily realized volatility indicators, we extract volatility indicators from daily price series by fitting a simple GARCH(1,1) by Bollerslev (1986).[7] In addition to volatility of stock and commodity prices, term structure indicators are used. The term structure is forward-looking, capturing information about future demand and supply, and has been found to be a leading predictor of GDP growth; e.g. Hong and Yogo (2012), Kang and Kwon (2020).

The data spans the period January 1st, 1990 to December 31st, 2019.[8] We are interested in evaluating model performance under two stylized settings. First, a researcher fits all models up until the

[6] This excludes the VIX volatility index, which has been identified as a leading indicator in some studies; e.g. Andreou et al. (2013), Jardet and Meunier (2020).

[7] We include a control `scale = 1` to ensure convergence of the optimization algorithm and only include a constant mean term in the base process for simplicity.

[8] In the Small-MD dataset experiments we make a small variation and instead include data starting from 1st January 1975, but *only* for the initial CV selection of ridge penalties for MFESN models. Our aim is to make sure that at least for the fixed coefficient model – where λ is cross-validated once and only one $\widehat{W}$ is estimated – the ridge estimator is somewhat robust. In practice, when we compare to expanding and rolling window estimators, where λ is re-selected at each window, we find that extending the initial CV data window has little impact on out-of-sample performance.

Great Recession, including data from Q1 1990 to Q4 2007. Second, fitting is done with data largely encompassing the crisis period, again from Q1 1990 but now up to Q4 2011. In both cases, the testing sample ranges from the next GDP growth observation after fitting up to Q4 2019. All exercises exclude the global COVID-19 economic depression, as we consider it as an extreme, unpredictable event which induces significant structural changes of the underlying macroeconomic dynamics.[9] We emphasize that in the 'Medium-MD' setup, we do not include any MIDAS model specification. The reasons are multiple, and are mainly related to the computational issues associated with the MIDAS regression framework. First, the number of MIDAS parameters increases unless a careful model selection step is performed. Note, however, that our DFM specifications include a MIDAS-type aggregation scheme, in a similar fashion to Marcellino and Schumacher (2010): the factor structure effectively mitigates the parameter poliferiation. We also observe that the MIDAS estimation can be hard to perform in practice due to the complexity of nonlinear optimization. In Appendix 7.5.1 we document, using a simple replication experiment, how the Almon-scheme MIDAS loss can have a large number of distinct local minima which can be randomly selected depending on the initial conditions of the numerical optimization algorithm. These issues are present notwithstanding the fact that we follow Kostrov (2021) in implementing closed-form gradients for the MIDAS models and employ multi-start optimization routines.

To avoid having to handle the many edge cases that daily data in its "raw" calendar releases involves, we use an interpolation approach. We set *ex ante* the number of working days in *any* month to be exactly 24: given that in forecasting the most recent information sets are more relevant, when interpolating daily data over months with less than calendar 24 observations, we linearly interpolate the "missing" data starting from a months' beginning (using the previous months' last observation). The choice of 24 as a daily frequency is transparent by noting that this is the closest number to actual commonly observed data releases, whilst also being a multiple of both 4 (approximate number of weeks per month) and 6 (upper bound on the number of working days per week).

5.2 Models

In this section, we present the set of models that we use throughout our empirical exercises. For a general overview, Table 5.2 summarizes all models, including hyperparameters.

As a baseline benchmark, we use the unconditional mean of the sample used for fitting. For GDP growth forecasting, there is evidence that the unconditional mean produces forecasts which are, in MSE terms, often competitive with linear models such as VARs, even at relatively short horizons (Arora et al., 2013). It is therefore an important point of reference for the performance of all other forecasts: in fact, we shall use it as the reference to present relative MSFE and RMFSE numbers in the tables below.

The non-trivial mixed-frequency model benchmark is given by a MIDAS model that we parametrize to include 30 daily lags and 9 monthly lags for all daily and monthly series, respectively. Since we use a dynamic MIDAS specification, 3 lags of the target variable plus intercept are included as linear regressors. We believe that this is a reasonable choice for MIDAS, as it strikes a balance between lag depth and parsimony, given that the strength of the Almon polynomial weighing is the reduction of an arbitrary number of lag coefficients to just a couple of parameters. To make optimization efficient, we also implement MIDAS loss gradients explicitly as detailed in Kostrov (2021). Our initial coefficient values for optimization are all zeros.[10]

Two distinct DFM specifications are used. The first uses the standard linear aggregation scheme, as detailed in Example 4.1 (we term it DFM [A]), while the second is a variation that implements an Almon weighting scheme – much like the MIDAS model – as presented in Example 4.2 (we term it DFM [B]). A key choice for a DFM model is the dimension of the factors. While a number of methods

[9] In the macroeconomic literature this falls under the category of "natural disaster" events, and should not be naively modeled together with previous observations. In this section, we therefore avoid dealing with post-COVID-19 macroeconomic data altogether.

[10] A subtlety in MIDAS model fitting that we have not discussed is the fact that, due to the Almon exponential equations, some weighting schemes might require to compute floating-point numbers that exceed common numerical precision. Therefore, we elect to always begin gradient descent at the origin of the coefficient space.

Table 5.1: Variables, Frequencies and Transformations for Small and Medium

S	M	Start Date	T	Code	Name	Description
Quarterly						
X	X	31/03/1959	5	GDPC1	Y	Real Gross Domestic Product
Monthly						
X	X	30/01/1959	5	INDPRO	XM1	Industrial Production Index
X	X	30/01/1959	5	PAYEMS	XM4	Payroll All Employees: Total nonfarm
X	X	30/01/1959	4	HOUST	XM5	Housing Starts: Total New Privately Owned
X	X	30/01/1959	5	RETAILx	XM7	Retail and Food Services Sales
X	X	31/01/1973	5	TWEXMMTH	XM11	Nominal effective exchange rate US
X	X	30/01/1959	2	FEDFUNDS	XM12	Effective Federal Funds Rate
X	X	30/01/1959	1	BAAFFM	XM14	Moody's Baa Corporate Bond Minus FEDFUNDS
X	X	30/01/1959	1	COMPAPFFx	XM15	3-Month Commercial Paper Minus FEDFUNDS
	X	30/01/1959	2	CUMFNS	XM2	Capacity Utilization: Manufacturing
	X	30/01/1959	2	UNRATE	XM3	Civilian Unemployment Rate
	X	30/01/1959	5	DPCERA3M086SBEA	XM6	Real personal consumption expenditures
	X	30/01/1959	5	AMDMNOx	XM8	New Orders for Durable Goods
	X	31/01/1978	2	UMCSENTx	XM9	Consumer Sentiment Index
	X	30/01/1959	6	WPSFD49207	XM10	PPI: Finished Goods
	X	30/01/1959	1	AAAFFM	XM13	Moody's Aaa Corporate Bond Minus FEDFUNDS
	X	30/01/1959	1	TB3SMFFM	XM16	3-Month Treasury C Minus FEDFUNDS
	X	30/01/1959	1	T10YFFM	XM17	10-Year Treasury C Minus FEDFUNDS
	X	30/01/1959	2	GS1	XM18	1-Year Treasury Rate
	X	30/01/1959	2	GS10	XM19	10-Year Treasury Rate
	X	30/01/1959	1	GS10-TB3MS	XM20	10-Year Treasury Rate - 3-Month Treasury Bill
Daily						
X	X	30/01/1959	8	DJINDUS	XD3	DJ Industrial price index
	X	31/12/1963	8	S&PCOMP	XD1	S&P500 price index
	X	01/05/1982	1	ISPCS00-S&PCOMP†	XD2	S&P500 basis spread
	X	11/09/1989	8	SP5EIND	XD4	S&P Industrial price index
	X	31/12/1969	8	GSCITOT	XD5	Spot commodity price index
	X	10/01/1983	8	CRUDOIL	XD6	Spot price oil
	X	02/01/1979	8	GOLDHAR	XD7	Spot price gold
	X	30/03/1982	8	WHEATSF	XD8	Spot price wheat
	X	01/11/1983	8	COCOAIC,COCINUS‡	XD9	Spot price cocoa
	X	30/03/1983	1	NCLC.03-NCLC.01	XD10	Futures price oil term structure
	X	30/10/1978	1	NGCC.03-NGCC.01	XD11	Futures price gold term structure
	X	02/01/1975	1	CWFC.03-CWFC.01	XD12	Futures price wheat term structure
	X	02/01/1973	1	NCCC.03-NCCC.01	XD13	Futures price cocoa term structure

Notes: S and M stand for small and medium dataset respectively. An 'X' indicates selection into the dataset. 'Start Date' is the date for which the series is first available (before data transformations). Following McCracken and Ng (2016, 2020), the transformation codes in column 'T' indicate with D for difference and log for natural logarithm 1: none, 2: D, 3: DD, 4: Log, 5: Dlog, 6: DDlog, 7: percentage change, 8: GARCH volatility. 'Codes' are the codes in the FRED-QD and FRED-MD datasets for quarterly and monthly data and Datastream mnemonic for the remaining frequencies. Missing values due to public holidays are interpolated by averaging over the previous five observations. ‡Available until 20/09/2021. ‡Average before 29/12/2017, COCINUS mean adjusted thereafter.

Model Name	Description	Parameters
Mean	Unconditional mean of estimation sample.	None
MIDAS	Almon-weighted MIDAS regression, linear (unconstrained) autoregressive component.	Autoregressive lags: 3 Monthly freq. lags: 9 Daily freq. lags: 30
DFM [A]	Stock aggregation, VAR(1) factor process.	Number of factors: 5 or 10 (small or medium dataset)
DFM [B]	Almon aggregation, VAR(1) factor process	Number of factors: 5 or 10 (small or medium dataset
singleESN [A]	S-MFESN model: Sparse-normal A, sparse-uniform C. Isotropic ridge regression fit.	Reservoir dim: 30 Sparsity: 33.3% $\rho = 0.5$, $\gamma = 1$, $\alpha = 0.1$
singleESN [B]	S-MFESN model: Sparse-normal A, sparse-uniform C. Isotropic ridge regression fit.	Reservoir dim: 120 Sparsity: 8.3% $\rho = 0.5$, $\gamma = 1$, $\alpha = 0.1$
multiESN [A]	M-MFESN model: Monthly and daily freq. reservoirs. Sparse-normal As, sparse-uniform Cs. Isotropic ridge regression fit.	Reservoir dims: M=100, D=20 Sparsity: M=10%, D=50% M: $\rho = 0.5$, $\gamma = 1.5$, $\alpha = 0$ D: $\rho = 0.5$, $\gamma = 0.5$, $\alpha = 0.1$
multiESN [B]	M-MFESN model: Monthly and daily freq. reservoirs. Sparse-normal As, sparse-uniform Cs. Isotropic ridge regression fit.	Reservoir dims: M=100, D=20 Sparsity: M=10%, D=50% M: $\rho = 0.08$, $\gamma = 0.25$, $\alpha = 0.3$ D: $\rho = 0.01$, $\gamma = 0.01$, $\alpha = 0.99$

Table 5.2: Table of models used in applied forecasting exercises. MFESN state matrices A are normalized by largest eigenvalue; matrices C are normalized by Euclidean norm. All MFESN models have aligned state designs.

have been developed over the years to systematically derive the number of factors – see, for example, the review of Stock and Watson (2016) – commonly used macroeconomic panels feature a number of challenges, such as weak factors (Onatski, 2012). To sidestep these issues, we directly assume that both DFM models have 5 unobserved factors, and further impose that they follow a VAR(1) process. One extant issue with integrating daily data is its very high release frequency compared to monthly and especially quarterly releases: computationally this can be extremely taxing, which might be one of the reasons why to our knowledge we are the first to provide DFM forecasts that include daily data. Our solution is to reduce aggregate daily data every 6 days by averaging, thus leaving 4 observations per month. This eases the computational burden to estimate coefficients and latent states considerably (12 versus 72 daily observations per quarter).

The first set of ESNs we propose is given by two S-MFESN models: one model uses a reservoir of 30 neurons (for shortness we term it singleESN [A]); the other has a larger reservoir with 120 neurons (termed singleESN [B]). The sparsity factor for both models, and of both A and C matrices, is adjusted to be $10/N$, where N is the reservoir size. Both MFESNs share the same hyperparameters, $\rho = 0.5$, $\gamma = 1$, $\alpha = 0.1$, which are values we have not tuned but are credible given other ESN implementations from the literature. To make a fair comparison with the closest alternative approach – dynamic factor models – we also elect to fit models using 6-day-averaged daily data. Note here that for MFESN models the computational savings of averaging are negligible, and are mostly apparent when tuning the ridge penalty via cross-validation.

Our second set of proposed models consists of two M-MFESN. Both have two reservoirs, one for each data frequency – monthly and daily – with 100 and 20 neurons, respectively; sparsity degrees are

1-Step-ahead GDP Forecasting - Small-MD Dataset

Model	Fixed Parameters				Expanding Window				Rolling Window			
	2007		2011		2007		2011		2007		2011	
	MSFE	MCS	MSFE	MCS	MSFE	MCS	MSFE	MCS	MSFE	MCS	MSFE	MCS
Mean	1.000		1.000	**	1.000	**	1.000	**	1.000	**	1.000	**
MIDAS	**0.533**	**	1.300		0.596	**	1.129	*	0.709	**	1.170	*
DFM [A]	0.799		1.337		0.980	*	1.320		0.919	*	1.226	
DFM [B]	0.885		1.221	**	0.982	*	1.022	**	0.948		1.028	**
singleESN [A]	0.721	**	1.015	**	0.597	**	0.867	**	**0.529**	**	0.863	**
singleESN [B]	0.758		**0.921**	**	0.602	**	**0.844**	**	0.561	**	0.930	**
multiESN [A]	0.802		1.250		0.635	**	0.874	**	0.621	**	**0.859**	**
multiESN [B]	0.590	**	0.969	**	**0.552**	**	0.895	**	0.530	**	0.921	**

Table 5.3: Relative MSFE and Model Confidence Set (MCS) comparison between models in 1-step-ahead forecasting exercises. Unconditional mean MSFE used as reference. MCS columns show inclusion among best models: $*$ indicates inclusion at 90% confidence, $**$ indicates inclusion at 75% confidence.

again adjusted to be $10/N$, where N is the reservoir size. The first M-MFESN has hyperparameters that are hand-selected among reasonable values: we note that the monthly frequency reservoir has no state leak and a larger input scaling, while the daily frequency reservoir features smaller scaling than usual (in hopes to avoid compressing high volatility events with the sigmoid activation) and the same leak rate as in S-MFESN models (we term this multiESN [A]). For the second M-MFESN, we change hyperparameters more radically: our aim is to set up a model that has a very high input memory, and that also features long-term smoothing of states. Note that here input scaling factors are small, spectral radiuses are an order of magnitude smaller than in previous models, while leak rates are large (we term this last model multiESN [B]).

Finally, in all S-MFESN and M-MFESN models we use the aligned state design, which allows us to forgo introducing more tuning parameters in terms of state lags in the observation equations.

5.3 Results

We now present our empirical results. Competing forecasts are compared using the Model Confidence Set (MCS) test derived by Hansen et al. (2011). One should note that due to the intrinsic nature of data availability of macroeconomic time series and panels, our sample sizes are modest. This implies that the small sample sensitivities of the MCS test need to be taken into account when evaluating our comparisons. In fact, recent analyses of the finite sample properties of the MCS methodology have shown that it requires signal-to-noise ratios which are unattainable in most empirical settings, an issue which undermines its applicability (Aparicio and de Prado, 2018). In view of this fact, we also conduct pairwise model comparison tests with the Modified Diebold-Mariano (MDM) test for predictive accuracy (Diebold and Mariano, 1995, Harvey et al., 1997).

As we also provide multiple-steps-ahead forecasts, we test for the best subset of models uniformly across all horizons using the uniform Multi-Horizon MCS (uMCS) test proposed by Quaedvlieg (2021). Since there is relatively little systematic knowledge regarding the power properties of the uMCS test in small samples, our inclusion of this procedure is meant as a statistical counterpoint to simple relative forecasting errors comparisons, which provide limited information about the significance of performance differences. Finally, we do not report uMCS test outcomes for the expanding window setup, because Quaedvlieg (2021) argues that in such context the test is invalid.

5.3.1 Small Dataset

We begin our discussion of the Small-MD forecasting results by reviewing Table 5.3. For both sample setups (2007 and 2011) and all three estimation setups (fixed, expanding and rolling windows) we provide relative MSFE metrics, with the unconditional mean being used as benchmark, and MCS tests at 90% and 75% confidence levels. Plots of each of the model forecasts are given in Figures 3 and 4; additional plots for cumulative SFE, RMSFEs and other metrics can be found in Appendix 7.6.

The overall finding is that MFESN models perform very well, and, when we exclude the 2007 fixed parameters setup, they perform the best. It is easy to see from Figure 3 (a) why the 2007 fixed window estimation case is different from other cases: the 2008 Financial Crisis induced a deep fall in quarter-to-quarter GDP growth that was in stark contrast with previous business cycle fluctuations. By keeping model parameters fixed, and using only information from 1990 to 2007 – periods where systematic fluctuations are small – DFM and MFESN models are fit to produce smooth, low-volatility forecasts; MIDAS, on the other hand, yields an exponential smoothing which can be more responsive to changes in monthly and daily series. From Figure 3 (b) and (c) it is possible to see that expanding and rolling window estimation resolves this weakness of state-space models.

Table 5.3 shows that MFESN models always perform better than the unconditional mean in terms of MSFE, something which no other model class achieves across all setups. Furthermore, at least one MFESN model for each subclass (single or multiple reservoir) is always included in the model confidence set at the highest confidence level. We remind again that the MCS test of Hansen et al. (2011) might be distorted due to the modest sample sizes considered, even more so in the 2011 test sample. To complement the MCS, we provide graphical tables for pairwise Modified Diebold-Mariano tests, with 10% level rejections highlighted in Figure 5. The MDM tests broadly agree with results of Table 5.3, although of course they do not account for multiple testing, and therefore can not be interpreted as yielding subsets of the most accurate forecasting models in a statistical sense.

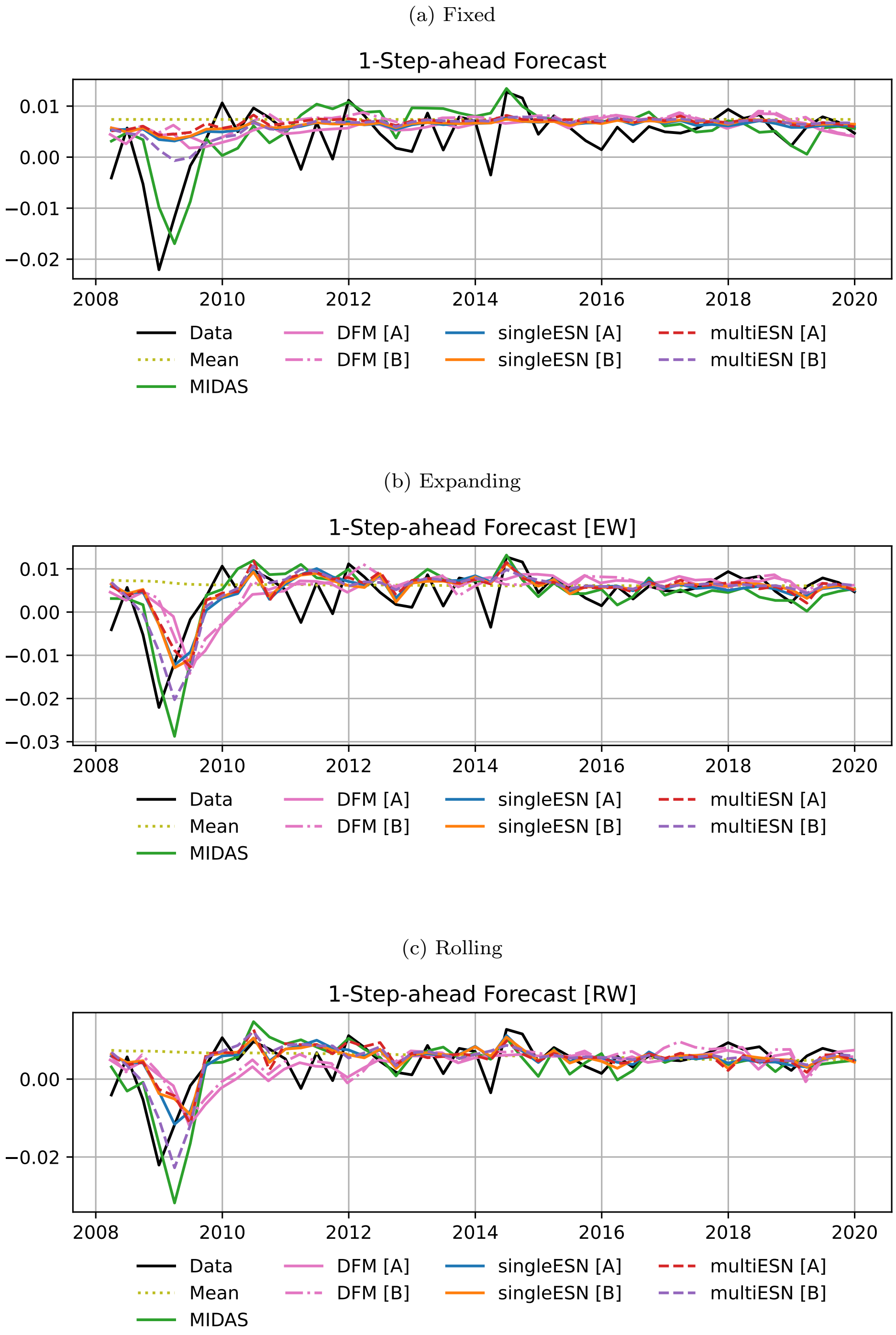
For multiple-steps-ahead forecasts, relative RMSFE and uMCS are reported in Tables 5.4 and 5.5: we constrain our exercise to $h \in \{1, \ldots, 8\}$ steps, since we are interested in GDP growth forecasts within 2 years. Note that for $h = 1$ our results are similar to, but do not reduce to the one-step-ahead results. We note that in order to make correct multistep RMSFE evaluations one must select h different vectors of residuals of the same length: this implies that residuals at the end of the forecasting sample must be trimmed off to compute short-term multistep RMSEs that are comparable to the long-term ones. We actually focus our discussion on Figures 7 and 8, which reproduce graphically the RMSFE numbers of the aforementioned Tables. Generally, we can notice that MIDAS, as well as S-MFESN models provide the worst performing multistep forecasts, with RMSFEs considerably exceeding the unconditional mean baseline after horizon 1. For MIDAS, we have already discussed above how the existence of multiple loss minima can generate numerical instabilities. Model re-fitting at each horizon can amplify this problem, as the loss landscape itsel changes as new observations are added in the fitting sample. We provide more discussions in Appendix 7.5.1. In the case of S-MFESN models, the reason is structural: we have discussed how in our framework multistep MFESN forecasting entails iterating the state map, which can have multiple attraction (stable) points. If the hyperparameters and estimated full model $\widehat{W}$s jointly do not define a contraction, the outcome is that the limit of the multistep forecast needs not be the estimated MFSEN model intercept. However, Figures 7 and 8 show that our M-MFESN models, multiESN [A] and multiESN [B], both perform on par or better than DFM models even after horizon $h = 4$. For example, in the 2007 expanding and rolling window experiments, multiESN [B] is able to outperform both DFMs and even an unconditional mean forecast by meaningful margins for forecasts up to a year into the future.

5.3.2 Medium Dataset

We now present the results for the Medium-MD dataset, which includes more than 30 regressors and many high-frequency daily series. The same metrics used in the previous section to evaluate the relative performance of difference methods are used for this dataset.

The main difference in our empirical exercises is that now we a priori exclude MIDAS from the set of forecasting methods. The main reason to avoid MIDAS is that even in medium-sized setups the

Figure 3: 1-Step-ahead GDP Forecasting – 2007 Sample – Small-MD Dataset

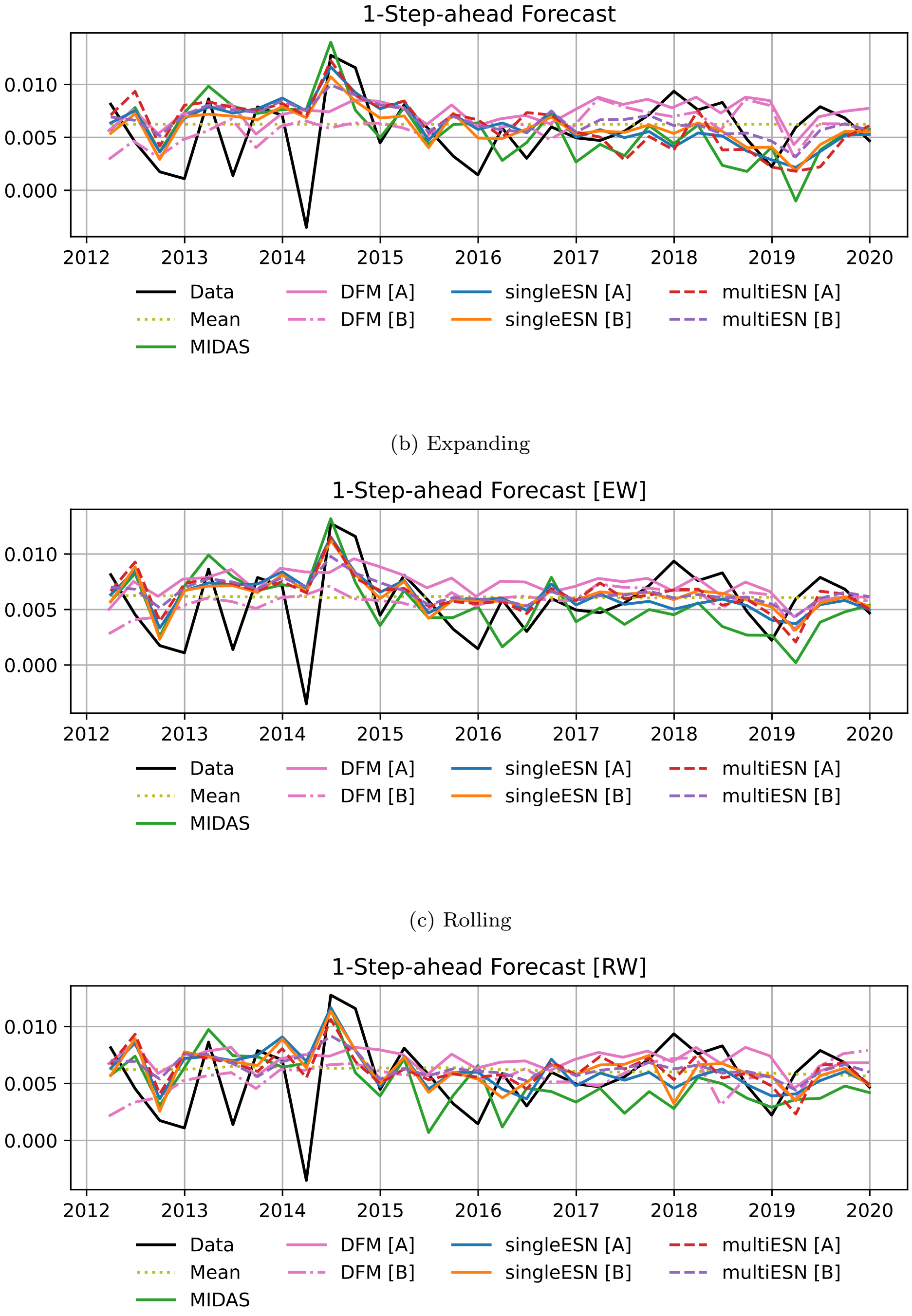


(a) Fixed

(b) Expanding

(c) Rolling

Figure 4: 1-Step-ahead GDP Forecasting – 2011 Sample – Small-MD Dataset

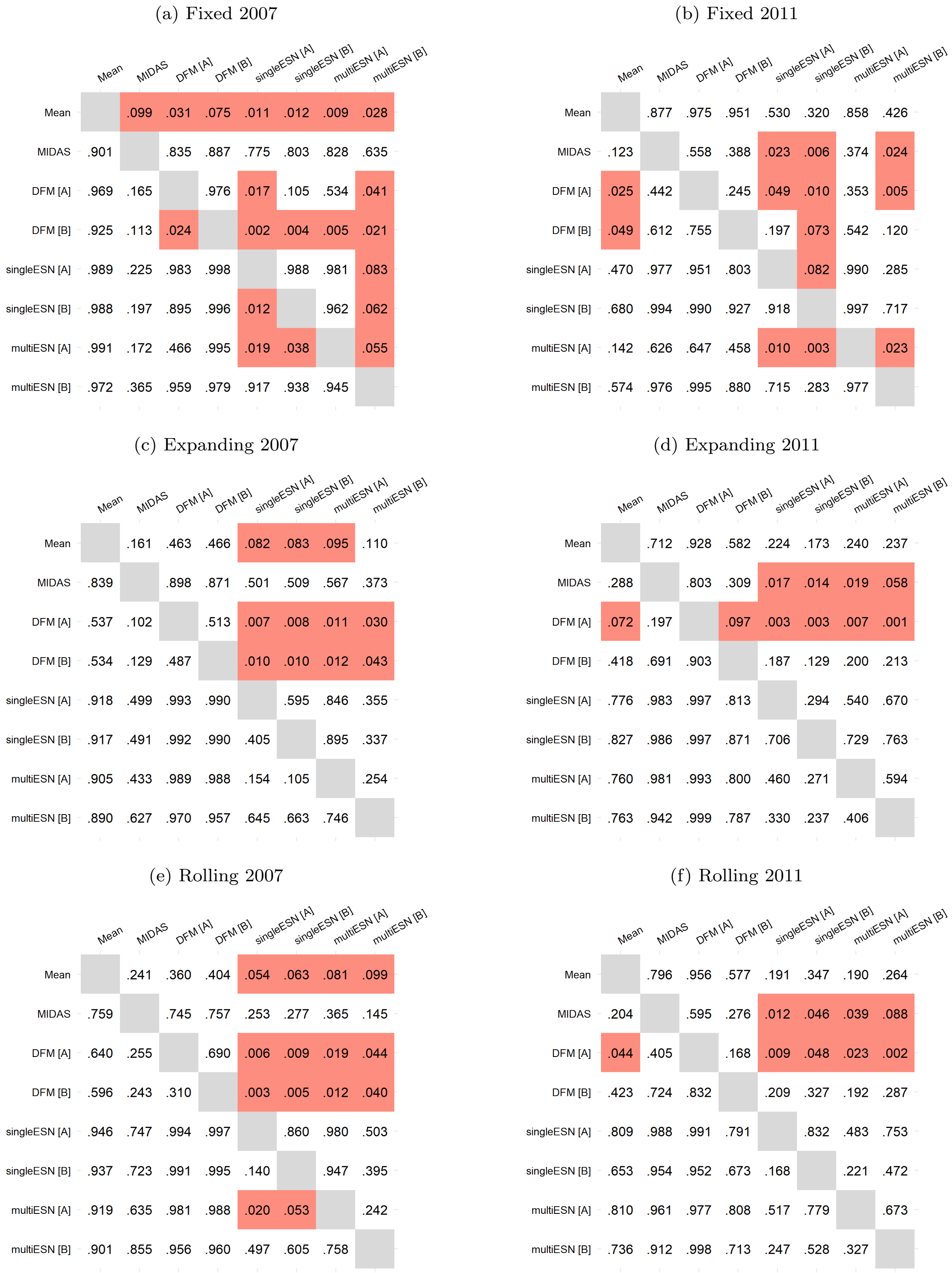


(a) Fixed

(b) Expanding

(c) Rolling

Figure 5: 1-Step-ahead GDP Forecasting – Modified Diebold-Mariano – Small-MD Dataset



(a) Fixed 2007

	Mean	MIDAS	DFM [A]	DFM [B]	singleESN [A]	singleESN [B]	multiESN [A]	multiESN [B]
Mean		.099	.031	.075	.011	.012	.009	.028
MIDAS	.901		.835	.887	.775	.803	.828	.635
DFM [A]	.969	.165		.976	.017	.105	.534	.041
DFM [B]	.925	.113	.024		.002	.004	.005	.021
singleESN [A]	.989	.225	.983	.998		.988	.981	.083
singleESN [B]	.988	.197	.895	.996	.012		.962	.062
multiESN [A]	.991	.172	.466	.995	.019	.038		.055
multiESN [B]	.972	.365	.959	.979	.917	.938	.945	

(b) Fixed 2011

	Mean	MIDAS	DFM [A]	DFM [B]	singleESN [A]	singleESN [B]	multiESN [A]	multiESN [B]
Mean		.877	.975	.951	.530	.320	.858	.426
MIDAS	.123		.558	.388	.023	.006	.374	.024
DFM [A]	.025	.442		.245	.049	.010	.353	.005
DFM [B]	.049	.612	.755		.197	.073	.542	.120
singleESN [A]	.470	.977	.951	.803		.082	.990	.285
singleESN [B]	.680	.994	.990	.927	.918		.997	.717
multiESN [A]	.142	.626	.647	.458	.010	.003		.023
multiESN [B]	.574	.976	.995	.880	.715	.283	.977	

(c) Expanding 2007

	Mean	MIDAS	DFM [A]	DFM [B]	singleESN [A]	singleESN [B]	multiESN [A]	multiESN [B]
Mean		.161	.463	.466	.082	.083	.095	.110
MIDAS	.839		.898	.871	.501	.509	.567	.373
DFM [A]	.537	.102		.513	.007	.008	.011	.030
DFM [B]	.534	.129	.487		.010	.010	.012	.043
singleESN [A]	.918	.499	.993	.990		.595	.846	.355
singleESN [B]	.917	.491	.992	.990	.405		.895	.337
multiESN [A]	.905	.433	.989	.988	.154	.105		.254
multiESN [B]	.890	.627	.970	.957	.645	.663	.746	

(d) Expanding 2011

	Mean	MIDAS	DFM [A]	DFM [B]	singleESN [A]	singleESN [B]	multiESN [A]	multiESN [B]
Mean		.712	.928	.582	.224	.173	.240	.237
MIDAS	.288		.803	.309	.017	.014	.019	.058
DFM [A]	.072	.197		.097	.003	.003	.007	.001
DFM [B]	.418	.691	.903		.187	.129	.200	.213
singleESN [A]	.776	.983	.997	.813		.294	.540	.670
singleESN [B]	.827	.986	.997	.871	.706		.729	.763
multiESN [A]	.760	.981	.993	.800	.460	.271		.594
multiESN [B]	.763	.942	.999	.787	.330	.237	.406	

(e) Rolling 2007

	Mean	MIDAS	DFM [A]	DFM [B]	singleESN [A]	singleESN [B]	multiESN [A]	multiESN [B]
Mean		.241	.360	.404	.054	.063	.081	.099
MIDAS	.759		.745	.757	.253	.277	.365	.145
DFM [A]	.640	.255		.690	.006	.009	.019	.044
DFM [B]	.596	.243	.310		.003	.005	.012	.040
singleESN [A]	.946	.747	.994	.997		.860	.980	.503
singleESN [B]	.937	.723	.991	.995	.140		.947	.395
multiESN [A]	.919	.635	.981	.988	.020	.053		.242
multiESN [B]	.901	.855	.956	.960	.497	.605	.758	

(f) Rolling 2011

	Mean	MIDAS	DFM [A]	DFM [B]	singleESN [A]	singleESN [B]	multiESN [A]	multiESN [B]
Mean		.796	.956	.577	.191	.347	.190	.264
MIDAS	.204		.595	.276	.012	.046	.039	.088
DFM [A]	.044	.405		.168	.009	.048	.023	.002
DFM [B]	.423	.724	.832		.209	.327	.192	.287
singleESN [A]	.809	.988	.991	.791		.832	.483	.753
singleESN [B]	.653	.954	.952	.673	.168		.221	.472
multiESN [A]	.810	.961	.977	.808	.517	.779		.673
multiESN [B]	.736	.912	.998	.713	.247	.528	.327	

Figure 6: p-values of the pairwise Modified Diebold-Mariano tests between models of Table 5.3. Tests are one-sided and carried out row-wise: the null hypothesis for row i and column j reads as "the ith-row model forecasts *more accurately* than the jth-column model". Rejections at 10% level are highlighted in red.

Multistep-ahead GDP Forecasting - Small-MD Dataset - 2007 Sample

Setup	Model	Horizon								uMCS
		1	2	3	4	5	6	7	8	
FIX	Mean	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**
FIX	MIDAS	0.823	1.672	2.737	1.816	2.213	2.791	1.888	1.921	
FIX	DFM [A]	0.890	0.969	1.014	1.077	1.341	1.701	2.001	2.180	
FIX	DFM [B]	0.937	1.069	1.202	1.344	1.799	2.310	2.638	2.801	
FIX	singleESN [A]	0.852	0.994	0.995	0.995	0.993	0.991	0.991	0.991	
FIX	singleESN [B]	0.871	0.986	0.989	0.989	**0.985**	**0.981**	**0.981**	**0.981**	
FIX	multiESN [A]	0.884	0.975	0.988	**0.990**	0.986	0.984	0.984	0.984	
FIX	multiESN [B]	**0.786**	**0.962**	**0.987**	0.993	0.992	0.991	0.991	0.992	**
EW	Mean	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	-
EW	MIDAS	0.814	1.283	1.518	1.596	1.697	1.391	1.951	1.800	-
EW	DFM [A]	0.985	1.109	1.123	1.114	1.217	1.226	1.241	1.539	-
EW	DFM [B]	0.989	1.082	1.149	1.199	1.315	1.412	1.373	1.425	-
EW	singleESN [A]	0.771	1.260	1.485	1.564	2.070	2.728	2.550	2.834	-
EW	singleESN [B]	0.772	1.031	1.135	1.319	1.831	2.279	2.449	2.556	-
EW	multiESN [A]	0.792	0.897	0.941	0.976	1.015	1.240	1.377	1.227	-
EW	multiESN [B]	**0.740**	**0.853**	**0.894**	**0.911**	**0.873**	**0.993**	**1.020**	**1.020**	-
RW	Mean	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**
RW	MIDAS	0.933	1.438	1.642	1.993	1.794	1.661	1.816	1.973	
RW	DFM [A]	0.931	1.017	1.033	1.020	1.024	1.003	**0.918**	**1.062**	*
RW	DFM [B]	0.942	0.973	0.970	1.045	1.059	1.203	1.225	1.263	
RW	singleESN [A]	**0.714**	1.320	1.693	1.972	2.733	3.669	3.391	3.719	*
RW	singleESN [B]	0.737	1.100	1.248	1.667	2.327	2.765	2.842	2.792	*
RW	multiESN [A]	0.773	0.972	1.053	1.111	1.187	1.293	1.505	1.131	
RW	multiESN [B]	0.716	**0.895**	**0.916**	**0.926**	**0.890**	**1.041**	1.102	1.105	**

Table 5.4: Relative RMSFE and Uniform Multi-Horizon Model Confidence Set (uMCS) comparison between models in multiple-steps-ahead forecasting exercises. Unconditional mean RMSFE used as reference. FIX: Fixed parameters, EW: Expanding window, and RW: Rolling window. uMCS columns show inclusion among best models: $*$ indicates inclusion at 90% confidence, $**$ indicates inclusion at 75% confidence.

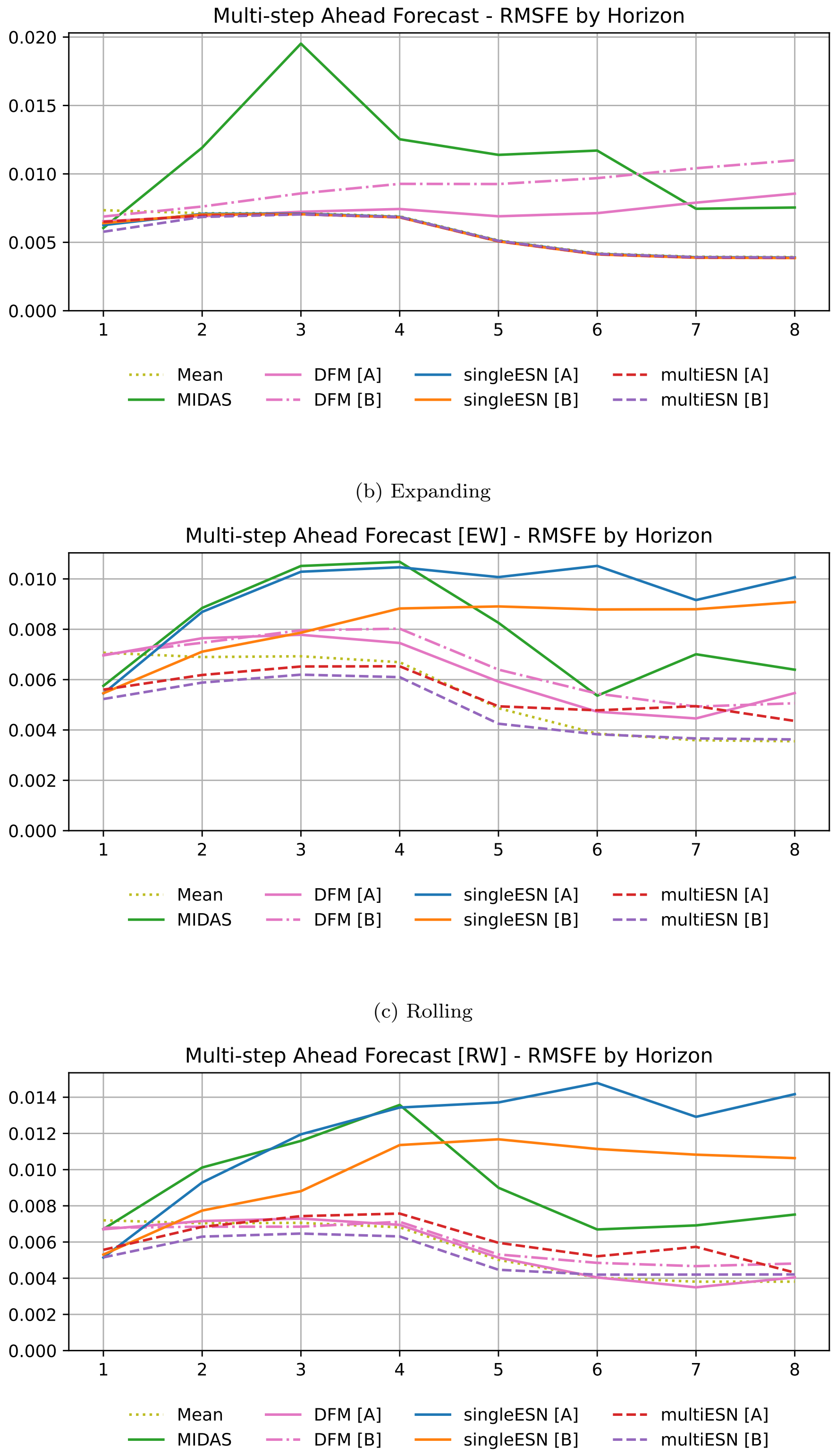
Multistep-ahead GDP Forecasting - Small-MD Dataset - 2011 Sample

Setup	Model	Horizon								uMCS
		1	2	3	4	5	6	7	8	
FIX	Mean	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**
FIX	MIDAS	1.090	1.721	1.793	2.203	2.363	1.997	2.846	2.328	*
FIX	DFM [A]	1.112	1.051	0.999	1.079	1.084	1.025	1.020	1.061	**
FIX	DFM [B]	1.058	**0.945**	**0.916**	1.003	1.012	0.970	1.038	1.033	**
FIX	singleESN [A]	0.978	1.705	2.561	2.704	3.314	3.151	2.999	3.316	**
FIX	singleESN [B]	**0.930**	1.095	1.885	2.356	2.650	2.704	2.880	2.844	**
FIX	multiESN [A]	1.059	1.148	1.262	1.312	1.339	1.409	1.424	1.162	*
FIX	multiESN [B]	0.981	1.007	0.985	**0.994**	**1.008**	**0.999**	**0.999**	**0.998**	**
EW	Mean	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	-
EW	MIDAS	1.005	1.382	1.339	1.354	1.609	1.444	1.803	1.263	-
EW	DFM [A]	1.144	1.132	1.057	1.093	1.076	1.067	1.038	1.016	-
EW	DFM [B]	0.985	**0.940**	**0.918**	0.995	1.010	**0.980**	1.050	**0.971**	-
EW	singleESN [A]	0.935	1.645	2.184	1.929	2.388	1.959	1.810	2.266	-
EW	singleESN [B]	**0.911**	1.092	1.101	1.529	2.195	1.843	1.847	2.060	-
EW	multiESN [A]	0.922	0.965	1.089	0.978	**0.977**	1.043	1.278	0.995	-
EW	multiESN [B]	0.944	0.992	0.978	**0.977**	0.991	0.985	**0.990**	0.996	-
RW	Mean	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**
RW	MIDAS	1.051	1.303	1.310	1.674	1.762	1.467	1.643	1.463	**
RW	DFM [A]	1.061	1.033	1.012	1.088	1.077	1.015	1.040	1.069	**
RW	DFM [B]	0.947	**0.893**	**0.901**	1.009	1.040	**0.966**	1.030	**0.949**	**
RW	singleESN [A]	0.919	1.788	2.359	2.483	2.981	2.401	2.234	2.690	*
RW	singleESN [B]	0.944	1.132	1.214	1.762	2.608	2.552	2.517	2.541	**
RW	multiESN [A]	**0.896**	1.047	1.222	1.124	1.122	1.410	1.666	1.316	**
RW	multiESN [B]	0.940	1.003	0.969	**0.989**	**0.979**	0.972	**0.967**	0.961	**

Table 5.5: Relative RMSFE and Uniform Multi-Horizon Model Confidence Set (uMCS) comparison between models in multiple-steps-ahead forecasting exercises. Unconditional mean RMSFE used as reference. FIX: Fixed parameters, EW: Expanding window, and RW: Rolling window. uMCS columns show inclusion among best models: $*$ indicates inclusion at 90% confidence, $**$ indicates inclusion at 75% confidence.

Figure 7: Multistep-ahead GDP Forecasting, RMSFE – 2007 Sample – Small-MD Dataset

(a) Fixed

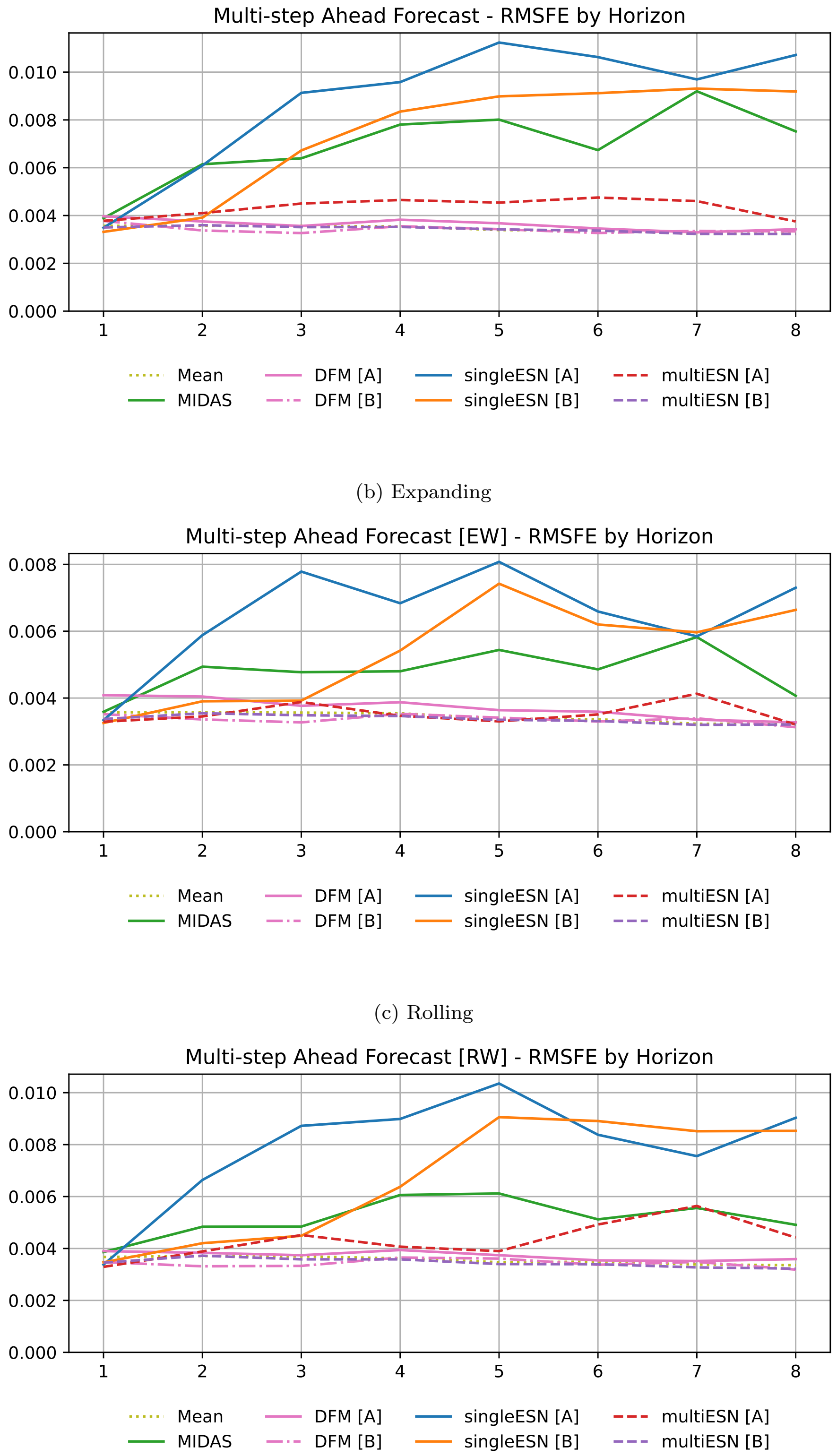


(b) Expanding

(c) Rolling

Figure 8: Multistep-ahead GDP Forecasting, RMSFE – 2011 Sample – Small-MD Dataset

(a) Fixed



(b) Expanding

(c) Rolling

1-Step-ahead GDP Forecasting - Medium-MD Dataset

Model	Fixed Parameters				Expanding Window				Rolling Window			
	2007		2011		2007		2011		2007		2011	
	MSFE	MCS	MSFE	MCS	MSFE	MCS	MSFE	MCS	MSFE	MCS	MSFE	MCS
Mean	1.000		1.000	*	1.000	**	1.000	**	1.000	**	1.000	**
DFM [A]	0.841		1.325	*	0.682	**	1.272	**	0.747	*	1.517	**
DFM [B]	1.118		1.408	*	0.821	*	1.117	**	0.926		1.186	**
singleESN [A]	0.967		1.717	*	0.775	*	1.072	**	0.791		1.493	*
singleESN [B]	0.826		1.278	*	0.655	**	1.028	**	0.561	**	0.944	**
multiESN [A]	0.901		1.080	*	0.618	**	0.913	**	0.556	**	0.884	**
multiESN [B]	**0.682**	**	**0.748**	**	**0.587**	**	**0.774**	**	**0.547**	**	**0.728**	**

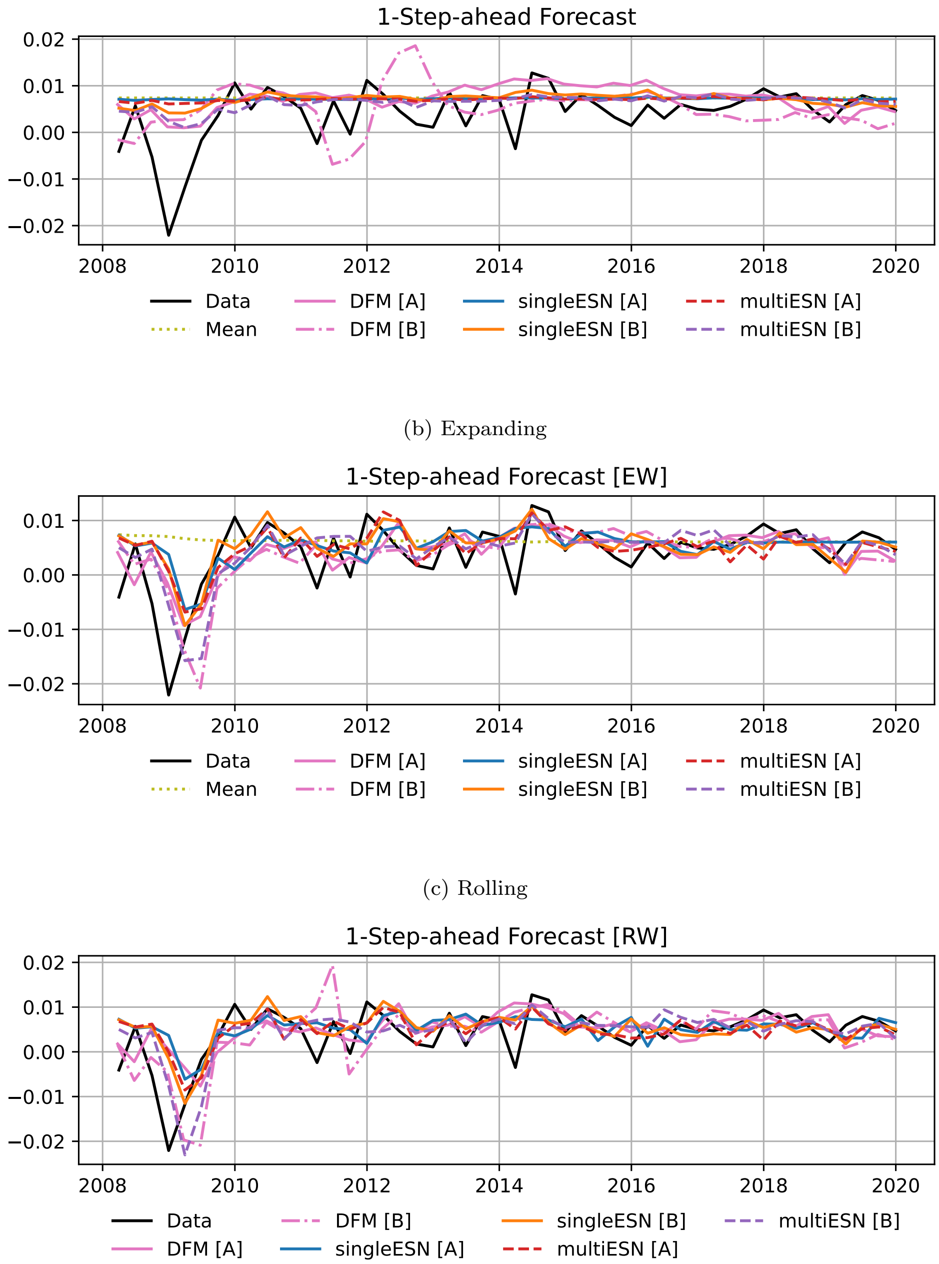
Table 5.6: Relative MSFE and Model Confidence Set (MCS) comparison between models in 1-step-ahead forecasting exercises. Unconditional mean MSFE used as reference. MCS columns show inclusion among best models: $*$ indicates inclusion at 90% confidence, $**$ indicates inclusion at 75% confidence.

computational burden due to numerical optimization makes it impractical to implement, especially in expanding or rolling window estimation setups. In Appendix 7.5.1 we make a more precise evaluation of the issues associated with the nonlinear optimization problem that is inherent in the MIDAS modeling framework. Put simply, even though MIDAS efficiently reduces the number of coefficients involved in multivariate regressions with lags, the loss function becomes highly non-convex. This means that even optimization routines initialized with many starting points might converge to different local minima over different runs. Moreover, in practical applications the optimization domain of MIDAS is often still moderately high-dimensional, because the number of Almon lag coefficients grows linearly with the number of coefficients. To give an example, to produce a MIDAS forecast in the Medium-MD setup we would need to optimize a non-convex loss involving at least 100 parameters.

Table 5.6 showcases the relative performance of DFM and MFESN models in the Medium-MD forecast setup. We find that the MFESN model multiESN [B] performs best in all setups, particularly under fixed parameters, where MCS testing reveals that it is the only model included at a 75% confidence level. Of course, for the MCS results we must again take into account the relatively small sample size, which could distort the selection of best model subsets. Modified Diebold-Mariano tests of Figure 11 largely agree with the MCS results: in the fixed parameter setup any pairwise comparison of an alternative model against MFESN multiESN [B] is rejected in favor of the latter. A visual inspection of one-step-ahead forecasts in Figures 9 and 10 also shows that DFM models estimated over the Medium-MD datasets produce forecasts with larger variability than MFESN methods, which is likely the key driver of the difference in performance.

The multistep-ahead experiments are run as for the Small-MD dataset, with a maximum horizon of 8 quarters. Tables 5.7 and 5.8 present the relative RMSFE performance of multistep forecasts for all models, and we use Figures 13 and 14 as references for our discussion. What can be seen visually – and is also reproduced in the Tables – is that multi-reservoir MFESN models and DFM model [A] have the better performance up to 4 quarter ahead; overall, taking into account also the longer term, expanding or rolling window estimation of model multiESN [B] yields the best forecasting results in the 2007 sample setup. The post-crisis 2011 sample setup makes comparison harder, as DFM and M-MFESN models largely produce results in line with the unconditional sample mean. This evaluation is confirmed by uMCS tests, consistently with the multistep-results obtained with the Small-MD dataset.

Figure 9: 1-Step-ahead GDP Forecasting – 2007 Sample – Medium-MD Dataset

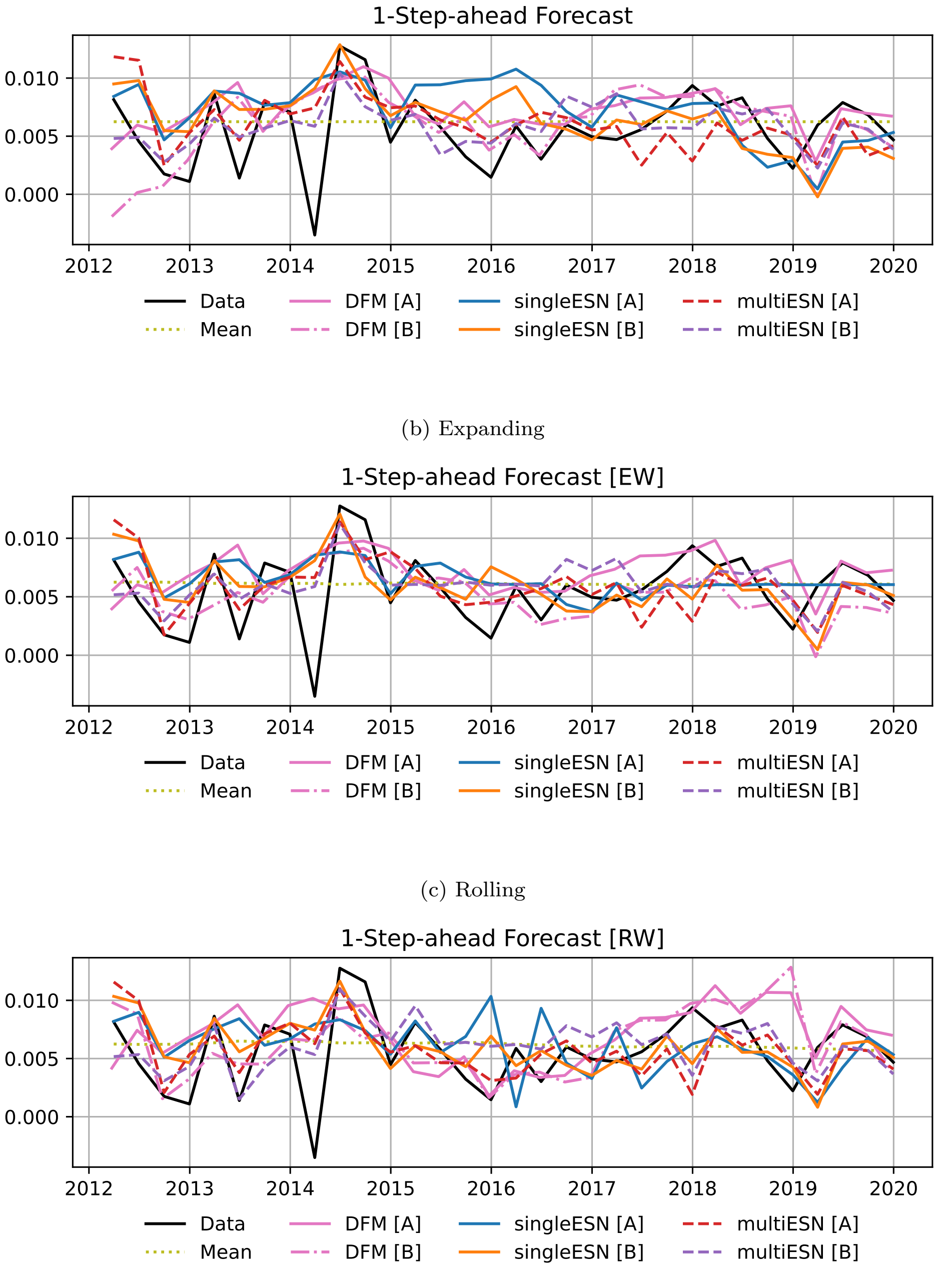


(a) Fixed

(b) Expanding

(c) Rolling

Figure 10: 1-Step-ahead GDP Forecasting – 2011 Sample – Medium-MD Dataset

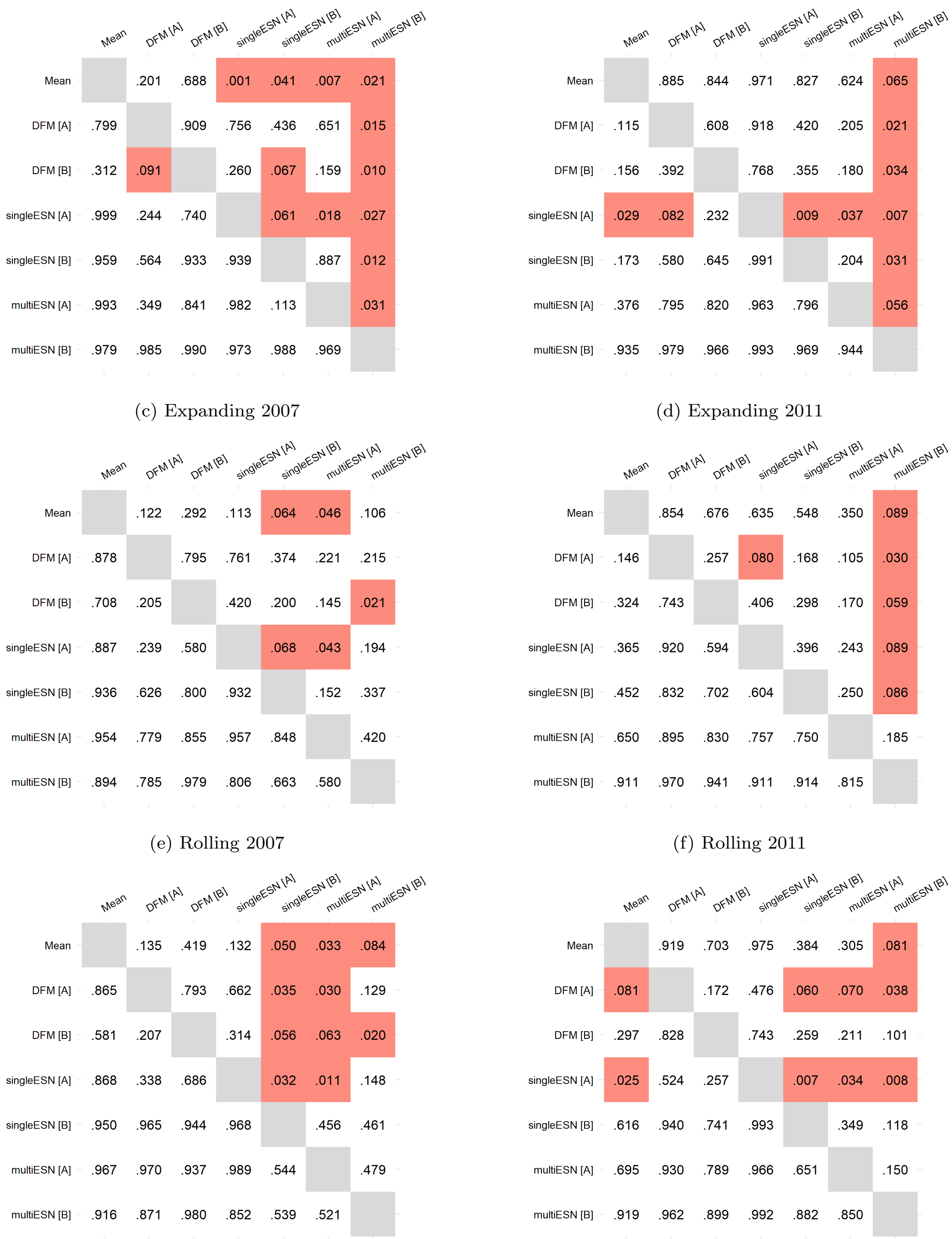


(a) Fixed

(b) Expanding

(c) Rolling

Figure 11: 1-Step-ahead GDP Forecasting – Modified Diebold-Mariano – Medium-MD Dataset



(a) Fixed 2007

	Mean	DFM [A]	DFM [B]	singleESN [A]	singleESN [B]	multiESN [A]	multiESN [B]
Mean		.201	.688	.001	.041	.007	.021
DFM [A]	.799		.909	.756	.436	.651	.015
DFM [B]	.312	.091		.260	.067	.159	.010
singleESN [A]	.999	.244	.740		.061	.018	.027
singleESN [B]	.959	.564	.933	.939		.887	.012
multiESN [A]	.993	.349	.841	.982	.113		.031
multiESN [B]	.979	.985	.990	.973	.988	.969	

(b) Fixed 2011

	Mean	DFM [A]	DFM [B]	singleESN [A]	singleESN [B]	multiESN [A]	multiESN [B]
Mean		.885	.844	.971	.827	.624	.065
DFM [A]	.115		.608	.918	.420	.205	.021
DFM [B]	.156	.392		.768	.355	.180	.034
singleESN [A]	.029	.082	.232		.009	.037	.007
singleESN [B]	.173	.580	.645	.991		.204	.031
multiESN [A]	.376	.795	.820	.963	.796		.056
multiESN [B]	.935	.979	.966	.993	.969	.944	

(c) Expanding 2007

	Mean	DFM [A]	DFM [B]	singleESN [A]	singleESN [B]	multiESN [A]	multiESN [B]
Mean		.122	.292	.113	.064	.046	.106
DFM [A]	.878		.795	.761	.374	.221	.215
DFM [B]	.708	.205		.420	.200	.145	.021
singleESN [A]	.887	.239	.580		.068	.043	.194
singleESN [B]	.936	.626	.800	.932		.152	.337
multiESN [A]	.954	.779	.855	.957	.848		.420
multiESN [B]	.894	.785	.979	.806	.663	.580	

(d) Expanding 2011

	Mean	DFM [A]	DFM [B]	singleESN [A]	singleESN [B]	multiESN [A]	multiESN [B]
Mean		.854	.676	.635	.548	.350	.089
DFM [A]	.146		.257	.080	.168	.105	.030
DFM [B]	.324	.743		.406	.298	.170	.059
singleESN [A]	.365	.920	.594		.396	.243	.089
singleESN [B]	.452	.832	.702	.604		.250	.086
multiESN [A]	.650	.895	.830	.757	.750		.185
multiESN [B]	.911	.970	.941	.911	.914	.815	

(e) Rolling 2007

	Mean	DFM [A]	DFM [B]	singleESN [A]	singleESN [B]	multiESN [A]	multiESN [B]
Mean		.135	.419	.132	.050	.033	.084
DFM [A]	.865		.793	.662	.035	.030	.129
DFM [B]	.581	.207		.314	.056	.063	.020
singleESN [A]	.868	.338	.686		.032	.011	.148
singleESN [B]	.950	.965	.944	.968		.456	.461
multiESN [A]	.967	.970	.937	.989	.544		.479
multiESN [B]	.916	.871	.980	.852	.539	.521	

(f) Rolling 2011

	Mean	DFM [A]	DFM [B]	singleESN [A]	singleESN [B]	multiESN [A]	multiESN [B]
Mean		.919	.703	.975	.384	.305	.081
DFM [A]	.081		.172	.476	.060	.070	.038
DFM [B]	.297	.828		.743	.259	.211	.101
singleESN [A]	.025	.524	.257		.007	.034	.008
singleESN [B]	.616	.940	.741	.993		.349	.118
multiESN [A]	.695	.930	.789	.966	.651		.150
multiESN [B]	.919	.962	.899	.992	.882	.850	

Figure 12: p-values of pairwise Modified Diebold-Mariano tests between models of Table 5.3. Tests are one-sided and carried out row-wise: the null hypothesis for row i and column j reads as “the ith-row model forecasts *more accurately* than the jth-column model”. Rejections at 10% level are highlighted in red.

Multistep-ahead GDP Forecasting - Medium-MD Dataset - 2007 Sample

Setup	Model	Horizon								uMCS
		1	2	3	4	5	6	7	8	
FIX	Mean	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**
FIX	DFM [A]	0.914	**0.947**	**0.955**	0.988	1.015	1.027	1.034	0.995	**
FIX	DFM [B]	1.046	1.204	1.293	1.341	1.649	1.984	2.101	2.070	*
FIX	singleESN [A]	0.985	0.995	0.995	0.995	0.994	0.992	0.992	0.992	
FIX	singleESN [B]	0.912	0.985	0.985	**0.985**	**0.980**	**0.976**	**0.976**	**0.976**	*
FIX	multiESN [A]	0.950	0.993	0.994	0.994	0.992	0.990	0.990	0.990	
FIX	multiESN [B]	**0.826**	0.972	0.988	0.990	0.989	0.986	0.985	0.985	*
EW	Mean	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	-
EW	DFM [A]	0.805	0.916	0.978	1.038	1.077	1.126	1.077	1.073	-
EW	DFM [B]	0.893	1.134	1.418	1.567	2.238	2.964	3.375	3.629	-
EW	singleESN [A]	0.879	1.125	1.305	1.442	1.860	2.166	2.361	2.443	-
EW	singleESN [B]	0.802	1.174	1.439	1.744	2.305	2.869	2.935	3.167	-
EW	multiESN [A]	0.780	0.935	1.012	1.005	1.093	1.337	1.328	1.313	-
EW	multiESN [B]	**0.760**	**0.874**	**0.911**	**0.891**	**0.863**	**0.971**	**1.030**	**1.051**	-
RW	Mean	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**
RW	DFM [A]	0.837	0.913	0.924	0.954	1.012	0.997	**1.018**	**1.005**	
RW	DFM [B]	0.932	1.116	1.232	1.414	1.952	2.704	3.183	3.294	
RW	singleESN [A]	0.873	1.274	1.530	1.652	2.095	2.575	2.786	3.014	
RW	singleESN [B]	0.732	1.190	1.490	1.712	2.218	2.861	2.967	3.094	
RW	multiESN [A]	0.732	0.914	0.960	1.011	1.202	1.618	1.683	1.572	
RW	multiESN [B]	**0.731**	**0.871**	**0.875**	**0.844**	**0.771**	**0.971**	1.014	1.014	**

Table 5.7: Relative RMSFE and Uniform Multi-Horizon Model Confidence Set (uMCS) comparison between models in multiple-steps-ahead forecasting exercises. Unconditional mean RMSFE used as reference. FIX: Fixed parameters, EW: Expanding window, and RW: Rolling window. uMCS columns show inclusion among best models: $\ast$ indicates inclusion at 90% confidence, $\ast\ast$ indicates inclusion at 75% confidence.

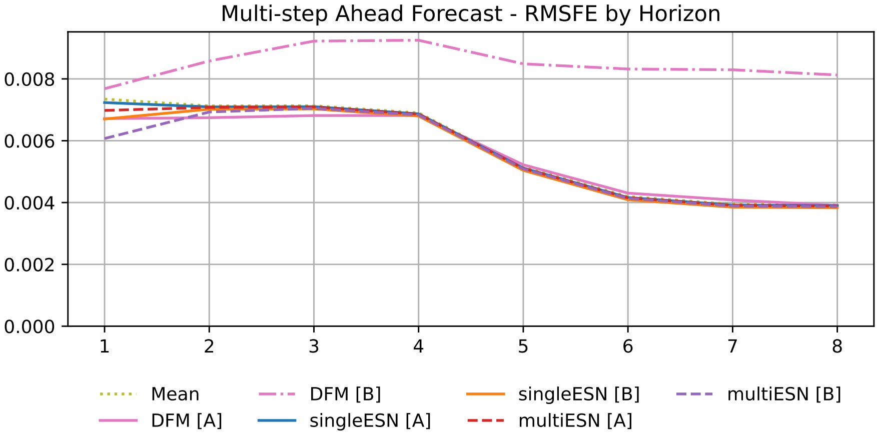
Multistep-ahead GDP Forecasting - Medium-MD Dataset - 2011 Sample

Setup	Model	Horizon								uMCS
		1	2	3	4	5	6	7	8	
FIX	Mean	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**
FIX	DFM [A]	1.126	0.987	0.962	1.054	1.031	**0.988**	**1.001**	**1.002**	**
FIX	DFM [B]	1.149	0.987	**0.885**	1.064	1.142	1.134	1.273	1.296	**
FIX	singleESN [A]	1.283	1.921	2.527	3.038	3.285	3.154	3.193	3.655	**
FIX	singleESN [B]	1.059	1.523	1.918	2.417	2.812	2.683	2.703	2.970	**
FIX	multiESN [A]	1.011	1.061	1.434	1.477	1.748	2.030	2.023	1.994	**
FIX	multiESN [B]	**0.841**	**0.945**	0.997	**0.978**	**1.004**	1.015	1.013	1.014	**
EW	Mean	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	-
EW	DFM [A]	1.092	0.942	**0.944**	1.049	1.026	**0.994**	**0.996**	**0.999**	-
EW	DFM [B]	0.971	1.046	1.031	1.114	1.238	1.116	1.223	1.310	-
EW	singleESN [A]	1.039	1.451	1.980	2.385	2.699	2.353	2.506	2.608	-
EW	singleESN [B]	0.992	1.828	2.465	3.072	3.547	3.357	3.368	3.610	-
EW	multiESN [A]	0.934	1.014	1.391	1.252	1.371	1.369	1.228	1.279	-
EW	multiESN [B]	**0.857**	**0.931**	1.003	**0.973**	**1.002**	1.009	1.025	1.029	-
RW	Mean	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**
RW	DFM [A]	1.113	0.982	**0.927**	1.038	1.030	0.997	1.016	1.028	**
RW	DFM [B]	0.881	0.996	1.021	1.098	1.150	1.114	1.114	1.212	**
RW	singleESN [A]	1.193	2.267	3.265	3.580	4.090	3.790	4.015	4.562	**
RW	singleESN [B]	0.927	1.933	2.612	3.265	3.753	3.567	3.556	3.792	**
RW	multiESN [A]	0.900	1.049	1.500	1.465	1.789	1.707	1.505	1.462	*
RW	multiESN [B]	**0.816**	**0.916**	0.977	**1.009**	**0.982**	**0.988**	**0.974**	**0.981**	**

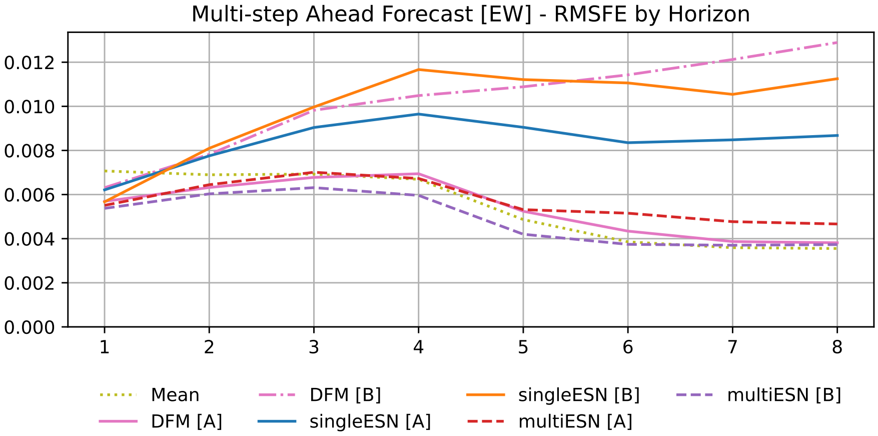
Table 5.8: Relative RMSFE and Uniform Multi-Horizon Model Confidence Set (uMCS) comparison between models in multiple-steps-ahead forecasting exercises. Unconditional mean RMSFE used as reference. FIX: Fixed parameters, EW: Expanding window, and RW: Rolling window. uMCS columns show inclusion among best models: $*$ indicates inclusion at 90% confidence, $**$ indicates inclusion at 75% confidence.

Figure 13: Multistep-ahead GDP Forecasting, RMSFE – 2007 Sample – Medium-MD Dataset

(a) Fixed



(b) Expanding



(c) Rolling

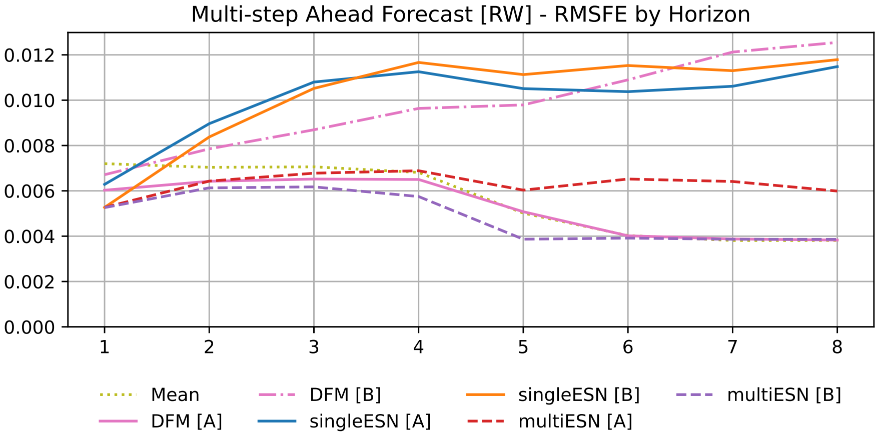
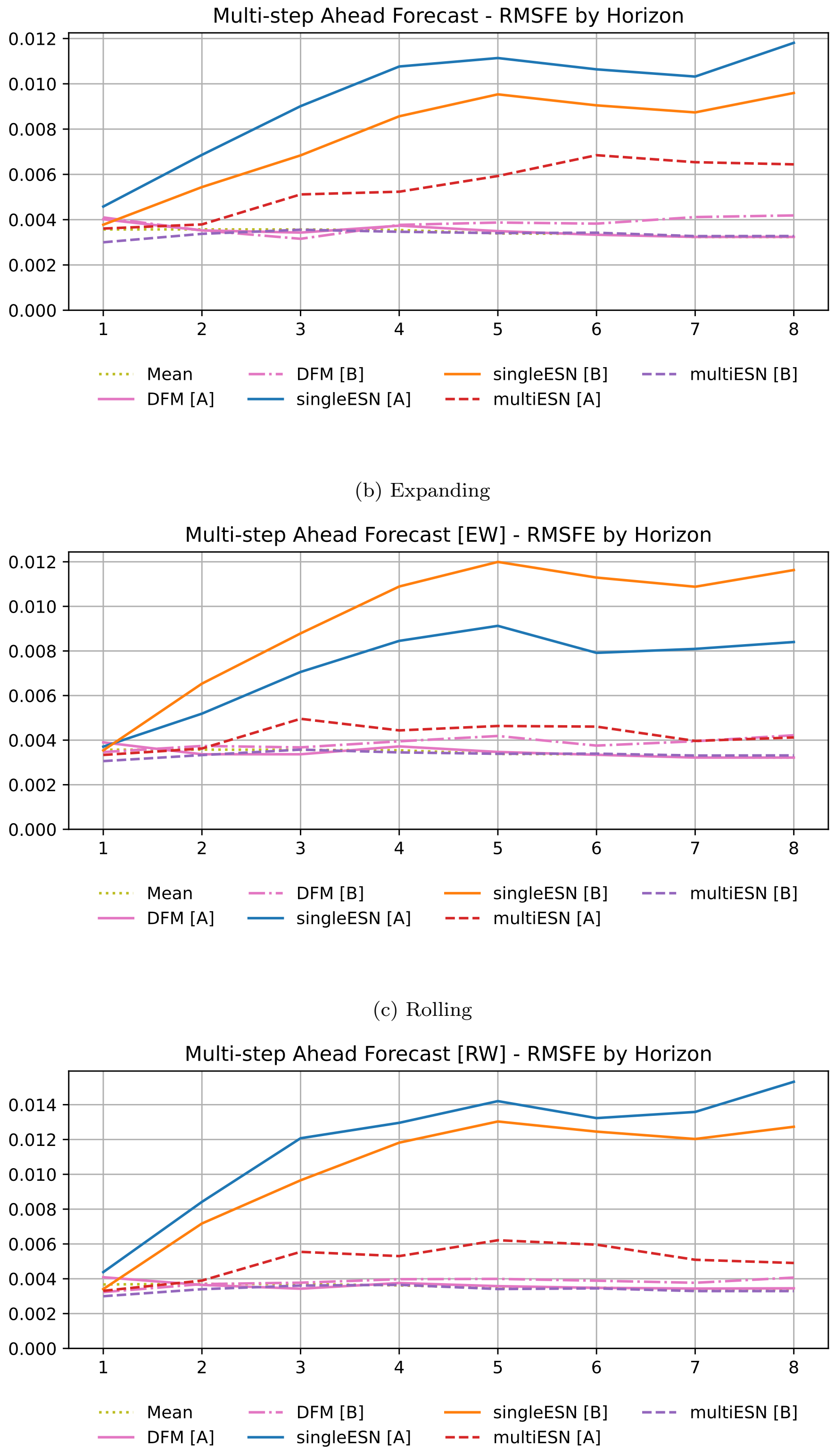


Figure 14: Multistep-ahead GDP Forecasting, RMSFE – 2011 Sample – Medium-MD Dataset

(a) Fixed



(b) Expanding

(c) Rolling

6 Conclusions

Macroeconomic forecasting – especially long-term forecasting of macroeconomic aggregates – is a topic of crucial importance for institutional policymakers, private companies and economic researchers. Given the modern-day availability of "big data" resources, methods capable of integrating heterogeneous data sources are increasingly sought to provide more precise and robust forecasts.

This paper presents a new methodological framework inspired by the Reservoir Computing (RC) literature to deal with data sampled at multiple frequencies and with multiple-steps ahead forecasts. First, we have presented two well-known methods, MIDAS and Dynamic Factor Models, the current benchmarks available in the literature. We have then taken Echo State Networks – a type of RC models – and formally extended them in order to allow modeling of data with multiple release frequencies. Our discussion encompasses model fitting, hyperparameter tuning, and forecast computation. As a result, we provide two classes of models, single- and multiple reservoir multi-frequency ESNs, that can be effectively applied to our empirical setup: forecasting US GDP growth using monthly and daily data series. In our applications, we find that MFESN models are computationally more efficient and easier to implement than DFMs and MIDAS, respectively, and perform better than or as well as benchmarks in terms of mean-square forecasting errors. In a number of setups these improvements are also statistically significant, as shown by our Model Confidence Set and Diebold-Mariano tests. Thus, we argue that our machine learning based methodology can be a useful addition to the toolbox of contemporary macroeconomic forecasters.

Lastly, we wish to highlight the many potential areas of research that we believe would be interesting to explore in the future. We have not discussed the role of the distribution from which we sample the entries of the reservoir matrices. While it is known that these can have significant effects on the forecasting capacity of an ESN model, the literature lacks definitive theoretical results (even for dynamical systems applications) or systematic studies with stochastic inputs and targets. The hyperparameter tuning routine we have developed neither allows separating individual hyperparameters, nor does it tackle the identification problem. Moreover, we assume that the ridge regression penalty strength, λ, is tuned *ex ante*: it would be interesting and desirable to understand if it is possible to jointly tune λ and $\boldsymbol{\varphi}$, or rather if one can fully separate their selection. In our preliminary experiments, we have noticed that the roles of the ridge penalty and the input scaling, for example, cannot be trivially disentangled – thus prompting the ψ-form normalization. Model selection for the dimension of MFESN models is another open question that would be key to exploring and designing more efficient and effective ESN models, especially when dealing with multiple frequencies and reservoirs.

Acknowledgments: The authors acknowledge financial support from UK Research and Innovation (grant number ES/V006347/1). GB thanks the hospitality and the generosity of the University of Warwick without which this paper would not have happened. GB is grateful to the Center for Doctoral Studies in Economics (CDSE) at the University of Mannheim for providing travel funding through the 2020 CDSE Teaching Award. The authors acknowledge support by the state of Baden-Württemberg through bwHPC. The authors acknowledge the use of the UCL Myriad High Performance Computing Facility (Myriad@UCL), and associated support services, in completing this work. MH acknowledges that the majority of his work was done when he was affiliated with UCL and hence thanks the support of UCL in this project. The authors acknowledge the support of the Swiss National Science Foundation (grant number 200021_175801/1). The authors also thank Thanos Moraitis and Alfonso Silva Ruiz for excellent research assistance and help in data collection and curation.

References

S. Almon. The distributed lag between capital appropriations and expenditures. *Econometrica*, 33(1): 178–196, 1965.

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

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov Chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

D. Aparicio and M. L. de Prado. How hard is it to pick the right model? MCS and backtest overfitting. *Algorithmic Finance*, 7(1-2):53–61, jun 2018. doi: 10.3233/af-180231.

T. Arcomano, I. Szunyogh, A. Wikner, J. Pathak, B. R. Hunt, and E. Ott. A hybrid approach to atmospheric modeling that combines machine learning with a physics-based numerical model. *Journal of Advances in Modeling Earth Systems*, 14(3):e2021MS002712, 2022.

M. T. Armesto, K. M. Engemann, and M. T. Owyang. Forecasting with Mixed Frequencies. *Federal Reserve Bank of St. Louis Review*, 92(6):521–536, 2010.

S. Arora, M. A. Little, and P. E. McSharry. Nonlinear and nonparametric modeling approaches for probabilistic forecasting of the US gross national product. *Studies in Nonlinear Dynamics and Econometrics*, 17(4):395–420, Sept. 2013. ISSN 1558-3708. doi: 10.1515/snde-2012-0029.

A. Babii, E. Ghysels, and J. Striaukas. Machine learning time series regressions with an application to nowcasting. *Journal of Business and Economic Statistics*, 40(3):1094–1106, 2022.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

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

J. Bai, E. Ghysels, and J. H. Wright. State Space Models and MIDAS Regressions. *Econometric Reviews*, 32(7):779–813, 2013.

M. Bańbura and M. Modugno. Maximum likelihood estimation of factor models on datasets with arbitrary pattern of missing data. *Journal of Applied Econometrics*, 29(1):133–160, 2014.

M. Bańbura and G. Rünstler. A look into the factor model black box: Publication lags and the role of hard and soft data in forecasting GDP. *International Journal of Forecasting*, 27(2):333–346, apr 2011. doi: 10.1016/j.ijforecast.2010.01.011.

M. Bańbura, D. Giannone, M. Modugno, and L. Reichlin. Now-casting and the real-time data flow. In *Handbook of Economic Forecasting*, pages 195–237. Elsevier, 2013. doi: 10.1016/b978-0-444-53683-9.00004-9.

A. Belloni, V. Chernozhukov, D. Chetverikov, and K. Kato. Some new asymptotic theory for least squares series: Pointwise and uniform results. *Journal of Econometrics*, 186(2):345–366, June 2015. ISSN 0304-4076. doi: 10.1016/j.jeconom.2015.02.014.

C. Bergmeir, R. J. Hyndman, and B. Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120:70–83, Apr. 2018. ISSN 01679473. doi: 10.1016/j.csda.2017.11.003.

F. Blasques, S. J. Koopman, M. Mallee, and Z. Zhang. Weighted maximum likelihood for dynamic factor analysis and forecasting with mixed frequency data. *Journal of Econometrics*, 193(2):405–417, 2016.

J. Boivin and S. Ng. Understanding and comparing factor-based forecasts. Technical report, may 2005.

T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31 (3):307–327, apr 1986. doi: 10.1016/0304-4076(86)90063-1.

C. Borio. Rediscovering the macroeconomic roots of financial stability policy: Journey, challenges, and a way forward. *Annual Review of Financial Economics*, 3(1):87–117, dec 2011. doi: 10.1146/annurev-financial-102710-144819.

C. Borio. The great financial crisis: Setting priorities for new statistics. *Journal of Banking Regulation*, 14(3-4):306–317, jul 2013. doi: 10.1057/jbr.2013.9.

C. Borio and P. W. Lowe. Asset prices, financial and monetary stability: Exploring the nexus. *SSRN Electronic Journal*, 2002. doi: 10.2139/ssrn.846305.

S. Boyd and L. Chua. Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems*, 32(11):1150–1161, 1985.

B. Buell, R. Cherif, C. Chen, Hyeon, J. Tang, and N. Wendt. Impact of COVID-19: Nowcasting and big data to track economic activity in Sub-Saharan Africa. *IMF Working Paper*, 124:1–61, may 2021.

W. Cao, X. Wang, Z. Ming, and J. Gao. A review on neural networks with random weights. *Neurocomputing*, 275:278–287, Jan. 2018. ISSN 09252312. doi: 10.1016/j.neucom.2017.08.040.

A. Carriero, A. B. Galvão, and G. Kapetanios. A comprehensive evaluation of macroeconomic forecasting methods. *International Journal of Forecasting*, 35(4):1226–1239, oct 2019. doi: 10.1016/j.ijforecast.2019.02.007.

M. Chauvet, Z. Senyuz, and E. Yoldas. What does financial volatility tell us about macroeconomic fluctuations? *Journal of Economic Dynamics and Control*, 52:340–360, mar 2015. doi: 10.1016/j.jedc.2015.01.002.

X. Chen. Large Sample Sieve Estimation of Semi-Nonparametric Models. In J. J. Heckman and E. E. Leamer, editors, *Handbook of Econometrics*, volume 6, pages 5549–5632. Elsevier, Jan. 2007. doi: 10.1016/S1573-4412(07)06076-X.

X. Chen and T. M. Christensen. Optimal uniform convergence rates and asymptotic normality for series estimators under weak dependence and weak conditions. *Journal of Econometrics*, 188(2):447–465, Oct. 2015. ISSN 0304-4076. doi: 10.1016/j.jeconom.2015.03.010.

X. Chen and E. Ghysels. News—good or bad—and its impact on volatility predictions over multiple horizons. *Review of Financial Studies*, 24(1):46–81, sep 2010. doi: 10.1093/rfs/hhq071.

X. Chen and H. White. Improved rates and asymptotic normality for nonparametric neural network estimators. *IEEE Transactions on Information Theory*, 45(2):682–691, Mar. 1999. ISSN 1557-9654. doi: 10.1109/18.749011.

M. P. Clements and A. Galvão. Forecasting US output growth using leading indicators: an appraisal using MIDAS models. *Journal of Applied Econometrics*, 7(7):1187–1206, 2009. URL http://onlinelibrary.wiley.com/doi/10.1002/jae.1075/full.

M. P. Clements and A. B. Galvão. Macroeconomic forecasting with mixed-frequency data. *Journal of Business and Economic Statistics*, 26(4):546–554, oct 2008. ISSN 0735-0015. URL http://pubs.amstat.org/doi/abs/10.1198/073500108000000015.

J. P. Crutchfield, W. L. Ditto, and S. Sinha. Introduction to focus issue: intrinsic and designed computation: information processing in dynamical systems-beyond the digital hegemony. *Chaos (Woodbury, N.Y.)*, 20(3):037101, sep 2010. ISSN 1089-7682.

D. Delle Monache and I. Petrella. Efficient matrix approach for classical inference in state space models. *Economics Letters*, 181:22–27, 2019. ISSN 0165-1765. doi: https://doi.org/10.1016/j.econlet.2019.04.012. URL https://www.sciencedirect.com/science/article/pii/S016517651930134X.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal statistical Society*, 39(1):1–38, 1977.

F. X. Diebold and R. S. Mariano. Comparing predictive accuracy. *Journal of Business and Economic Statistics*, 13:253–263, 1995. doi: 10.1198/073500102753410444.

J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

R. Douc, E. Moulines, J. Olsson, and R. Van Handel. Consistency of the maximum likelihood estimator for general hidden Markov models. *The Annals of Statistics*, 39(1):474–513, 2011.

A. Doucet, N. d. Freitas, and N. J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, 2001. ISBN 9780387951461.

K. Doya. Bifurcations in the learning of recurrent neural networks. In *Proceedings of IEEE International Symposium on Circuits and Systems*, volume 6, pages 2777–2780. IEEE, 1992. ISBN 0-7803-0593-0. doi: 10.1109/ISCAS.1992.230622. URL http://ieeexplore.ieee.org/document/230622/.

C. Doz, D. Giannone, and L. Reichlin. A two-step estimator for large approximate dynamic factor models based on kalman filtering. *Journal of Econometrics*, 164(1):188–205, 2011.

I. Farkas, R. Bosak, and P. Gergel. Computational analysis of memory capacity in echo state networks. *Neural Networks*, 83:109–120, 2016. ISSN 18792782. doi: 10.1016/j.neunet.2016.07.012.

L. Ferrara, C. Marsilli, and J.-P. Ortega. Forecasting growth during the Great Recession: is financial volatility the missing ingredient? *Economic Modelling*, 36:44–50, 2014.

M. Forni, M. Hallin, M. Lippi, and L. Reichlin. The generalized dynamic factor model: one-sided estimation and forecasting. *Journal of the American Statistical Association*, 100(471):830–840, 2005.

C. Foroni and M. Marcellino. A comparison of mixed approaches for modelling euro area macroeconomic variables. Technical report, EUI, 2011.

N. Francis, E. Ghysels, and M. T. Owyang. The low-frequency impact of daily monetary policy shocks. Technical report, 2011.

P. Fuleky, editor. *Macroeconomic Forecasting in the Era of Big Data*. Springer International Publishing, 2020. doi: 10.1007/978-3-030-31150-6.

P. Gagliardini, E. Ghysels, and M. Rubin. Indirect inference estimation of mixed frequency stochastic volatility state space models using MIDAS regressions and ARCH models. *Journal of Financial Econometrics*, 15(4):509–560, 2017.

A. B. Galvão. Changes in predictive ability with mixed frequency data. *International Journal of Forecasting*, 29(3):395–410, jul 2013. doi: 10.1016/j.ijforecast.2012.10.006.

A. B. Galvão and M. Marcellino. Endogenous monetary policy regimes and the great moderation. Technical report, EUI, 2010.

J. Geweke. The dynamic factor analysis of economic time series. *Latent variables in socio-economic models*, 1977.

E. Ghysels. Macroeconomics and the reality of mixed frequency data. *Journal of Econometrics*, 193(2):294–314, aug 2016. doi: 10.1016/j.jeconom.2016.04.008.

E. Ghysels and J. H. Wright. Forecasting professional forecasters. *Journal of Business & Economic Statistics*, 27(4):504–516, oct 2009. doi: 10.1198/jbes.2009.06044.

E. Ghysels, P. Santa-Clara, and R. Valkanov. The MIDAS touch : Mixed data sampling regression models. Technical Report 919, mimeo, 2004.

E. Ghysels, A. Sinko, and R. Valkanov. Midas regressions: Further results and new directions. *Econometric reviews*, 26(1):53–90, 2007.

D. Giannone, L. Reichlin, and D. Small. Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics*, 55(4):665–676, may 2008. ISSN 03043932. URL http://dx.doi.org/10.1016/j.jmoneco.2008.05.010.

S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing for nonlinear time series. *Journal of the american statistical association*, 99(465):156–168, 2004.

L. Gonon and J.-P. Ortega. Reservoir computing universality with stochastic inputs. *IEEE transactions on neural networks and learning systems*, 31(1):100–112, 2019.

L. Gonon and J.-P. Ortega. Fading memory echo state networks are universal. *Neural Networks*, 138: 10–13, 2021.

L. Gonon, L. Grigoryeva, and J.-P. Ortega. Risk bounds for reservoir computing. *Journal of Machine Learning Research*, 21(240):1–61, 2020a.

L. Gonon, L. Grigoryeva, and J.-P. Ortega. Memory and forecasting capacities of nonlinear recurrent networks. *Physica D*, 414(132721):1–13., 2020b.

L. Gonon, L. Grigoryeva, and J.-P. Ortega. Approximation error estimates for random neural networks and reservoir systems. *To appear in the Annals of Applied Probability*, 2022.

A. Goudarzi, S. Marzen, P. Banda, G. Feldman, M. R. Lakin, C. Teuscher, and D. Stefanovic. Memory and information processing in recurrent neural networks. Technical report, 2016. URL https://arxiv.org/pdf/1604.06929.pdf.

D. Gramlich, G. L. Miller, M. V. Oet, and S. J. Ong. Early warning systems for systemic banking risk: Critical review and modeling implications. *Banks and Bank Systems*, 5(2):199–211, 2010.

L. Grigoryeva and J.-P. Ortega. Universal discrete-time reservoir computers with stochastic inputs and linear readouts using non-homogeneous state-affine systems. *Journal of Machine Learning Research*, 19(24):1–40, 2018a. URL http://arxiv.org/abs/1712.00754.

L. Grigoryeva and J.-P. Ortega. Echo state networks are universal. *Neural Networks*, 108:495–508, 2018b.

L. Grigoryeva and J.-P. Ortega. Differentiable reservoir computing. *Journal of Machine Learning Research*, 20(179):1–62, 2019.

L. Grigoryeva and J.-P. Ortega. Dimension reduction in recurrent networks by canonicalization. *Journal of Geometric Mechanics*, 13(4):647–677, 2021. doi: 10.3934/jgm.2021028.

L. Grigoryeva, J. Henriques, L. Larger, and J.-P. Ortega. Optimal nonlinear information processing capacity in delay-based reservoir computers. *Scientific Reports*, 5(12858):1–11, 2015. doi: 10.1038/srep12858.

L. Grigoryeva, J. Henriques, L. Larger, and J.-P. Ortega. Nonlinear memory capacity of parallel time-delay reservoir computers in the processing of multidimensional signals. *Neural Computation*, 28: 1411–1451, 2016.

M. Hallin and R. Liška. Determining the number of factors in the general dynamic factor model. *Journal of the American Statistical Association*, 102(478):603–617, 2007.

P. R. Hansen, Z. Huang, and H. H. Shek. Realized GARCH: a joint model for returns and realized measures of volatility. *Journal of Applied Econometrics*, 27(6):877–906, 2011.

A. G. Hart, J. L. Hook, and J. H. P. Dawes. Echo State Networks trained by Tikhonov least squares are $L^2(\mu)$ approximators of ergodic dynamical systems. *Physica D: Nonlinear Phenomena*, 421:132882, 2021.

D. Harvey, S. Leybourne, and P. Newbold. Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, 13(2):281–291, jun 1997. doi: 10.1016/s0169-2070(96)00719-4.

T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, second edition, 2009.

T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2):949–986, Apr. 2022. ISSN 0090-5364, 2168-8966. doi: 10.1214/21-AOS2133.

J. Hatzius, P. Hooper, F. Mishkin, K. Schoenholtz, and M. Watson. Financial conditions indexes: A fresh look after the financial crisis. Technical report, jul 2010.

S. Hihi and Y. Bengio. Hierarchical recurrent neural networks for long-term dependencies. *Advances in neural information processing systems*, 8, 1995.

I. Hindrayanto, S. J. Koopman, and J. de Winter. Forecasting and nowcasting economic growth in the euro area using factor models. *International Journal of Forecasting*, 32(4):1284–1305, oct 2016. doi: 10.1016/j.ijforecast.2016.05.003.

H. Hong and M. Yogo. What does futures market interest tell us about the macroeconomy and asset prices? *Journal of Financial Economics*, 105(3):473–490, sep 2012. doi: 10.1016/j.jfineco.2012.04.005.

J. L. Horowitz. *Semiparametric and Nonparametric Methods in Econometrics.* Springer, New York, NY, USA, 2009. ISBN 978-0-387-92870-8. URL https://link.springer.com/book/10.1007/978-0-387-92870-8.

F. Huber, G. Koop, L. Onorante, M. Pfarrhofer, and J. Schreiner. Nowcasting in a pandemic using non-parametric mixed frequency VARs. *ECB Working Paper Series*, 2510:1–40, jan 2021.

R. Ingenito and B. Trehan. Using monthly data to predict quarterly output. *Econometric Reviews*, pages 3–11, 1996.

A. Inoue, L. Jin, and B. Rossi. Rolling window selection for out-of-sample forecasting with time-varying parameters. *Journal of Econometrics*, 196(1):55–67, Jan. 2017. ISSN 03044076. doi: 10.1016/j.jeconom.2016.03.006.

H. Ishwaran and J. Rao. Geometry and properties of generalized ridge regression in high dimensions. In S. Ahmed, editor, *Contemporary Mathematics*, volume 622, pages 81–93. American Mathematical Society, Providence, Rhode Island, 2014. doi: 10.1090/conm/622/12438.

H. Jaeger and H. Haas. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science*, 304(5667):78–80, 2004. doi: 10.1126/science.1091277.

C. Jardet and B. Meunier. Nowcasting world GDP growth with high-frequency data. *SSRN Electronic Journal*, 2020. doi: 10.2139/ssrn.3749139.

Ò. Jordà. Estimation and Inference of Impulse Responses by Local Projections. *American Economic Review*, 95(1):161–182, Feb. 2005. ISSN 0002-8282. doi: 10.1257/0002828053828518.

B. Jungbacker and S. J. Koopman. Likelihood-based dynamic factor analysis for measurement and forecasting, 2015.

J. Kang and K. Y. Kwon. Can commodity futures risk factors predict economic growth? *Journal of Futures Markets*, 40(12):1825–1860, sep 2020. doi: 10.1002/fut.22155.

N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, N. Chopin, et al. On particle methods for parameter estimation in state-space models. *Statistical science*, 30(3):328–351, 2015.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A. Kostrov. *Essays on the use of MIDAS regressions in banking and finance*. PhD thesis, Universität St. Gallen, 2021.

R. Legenstein and W. Maass. What makes a dynamical system computationally powerful? In S. Haykin, editor, *New directions in statistical signal processing: from systems to brain*. MIT Press, Cambridge, MA, 2007.

F. LeGland and L. Mevel. Recursive estimation in hidden Markov models. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 4, pages 3468–3473. IEEE, 1997.

M. Leippold and H. Yang. Particle filtering, learning, and smoothing for mixed-frequency state-space models. *Econometrics and Statistics*, 12:25–41, 2019.

M. Lukoševičius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation*, 14:2531–2560, 2002.

O.-A. Maillard and R. Munos. Linear Regression With Random Projections. *Journal of Machine Learning Research*, 13(89):2735–2772, 2012. ISSN 1533-7928.

M. Marcellino and C. Schumacher. Factor MIDAS for nowcasting and forecasting with ragged-edge data: A model comparison for German GDP. *Oxford Bulletin of Economics and Statistics*, 72(4):518–550, 2010. ISSN 03059049. URL http://doi.wiley.com/10.1111/j.1468-0084.2010.00591.x.

M. Marcellino, J. H. Stock, and M. W. Watson. A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series. *Journal of Econometrics*, 135(1-2):499–526, nov 2006. ISSN 03044076. URL http://dx.doi.org/10.1016/j.jeconom.2005.07.020.

R. S. Mariano and Y. Murasawa. A new coincident index of business cycles based on monthly and quarterly series. *Journal of applied Econometrics*, 18(4):427–443, 2003.

C. Marsilli. *Mixed-Frequency Modeling and Economic Forecasting*. PhD thesis, Université de Franche-Comté, 2014.

M. W. McCracken and S. Ng. FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4):574–589, sep 2016. doi: 10.1080/07350015.2015.1086655.

M. W. McCracken and S. Ng. FRED-QD: A quarterly database for macroeconomic research. Technical report, 2020.

L. Monteforte and G. Moretti. Real-time forecasts of inflation: The role of financial variables. *Journal of Forecasting*, 32(1):51–61, mar 2012. doi: 10.1002/for.1250.

J. Morley. Macro-finance linkages. *Journal of Economic Surveys*, 30(4):698–711, apr 2015. doi: 10.1111/joes.12108.

A. Onatski. Asymptotics of the principal components estimator of large factor models with weakly influential factors. *Journal of Econometrics*, 168(2):244–258, June 2012. ISSN 0304-4076. doi: 10.1016/j.jeconom.2012.01.034.

L. Paranhos. Predicting inflation with neural networks, 2021.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. *Proceedings of Machine Learning Research*, 28(3):1310–1318, 2013.

J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott. Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data. *Chaos*, 27(12), 2017. ISSN 10541500. doi: 10.1063/1.5010300.

J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott. Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach. *Physical Review Letters*, 120(2): 24102, 2018. ISSN 10797114. doi: 10.1103/PhysRevLett.120.024102. URL https://doi.org/10.1103/PhysRevLett.120.024102.

D. Qin, S. van Huellen, Q. C. Wang, and T. Moraitis. Algorithmic modelling of financial conditions for macro predictive purposes: Pilot application to USA data. *Econometrics*, 10(2):22, apr 2022. doi: 10.3390/econometrics10020022.

R. Quaedvlieg. Multi-Horizon Forecast Comparison. *Journal of Business & Economic Statistics*, 39(1): 40–53, Jan. 2021. ISSN 0735-0015, 1537-2707. doi: 10.1080/07350015.2019.

A. Rodan and P. Tino. Minimum complexity echo state network. *IEEE Transactions on Neural Networks*, 22(1):131–44, jan 2011. ISSN 1941-0093.

H. Salehinejad, J. Baarbe, S. Sankar, J. Barfett, E. Colak, and S. Valaee. Recent advances in recurrent neural networks. 2017.

T. J. Sargent, C. A. Sims, et al. Business cycle modeling without pretending to have too much a priori economic theory. *New methods in business cycle research*, 1:145–168, 1977.

F. Schorfheide, D. Song, and A. Yaron. Identifying long-run risks: A bayesian mixed-frequency approach. *Econometrica*, 86(2):617–654, 2018.

J. H. Stock and M. W. Watson. Evidence on structural instability in macroeconomic time series relations. *Journal of Business & Economic Statistics*, 14(1):11–30, 1996. doi: 10.1080/07350015.1996.10524626.

J. H. Stock and M. W. Watson. Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics*, 20(2):147–162, apr 2002. doi: 10.1198/073500102317351921.

J. H. Stock and M. W. Watson. Forecasting with Many Predictors. In G. Elliot, C. W. Granger, and A. Timmermann, editors, *Handbook of Economic Forecasting*, volume 1. Elsevier edition, 2006.

J. H. Stock and M. W. Watson. Dynamic factor models, factor-augmented vector autoregressions, and structural vector autoregressions in macroeconomics. In *Handbook of macroeconomics*, volume 2, pages 415–525. Elsevier, 2016.

G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115: 100–123, 2019. ISSN 18792782. doi: 10.1016/j.neunet.2019.03.005. URL https://doi.org/10.1016/j.neunet.2019.03.005.

L. van der Maaten. Learning a Parametric Embedding by Preserving Local Structure. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, pages 384–391. PMLR, Apr. 2009.

S. van Huellen, D. Qin, S. Lu, H. Wang, Q. C. Wang, and T. Moraitis. Modelling opportunity cost effects in money demand due to openness. *International Journal of Finance & Economics*, 27(1): 697–744, aug 2020. doi: 10.1002/ijfe.2175.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

M. W. Watson and R. F. Engle. Alternative algorithms for the estimation of dynamic factor, mimic and varying coefficient regression models. *Journal of Econometrics*, 23(3):385–400, 1983.

A. Wikner, J. Pathak, B. R. Hunt, I. Szunyogh, M. Girvan, and E. Ott. Using data assimilation to train a hybrid forecast system that combines machine-learning and knowledge-based components. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(5):53114, 2021.

P. C. Young. *Recursive estimation and time-series analysis: an introduction.* Springer Science & Business Media, 2012.

7 Appendix

7.1 ESN Implementation

Cross-validation. Because the initial cross-validation of λ uses an extended sample to try and improve generalization – specifically, our concern is for the fixed estimation setups – we use two slightly different approaches:

- In all setups – fixed, expanding, rolling – the *initial* ridge penalty cross-validation is done on the extended sample (starting January 1st, 1975 instead of January 1st, 1990). We construct 10 folds with 5 out-of-sample observations starting from the end of the sample. Each fold and out-of-sample observation set is re-normalized.
- Only in the expanding and rolling setups, for each subsequent window (the ones that now include at least one testing observation), we use the true sample (starting January 1st, 1990) and construct 5 folds, again with 5 out-of-sample observations. This is done to keep cross-validation computational complexity low and avoid making some folds too small, which could hurt larger MFESN models.

In practice, simple experiments show that there is not much difference between using 5 or 10 folds in the initial cross-validation.

7.2 MIDAS Implementation

While the MIDAS regression framework is straightforward to discuss in terms of equations, some care must be taken when implementing it computationally. A key assumption that can be imposed is that the integer frequencies $\boldsymbol{\kappa} := \{\kappa_1, \ldots, \kappa_L\}$ of M regressors are such that $\kappa_{\max} := \max(\boldsymbol{\kappa})$ is a multiple of each of the κ_l, $l \in [L]$. In this case MIDAS parameter estimation can be written in matrix form, which allows for efficient numerical implementation, which we spell out in the following paragraphs.

Let $q_l = \kappa_{\max}/\kappa_l$, $l \in [L]$ denote the frequency ratios and define $\boldsymbol{y} := (y_1, y_2, \ldots, y_T)^\top$ the vector of target observations, where T is the sample length in reference time scale. Additionally, let $\boldsymbol{x}_l := (x_{l,1}, x_{l,2}, \ldots, x_{l,T_l})^\top$ be $T_l = T \cdot \kappa_l$ long vector which consists of observations of the l-th covariate x_l released with frequency κ_l. In order for the parameters of the MIDAS model in (4.4) to be identifiable, we assume that

$$T > 1 + p + \sum_{l=1}^{L} \left\lceil \frac{K_l}{\kappa_l} \right\rceil .$$

Since $\kappa_{\max}$ is a multiple of each of the L frequencies, for each series we introduce

$$\mathbf{Y} = \boldsymbol{y} \otimes \boldsymbol{i}_{\kappa_{\max}}, \quad \mathbf{X}_l = \boldsymbol{x}_l \otimes \boldsymbol{i}_{q_l},$$

where $\boldsymbol{i}_{q_l}$ and $\boldsymbol{i}_{\kappa_{\max}}$ are vectors of ones of lengths q_l and $\kappa_{\max}$, respectively. In the absence of missing observations, we have that $\mathbf{Y}, \mathbf{X}_l \in \mathbb{R}^{T_{\max}}$ with $T_{\max} = T \cdot \kappa_{\max}$ observations. We now construct preliminary regression matrices such that their maximal rows number is $T_{\max}$ without accounting for the lags structure of both the target (autoregressive lags) and regressors (MIDAS lags) and we introduce zeros where no observations are available.[11] Define for $p \geq 1$

$$Y_p = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ y_1 & 0 & \cdots & 0 \\ y_2 & y_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ y_{T-2} & y_{T-3} & \cdots & y_{T-p-1} \\ y_{T-1} & y_{T-2} & \cdots & y_{T-p} \end{pmatrix} \otimes \boldsymbol{i}_{\kappa_{\max}} \tag{7.1}$$

[11] At the time of implementation of this procedure in any convenient coding environment it is more natural to introduce placeholders instead and to perform the subsequently discussed truncation via matrix manipulation rather than by using matrix multiplication.

and for $K_l \geq 0$

$$X_{K_l} = \begin{pmatrix} x_{l,1} & 0 & \cdots & 0 \\ x_{l,2} & x_{l,1} & \cdots & 0 \\ x_{l,3} & x_{l,2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ x_{l,T_l-1} & x_{l,T_l-2} & \cdots & x_{l,T_l-K_l-1} \\ x_{l,T_l} & x_{l,T_l-1} & \cdots & x_{l,T_l-K_l} \end{pmatrix} \otimes \boldsymbol{i}_{q_l}. \tag{7.2}$$

In the special case $p = 0$ (the MIDAS model in this case is called *static*, since it does not contain an autoregressive term) we take Y_p as empty. We now follow by noticing that one should not use Y_p and X_{K_l} as autoregressive and mixed-frequency regression matrices, respectively, due to the fact that some observations are missing. To overcome this we introduce

$$s := \max\left\{p, \left\lceil \frac{K_1}{q_1} \right\rceil, \ldots, \left\lceil \frac{K_L}{q_L} \right\rceil\right\} \cdot \kappa_{\max} \tag{7.3}$$

and the so-called upper truncation (selection) matrix

$$U = \begin{pmatrix} \mathbb{O}_{T_{\max}-s+1,s-1} & \mathbb{I}_{T_{\max}-s+1,T_{\max}-s+1} \end{pmatrix} \tag{7.4}$$

and we obtain the following required response vector and regression matrices

$$\begin{aligned} \mathbf{Y}^{\text{resp}} &= U\mathbf{Y}, \\ Y_p^{\text{reg}} &= UY_p, \\ X_{K_l}^{\text{reg}} &= UX_{K_l}, \\ \mathbf{Z}^{\text{reg}} &= \begin{pmatrix} Y_p^{\text{reg}} & X_{K_1}^{\text{reg}} & \cdots & X_{K_L}^{\text{reg}} \end{pmatrix}, \end{aligned}$$

where Y_p^{reg} is empty whenever $p = 0$.

We can now observe that $\mathbf{Y}^{\text{resp}}$ and $\mathbf{Z}^{\text{reg}}$ are sufficient to construct all MIDAS forecasting and nowcasting regressions. In practice, some care needs to be taken to make sure that data is correctly aligned: for example, in the case of nowcasting exercise regressors in $\mathbf{Z}^{\text{reg}}$ and targets in $\mathbf{Y}^{\text{resp}}$ have to be aligned in a different fashion than in the case of forecasting exercises. Provided the aligned data is executed correctly, the estimation of MIDAS parameters can be carried out efficiently. An important thing to mention is that the truncation with the help of s in (7.3) may be too restrictive, as it may lead to excluding up to $K_{\max}-1$ rows from $\mathbf{Z}^{\text{reg}}$ that could be used for estimation. This can be avoided at the time of implementing. In our repository available at `RCEconModelling/Reservoir-Computing-for-Macroeconomic-Modelling` we consider this detail and exclude from the final regression matrices only those rows which cannot be used due to the lag requirements in the model. We warn the reader that this comes at a cost, namely the codes are lengthier and less elegant.

7.3 Mixed-frequency DFM Implementation

This section gives additional details on implementing non-homogeneous dynamic factor models, such as the mixed frequency model introduced in the main text. We notice that the conditioning notation in this section should not be confused with our temporal notation in Definition 2.1.

Kalman Filtering and Computational Complexity. The sufficient statistics of the posterior distribution of the latent factor $\boldsymbol{f}_t|\boldsymbol{y}_{0:t}$ can be updated recursively by the Kalman filter updates in the linear Gaussian setting. First, propagate the prior

$$\begin{aligned} \widehat{\boldsymbol{f}}_{t+1|t,\theta} &= A_\theta \widehat{\boldsymbol{f}}_{t|t,\theta} \\ \widehat{\Sigma}_{t+1|t,\theta} &= A_\theta \widehat{\Sigma}_{t+1|t,\theta} A_\theta^\top + S_{t+1,\theta} S_{t+1,\theta}^\top. \end{aligned}$$

Compute the innovation covariance

$$\Gamma_{t+1,\theta} = \Lambda_{t+1}\widehat{\Sigma}_{t+1|t,\theta}\Lambda_{t+1}^{\top} + R_\theta R_\theta^{\top}$$

and the Kalman gain

$$K_{t+1,\theta} = \widehat{\Sigma}_{t+1|t,\theta}\Lambda_{t+1,\theta}^{\top}\Gamma_{t+1,\theta}^{-1}.$$

Then, update the statistics with the new information y_{t+1},

$$\begin{aligned}\widehat{\boldsymbol{f}}_{t+1|t+1,\theta} &= \widehat{\boldsymbol{f}}_{t+1|t,\theta} - K_{t+1,\theta}\left(y_{t+1} - \Lambda_{t+1,\theta}\widehat{\boldsymbol{f}}_{t+1|t,\theta}\right)\\ \widehat{\Sigma}_{t+1|t+1,\theta} &= \left(\mathbb{I} - K_{t+1,\theta}\Lambda_{t+1,\theta}\right)\widehat{\Sigma}_{t+1|t,\theta}.\end{aligned}$$

Notice that the inverse of the log-determinant of the innovation matrices $\Gamma_{t,\theta}$ are required for computing the Kalman gains and the marginal log-likelihood, respectively, which yield a cubic computational complexity in the dimension of the observation process. Alternatively, one can apply matrix inversion or determinant lemmas to obtain a computational complexity that is cubic in the dimension of the Markovian factor process $\boldsymbol{f}_t$. For an alternative approach in high-dimensions that imposes a dynamic factor structure after a projection of the observations onto a low-dimensional space, see Jungbacker and Koopman (2015).

Model Selection. The model parameters θ are learned to jointly maximize the log-likelihood of the observed data for all frequencies. This is in contrast to the parameter estimation approach for MIDAS that maximizes the log-likelihood of the low-frequency process only conditional on observing the high-frequency time series. We remark that a different log-likelihood weighting for the different frequencies in DFMs has been suggested in Blasques et al. (2016), but requires cross-validation to optimize such weightings. Nevertheless, the introduced DFM contain several hyperparameters that need to be chosen, such as the latent state space dimension k or the order p of the latent Markov process. One possibility is to select such hyperparameters by evaluating the low-frequency predictions on a validation set. Approaches for choosing the dimensions of the latent factor process have been under-explored in the mixed-frequency setting, but see Bai and Ng (2007), Hallin and Liška (2007) for possible criteria in general dynamic factor models. In our implementation, we choose $p = 1$, as this allows for a differentiable model parametrization with stationary factor dynamics. We set $k = 5$ for the small dataset and $k = 10$ for the medium dataset.

Parameter Estimation and Forecasting. Based on the results from the Kalman filtering recursions, the model parameters θ are learned by maximizing the marginal-log-likelihood using $\ell_t(\theta) = -\log p_\theta(\boldsymbol{y}_{0:t}) = -\sum_{s=0}^{t}\log p_\theta(\boldsymbol{y}_s|\boldsymbol{y}_{0:s-1})$ where $p_\theta(\boldsymbol{y}_s|\boldsymbol{y}_{0:s-1})$ is Gaussian with mean $\Lambda_{s,\theta}\widehat{\boldsymbol{f}}_{s|s-1,\theta}$ and covariance $\Gamma_{s,\theta}$. Gradients of $\ell_t(\theta)$ can be computed using algorithmic differentiation.

For fixed $\theta \in \Theta$ and $h \in \mathbb{N}$, let

$$\mu_{t+h|t,\theta}(\boldsymbol{y}_{t+h}|\boldsymbol{y}_{0:t}) = \int g_{t+h,\theta}(\boldsymbol{y}_{t+h}|\boldsymbol{f}_{t+h})\prod_{\ell=1}^{h} h_{t+\ell,\theta}(\boldsymbol{f}_{t+\ell}|\boldsymbol{f}_{t+\ell-1})\mathrm{d}\boldsymbol{f}_{t+\ell}\pi_{t|t,\theta}(\boldsymbol{f}_t|\boldsymbol{y}_{0:t})\mathrm{d}\boldsymbol{f}_t$$

be the h-step predictive distribution of the data, while $\pi_{t|t,\theta}(\boldsymbol{f}_t|\boldsymbol{y}_{0:t})$ is the filtering distribution of the latent state $\boldsymbol{f}_t|\boldsymbol{y}_{0:t}$. The mean of $\mu_{t+h|t,\theta}(\cdot|\boldsymbol{y}_{0:t})$ is $\widehat{\boldsymbol{y}}_{t+h|t,\theta} = \mathbb{E}_\theta\left[\boldsymbol{y}_{t+h}|\boldsymbol{y}_{0:t}\right]$. For some $t, \tau \geq 0$, let us write $\widehat{f}_{t+\tau|t,\theta} = \mathbb{E}_\theta[\boldsymbol{f}_{t+\tau}|\boldsymbol{y}_{0:t}]$ and $\Sigma_{t+\tau|t,\theta} = \mathrm{Cov}_\theta[\boldsymbol{f}_{t+\tau} - \widehat{\boldsymbol{f}}_{t+\tau|t,\theta}|\boldsymbol{y}_{0:t}]$ for the mean and covariance of the latent process, respectively. For linear Gaussian dynamics, Kalman filtering allows for computing the filtered mean $\widehat{\boldsymbol{f}}_{t|t,\theta}$ and covariance matrices $\widehat{\Sigma}_{t|t,\theta}$ analytically.

For fixed θ, the τ-step ahead prediction function $H_{t,\theta}^{\tau}(\boldsymbol{y}_{0:t}) = \widehat{\boldsymbol{y}}_{t+\tau|t,\theta} = \Lambda_{t+\tau,\theta}\widehat{\boldsymbol{f}}_{t+\tau|t,\theta}$ is linear due to the Kalman filtering recursion. For $s \leq t$, consider also the prediction $H_{s,t}^{\star\tau}(\boldsymbol{y}_{0:t}) = \mathbb{E}_{\theta^\star(\boldsymbol{y}_{0:s})}[\boldsymbol{y}_{t+\tau}|\boldsymbol{y}_{0:t}]$ that is based on the dataset $\boldsymbol{y}_{0:t}$. where $\theta^\star(\boldsymbol{y}_{0:s}) = \arg\min_\theta \ell_s(\theta)$ maximizes the marginal likelihood of the data $\boldsymbol{y}_{0:s}$ only. This setting allows to implement the different parameter estimation setups from Section 3.2.1. For instance, the fixed parameter setup corresponds to fixing s which yields a fixed training set $\boldsymbol{y}_{0:s}$ to estimate θ. In the expanding window setup, both s and t are expanded, while a rolling window setting updates the set $\boldsymbol{y}_{0:s}$ by rolling over the data.

7.4 High-Frequency Forecasts

In order to better understand how the use of high-frequency data impacts forecasting, as an additional empirical experiment we investigate high-frequency (HF) forecasts of all models for in the Small-MD dataset. We restrict our analysis to this dataset because the computational burden to construct HF forecasts can be high: when using daily data and using our suggested 24 days-per-month interpolation, one quarter amounts to 72 daily frequency observations, which means HF forecasts can involve thousands of data points, and for DFM and M-MFESN models this setup can be quite computationally onerous.

Constructing HF forecasts with MIDAS is trivial once the aggregation weights have been estimated, even thought a practical implementation requires care in constructing the appropriate lag matrices. Recall for Section 4.1 that the MIDAS equation with L regressors $\{x^{(l)}_{\tau_l}\}_{l=1}^{L}$ can be written as

$$y_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{l=1}^{L} \beta_l \sum_{k_l=0}^{K_l} \varphi(\boldsymbol{\theta}_l, k_l) x^{(l)}_{t,-k_l|\kappa_l} + \epsilon_t .$$

For clarity, we suppress the dynamic autoregressive component, as it has the same frequency as the target. Now assume that we include $n_{(\mathrm{m})}$ monthly and $n_{(\mathrm{d})}$ daily frequency regressors in the model that are sampled $\kappa_m = 3$ and $\kappa_d = 72$ times per quarter and hence $\kappa_{\max} = 72$. Therefore we can partition the regression above in the following way

$$y_t = \sum_{i=1}^{n_{(\mathrm{m})}} \beta_i \sum_{k_i=0}^{K_i} \varphi(\boldsymbol{\theta}_i, k_i) x^{(i)}_{t,-k_i|3} + \sum_{j=1}^{n_{(\mathrm{d})}} \beta_j \sum_{k_j=0}^{K_j} \varphi(\boldsymbol{\theta}_j, k_j) x^{(j)}_{t,-k_j|72} + \epsilon_t .$$

such that $L = n_{(\mathrm{m})} + n_{(\mathrm{d})}$.

Assuming parameter estimates $\{(\widehat{\beta}_i, \widehat{\boldsymbol{\theta}}_i)\}_{i=1}^{n_{(\mathrm{m})}}$ and $\{(\widehat{\beta}_j, \widehat{\boldsymbol{\theta}}_j)\}_{j=1}^{n_{(\mathrm{d})}}$ are available, the HF forecast $\widehat{y}_{t+1,s|72}$ is given by

$$\widehat{y}_{t+1,s|72} = \sum_{i=1}^{L_m} \widehat{\beta}_i \sum_{k_i=0}^{K_i} \varphi(\widehat{\boldsymbol{\theta}}_i, k_i) x^{(i)}_{t-1,2+\lfloor (s+1)/24 \rfloor - k_i|3} + \sum_{j=1}^{L_d} \widehat{\beta}_j \sum_{k_j=0}^{K_j} \varphi(\widehat{\boldsymbol{\theta}}_j, k_j) x^{(j)}_{t,s-k_j|72} .$$

For DFMs, high-frequency forecasts can be constructed using (4.10) and (4.11) to iterate factors forward in time and then aggregating them according to estimated loadings or a weighting scheme.

Multi-frequency ESN models are also able to yield high-frequency forecasts in a trivial manner. For simplicity, let us consider the case as in Example 3.1 of an aligned S-MFESN model that has been fitted to a quarterly target with monthly and daily data. The reservoir is run in high-frequency, $\kappa_{\max}$ steps per quarter, according to state equation

$$\boldsymbol{x}^{(\mathtt{m},\mathtt{d})}_{t,s|72} = \alpha \boldsymbol{x}^{(\mathtt{m},\mathtt{d})}_{t,s-1|72} + (1-\alpha)\sigma(A\boldsymbol{x}^{(\mathtt{m},\mathtt{d})}_{t,s-1|72} + C\boldsymbol{z}^{(\mathtt{m},\mathtt{d})}_{t,s|72} + \boldsymbol{\zeta}).$$

Suppose a coefficient matrix $\widehat{W}$ has been estimated. Then, as states between low-frequency periods t and $t+1$ are collected, we can immediately construct the high-frequency one-step-ahead forecasts by setting

$$\widetilde{y}_{t+1,s|72} = \widehat{W}^{\top} \boldsymbol{x}^{(\mathtt{m},\mathtt{d})}_{t,s|72} .$$

For M-MFESN models HF forecasts require slightly more care. For example, when dealing the multi-reservoir MFESN model of Example 3.2, we must repeat the most recent monthly state at daily frequency correctly.

7.5 Robustness Analysis

7.5.1 MIDAS

As we discuss briefly in the main text, parameter optimization is a principal problem in implementing any MIDAS model. Even though explicit formulas exist for both gradient and Hessian of the MIDAS loss objective when an Almon weighting scheme is used (see Kostrov (2021)), there is no known guarantee that the loss itself is convex or even locally convex. In practice, for a given starting point (or point set) a numerical optimizer might only converge to a local minimum.

We observe this in practice, and we explore its effects on the robustness of MIDAS forecasts. We report summary results for our simulations in Figure 17. Our proposal is, given a MIDAS model specification and a set of starting points for evaluating the loss, to run an optimizer (for example, L-BFGS-B with explicit gradient) and select the smallest local minimum. By repeating this procedure multiple times, we collect a set of MIDAS parameters and study both the variation between the parameter vectors and the implied one-step ahead forecasts.

To be precise, our procedure is as follows:

1. For a total of B repetitions:

   (a) Choose M initialization points for the optimizer. We draw 64 points inside the hypercube of edge length 0.025 using a low-discrepancy Sobol sequence. The choice of a down-scaled hypercube as a domain comes from the empirical fact that the Almon exponential scheme may produce extremely large values even for relatively small coefficients. A straightforward way too see this is to notice that given any arbitrary small value for θ_1 and θ_2 in (4.3), for lag index k sufficiently large weight $\exp(\theta_1 k + \theta_2 k^2)$ will overflow at any given numerical precision. This means that one should adjust the MIDAS optimization domain based on the number of lags in the model.

   (b) For each initialization point, run the optimizer of choice.

   (c) Among the resulting M (local) loss minimizers, select and store the one with the lowest loss value.

2. With the resulting B stored minimizer:

   - Construct a low-dimensional projection of the high-dimensional minima to see their relative location in the parameter space and to compare their gradient and loss values, see Figure 17 (a)-(b).
   - Use each minimizer to produce MIDAS one-step ahead forecasts and plot quantile frequency plots of the forecast variation due to initialization; see Figure 17 (c).

Figure 17 shows that the best minimizers among initial Sobol sets are clustered together. To construct this 2D projection of the high-dimensional Almon coefficient space (including autoregressive lags and intercept), we use the well-known t-SNE procedure developed in van der Maaten (2009), which is an unsupervised dimensionality reduction algorithm capable of preserving the latent high-dimensional structure. This approach naturally implies that the Euclidean distances in the plot are suggestive of ”clustering” rather than the actual latent distance between points. In Figures 17 (a)-(b), we see that the L-BFGS-B optimizer with explicit gradient achieves good convergence in terms of gradient norm and also that the resulting cluster of minimizers has close loss values. However, one can see that there is no single loss minimum: Figure 17 instead suggests that the local structure of the MIDAS loss function is very uneven, and therefore many distinct local minima can be achieved even when choosing a large number of initialization values for the optimizer. This means that the “multistart” strategy suggested in Kostrov (2021) to alleviate issues in MIDAS model estimation is insufficient.

The effects of non-negligible variation in parameter values on forecasts appear to be significant. Looking at Figure 17 (c), we can see wide frequency bands for the one-step ahead forecasts constructed using the Small-MD dataset and fixed parameter values. In particular, the Financial Crisis period seems to induce larger variation in forecasts, consistent with the intuition that data with larger variation causes stronger model sensitivity when making forecasts.

7.5.2 MFESN

Since ESN models, and thus MFESN models, require random sampling of parameter matrices, the size of which is often large, there is inherent variability in any ESN model forecast. In theory, because all MFESN parameters in matrices $(A, C, \boldsymbol{\zeta})$ are drawn independently of each other, one could try to decompose the variance of any MFESN into the share due to parameter sampling and the share due to data sampling. Unfortunately, in practice, such decomposition is hard to derive. Cross-validation of ridge penalties and rolling and expanding window estimation are non-trivial data-dependent operations that complicate inference. In this work, we limit ourselves to numerically evaluating the effect of reservoir coefficient sampling on MFESN forecast variance.

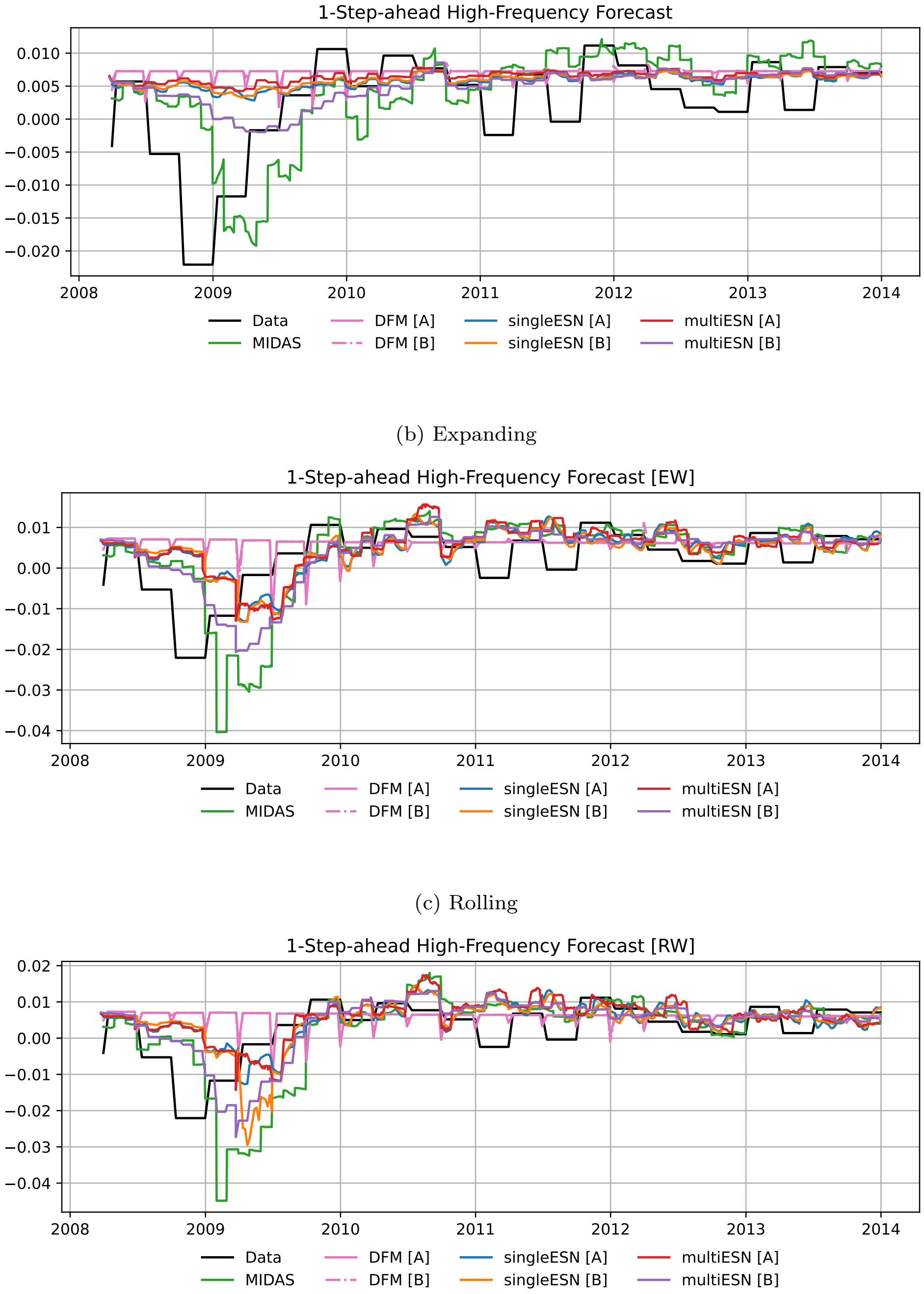
Our approach is straightforward: given an MFESN model specification, c.f. Table 5.2, and a forecasting setup (fixed parameters, expanding or rolling window), we resample the reservoir design matrices, perform cross-validation and possibly train-test sample windowing, and finally construct point-wise forecasts. Once a sufficient large set of resampling forecasts have been computed, we plot frequency intervals in Figures 19 and 21. From Figure 19, we can see that the single-reservoir MFESN model with reservoir size 30 produces forecasts with a meaningful amount of variability induced by parameter resampling. Forecasts exhibit more variation when using an expanding or rolling window estimation strategy, even though the overall forecasts align with the GDP realizations. A similar discussion to that of MIDAS applies here: forecast sensitivity increases with underlying data variation, exacerbated in periods of systemic economic crisis.

Figure 21 suggests that larger MFESN models produce significantly more stable forecasts regarding model resampling. Note that the M-MFESN model [A] has a monthly frequency reservoir that is approximately 3 times the size of the S-MFESN model [A]. This stability is preserved even in expanding or rolling window settings, even though a slightly higher variation is apparent at the height of the 2008 Financial Crisis. We hypothesize that this reduction in variance due to model parameter sampling is due to the concentration of measure phenomena that prevail in high-dimensional spaces. Figure 21 suggests that larger MFESN models produce significantly more stable forecasts regarding model resampling. Note that the M-MFESN model [A] has a monthly frequency reservoir that is approximately 3 times the size of the S-MFESN model [A]. This stability is preserved even in expanding or rolling window settings, even though a slightly higher variation is apparent at the height of the 2008 Financial Crisis. We hypothesize that this reduction in variance due to model parameter sampling is due to the concentration of measure phenomena that prevail in high-dimensional spaces.

7.6 Additional Figures

Figure 15: 1-Step-ahead High-Frequency GDP Forecasting – 2007 Sample – Small-MD Dataset

(a) Fixed

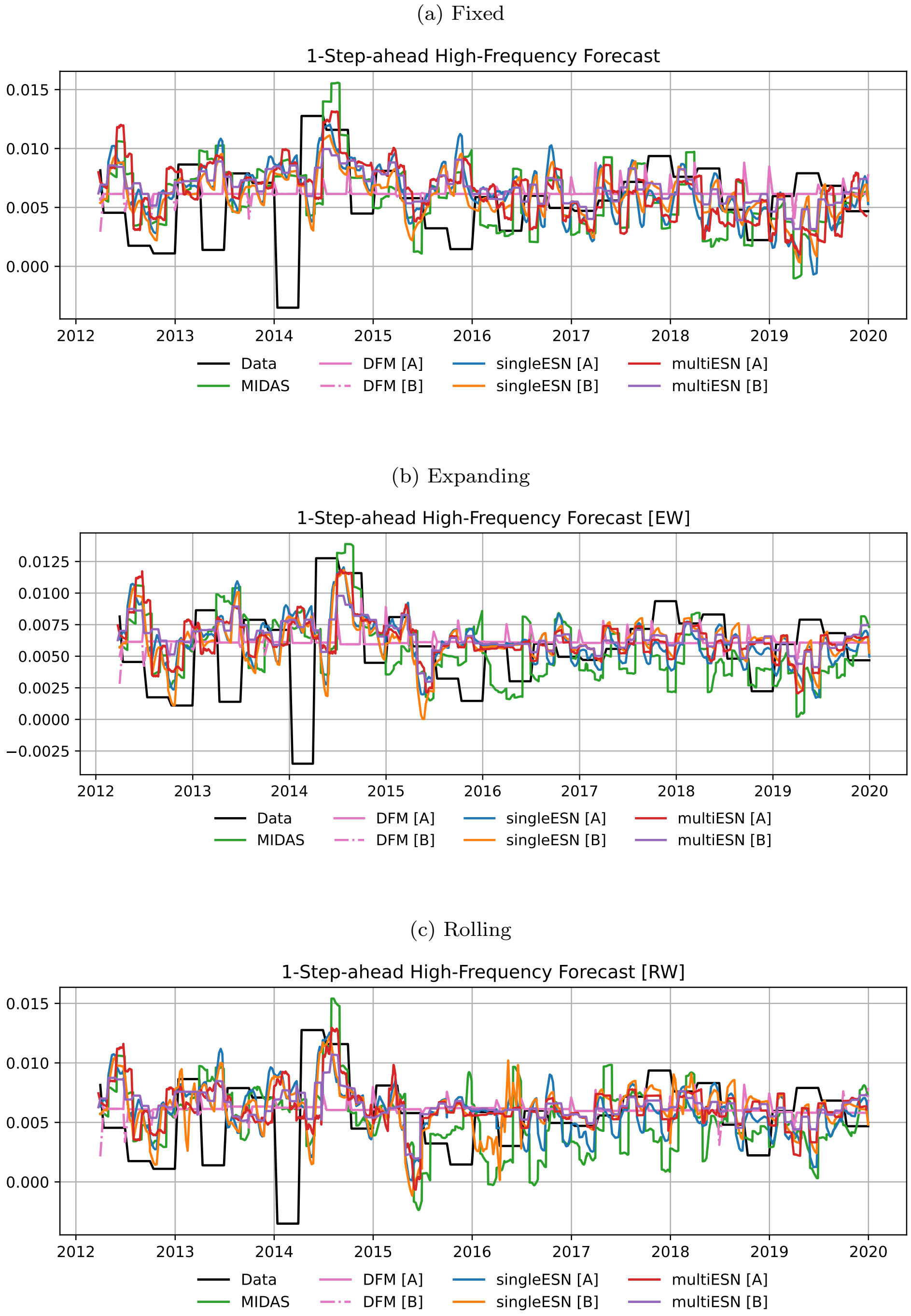


(b) Expanding

(c) Rolling

Note: Forecasts for the 2007 sample are presented up to Q4 2013 to better display the high-frequency behavior of models during the Financial Crisis period.

Figure 16: 1-Step-ahead High-Frequency GDP Forecasting – 2011 Sample – Small-MD Dataset



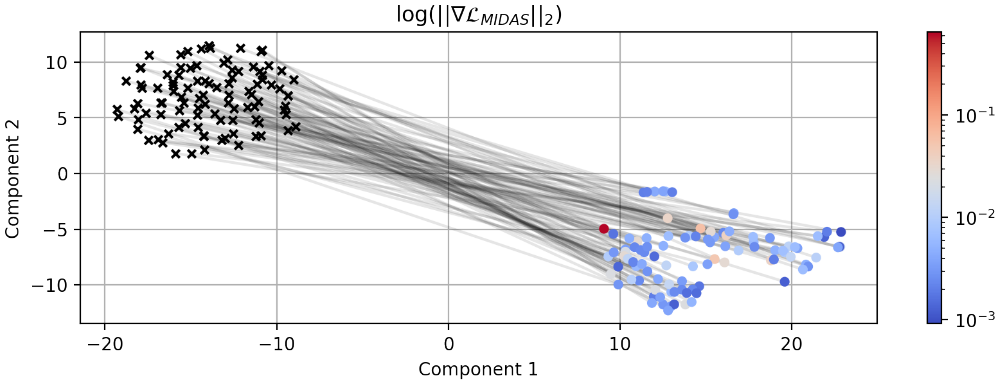
(a) Fixed

(b) Expanding

(c) Rolling

Figure 17: MIDAS Robustness Plots – 2007 Sample – Small-MD Dataset

(a) MIDAS Loss t-SNE Embedding: Gradient Norm



(b) MIDAS Loss t-SNE Embedding: Loss Norm

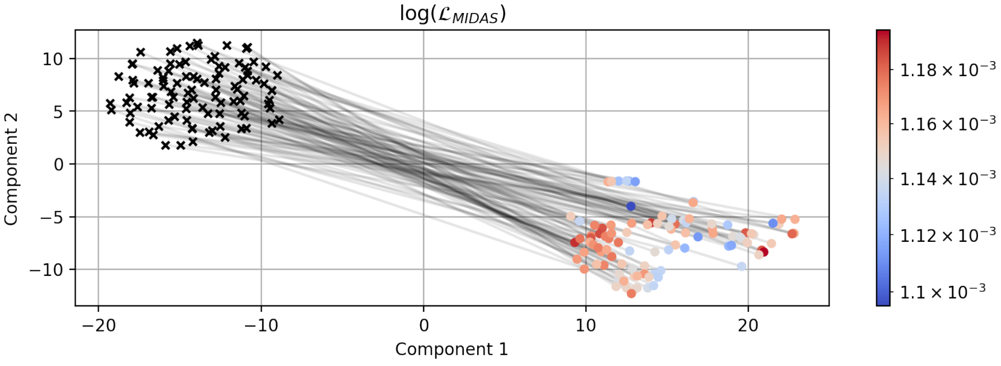
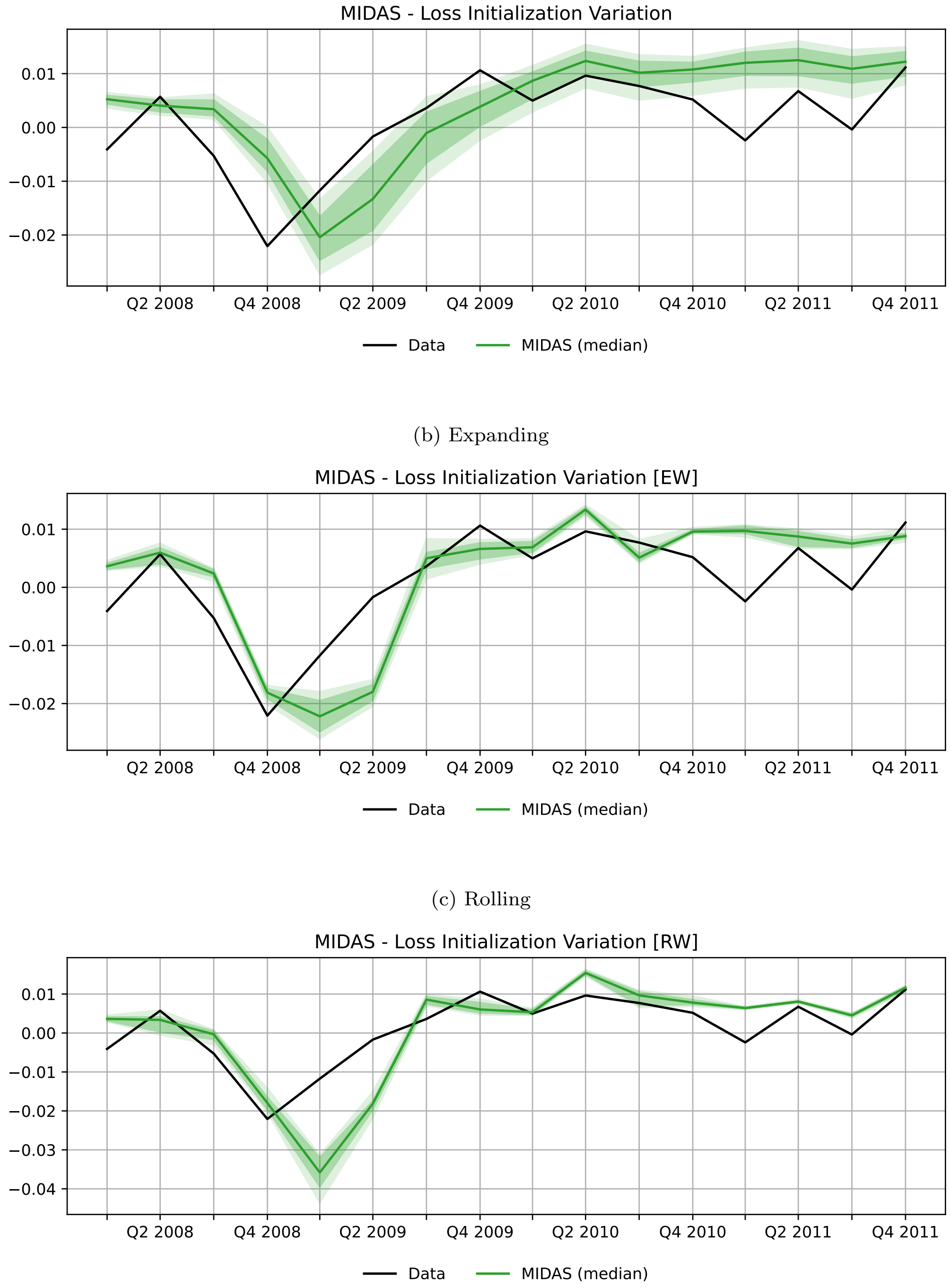


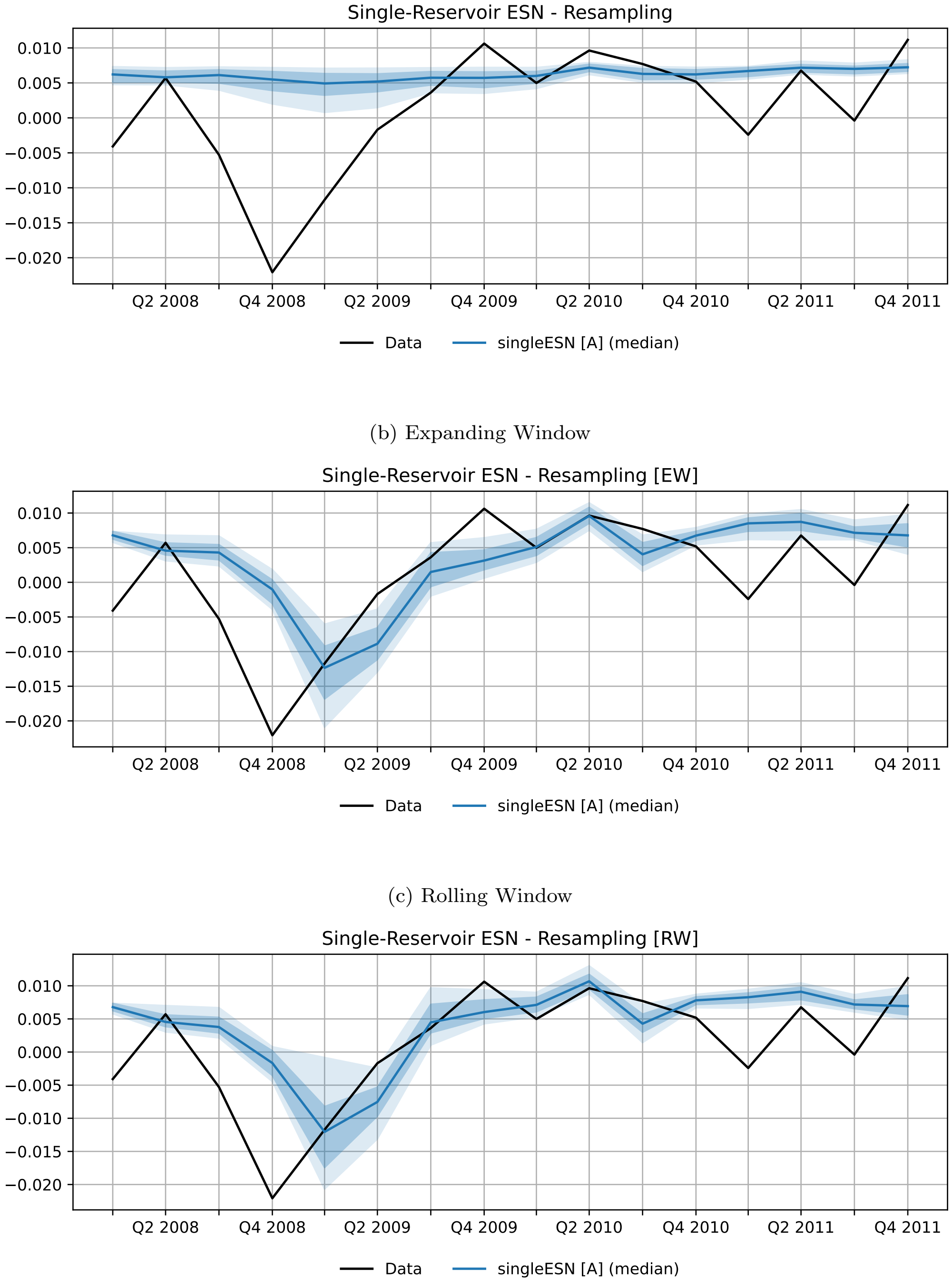
Figure 18: MIDAS Robustness Plots – 2007 Sample – Small-MD Dataset


(a) Fixed Parameters

(b) Expanding

(c) Rolling

Figure 19: ESN Robustness Plots – 2007 Sample – Small-MD Dataset



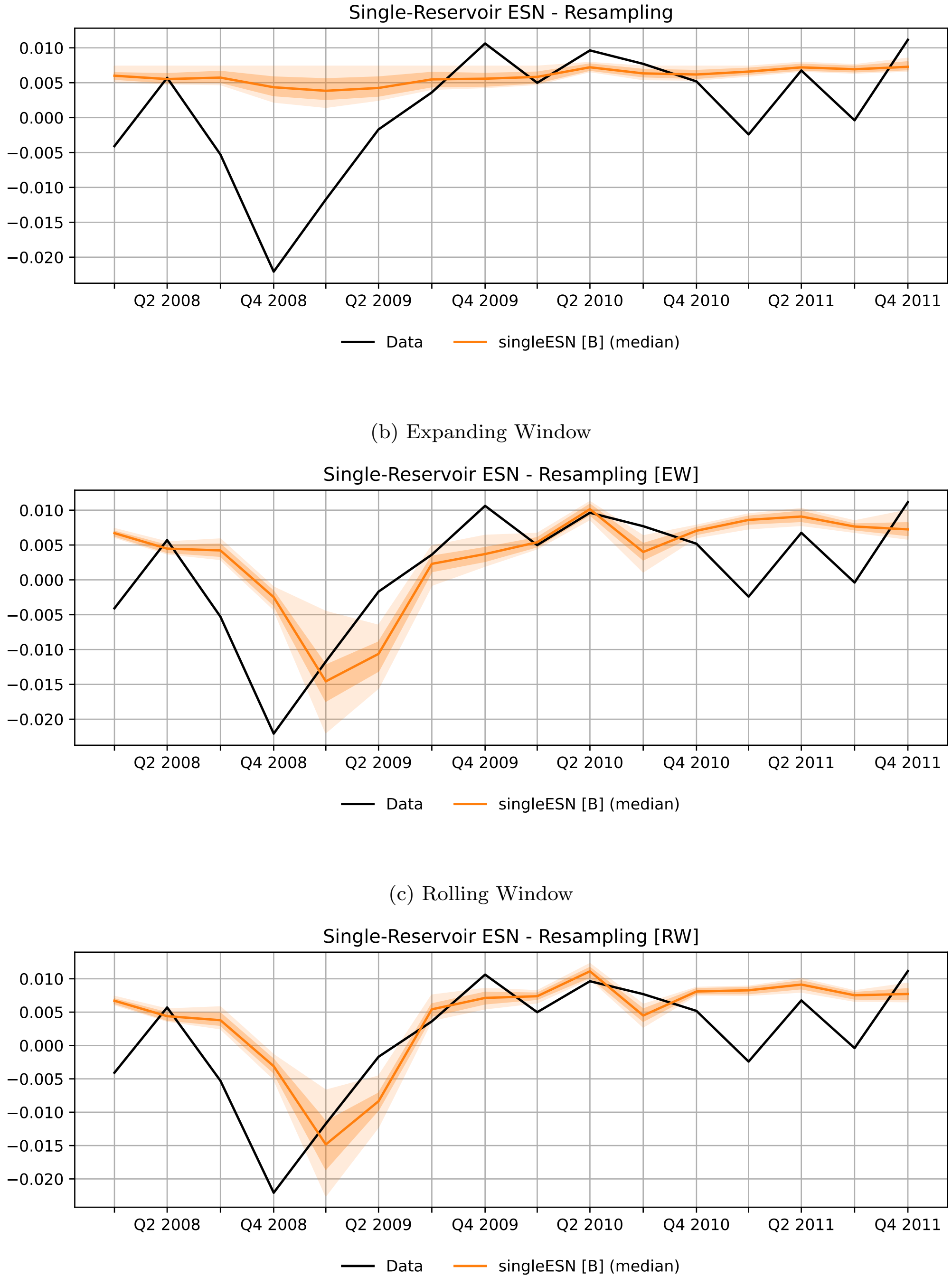
(a) Fixed Parameters

(b) Expanding Window

(c) Rolling Window

Figure 20: ESN Robustness Plots – 2007 Sample – Small-MD Dataset

(a) Fixed Parameters

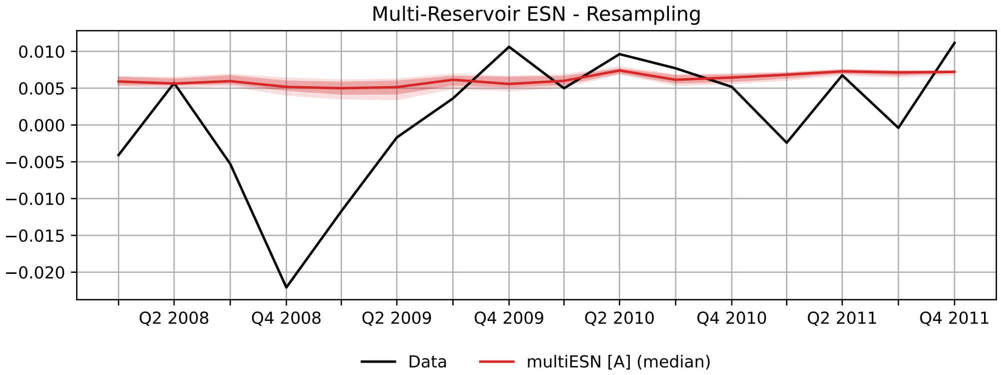


(b) Expanding Window

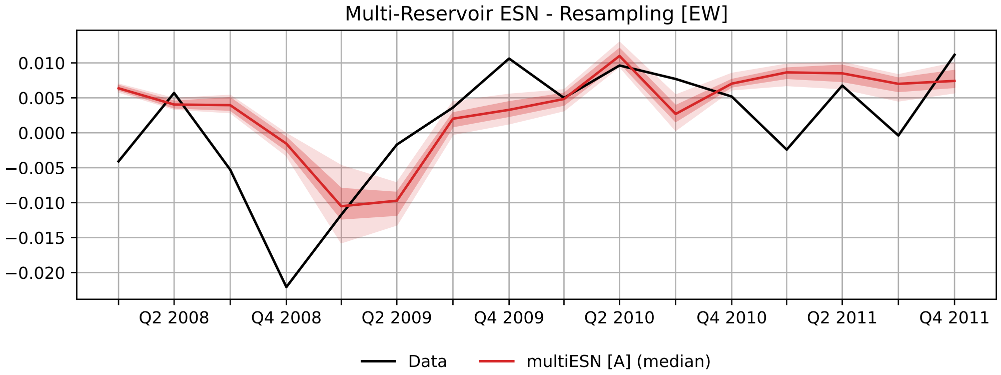
(c) Rolling Window

Figure 21: ESN Robustness Plots – 2007 Sample – Small-MD Dataset

(a) Fixed Parameters



(b) Expanding Window



(c) Rolling Window

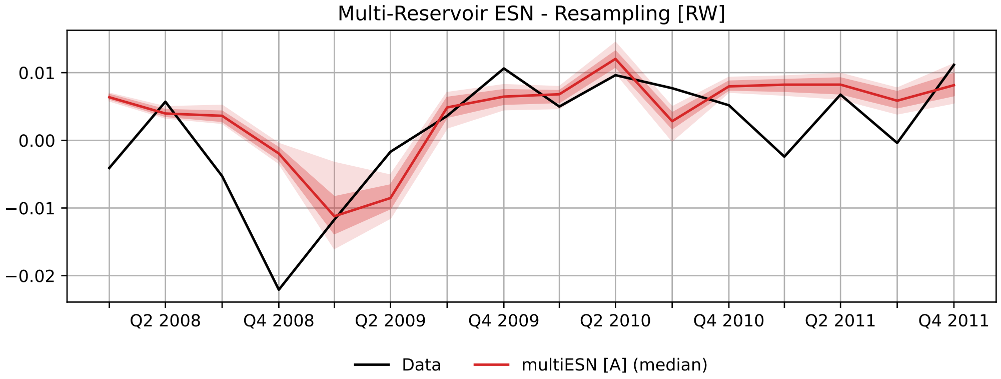
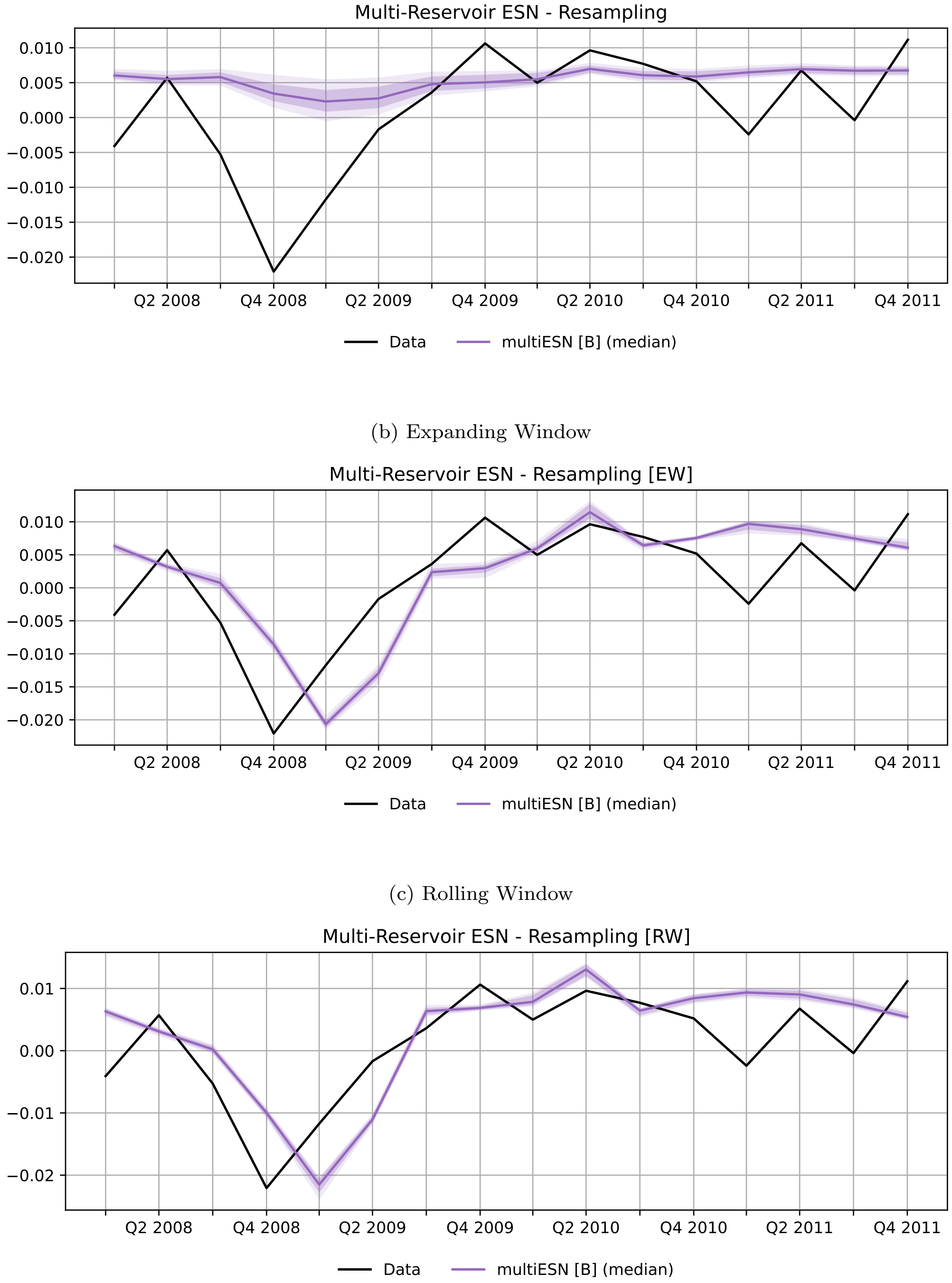


Figure 22: ESN Robustness Plots – 2007 Sample – Small-MD Dataset

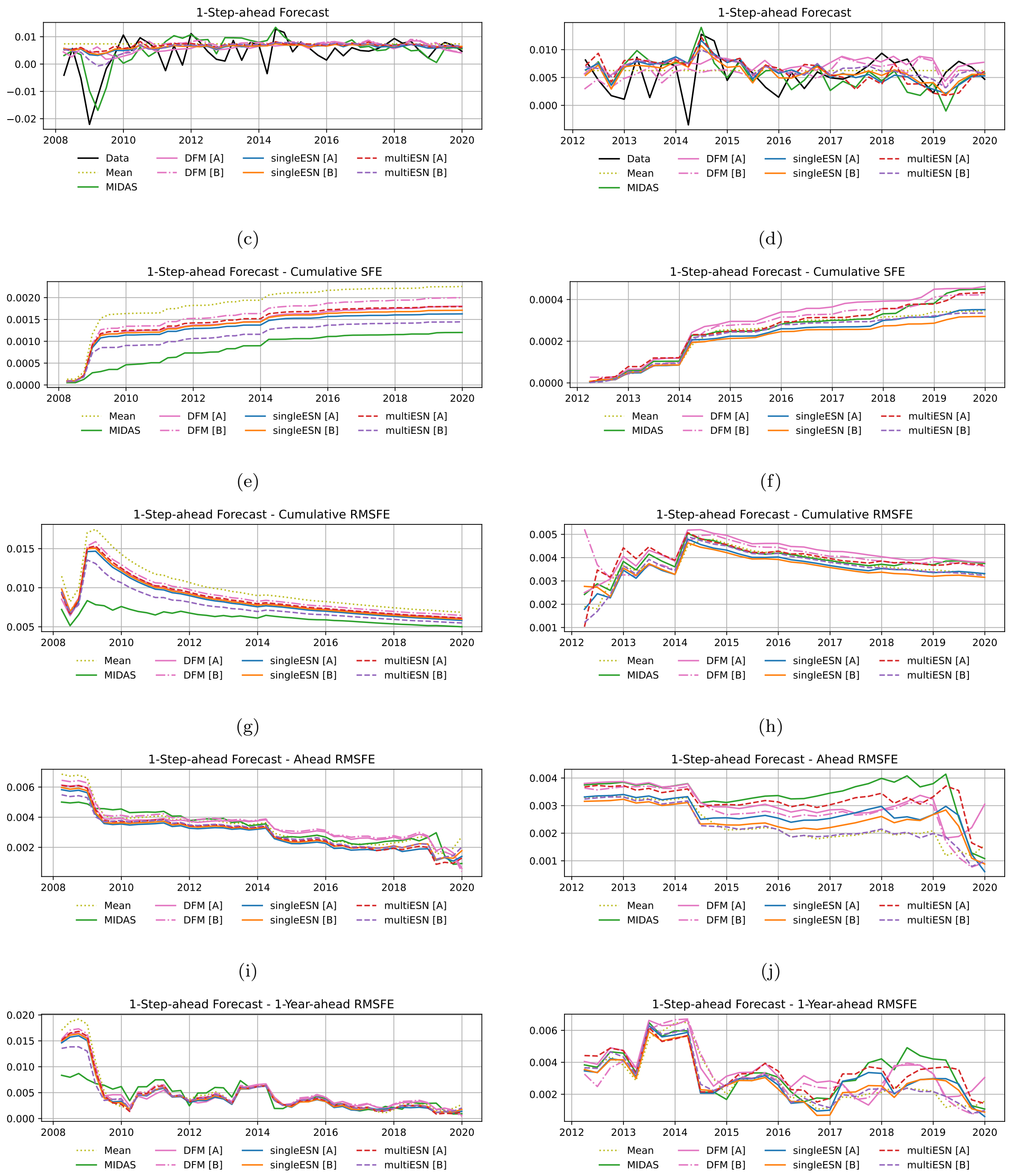
(a) Fixed Parameters



(b) Expanding Window

(c) Rolling Window

Figure 23: 1-Step-ahead GDP Forecasting, Fixed Parameters - Small-MD Dataset



(a) Pre-crisis model

(b) Post-crisis model

(c)

(d)

(e)

(f)

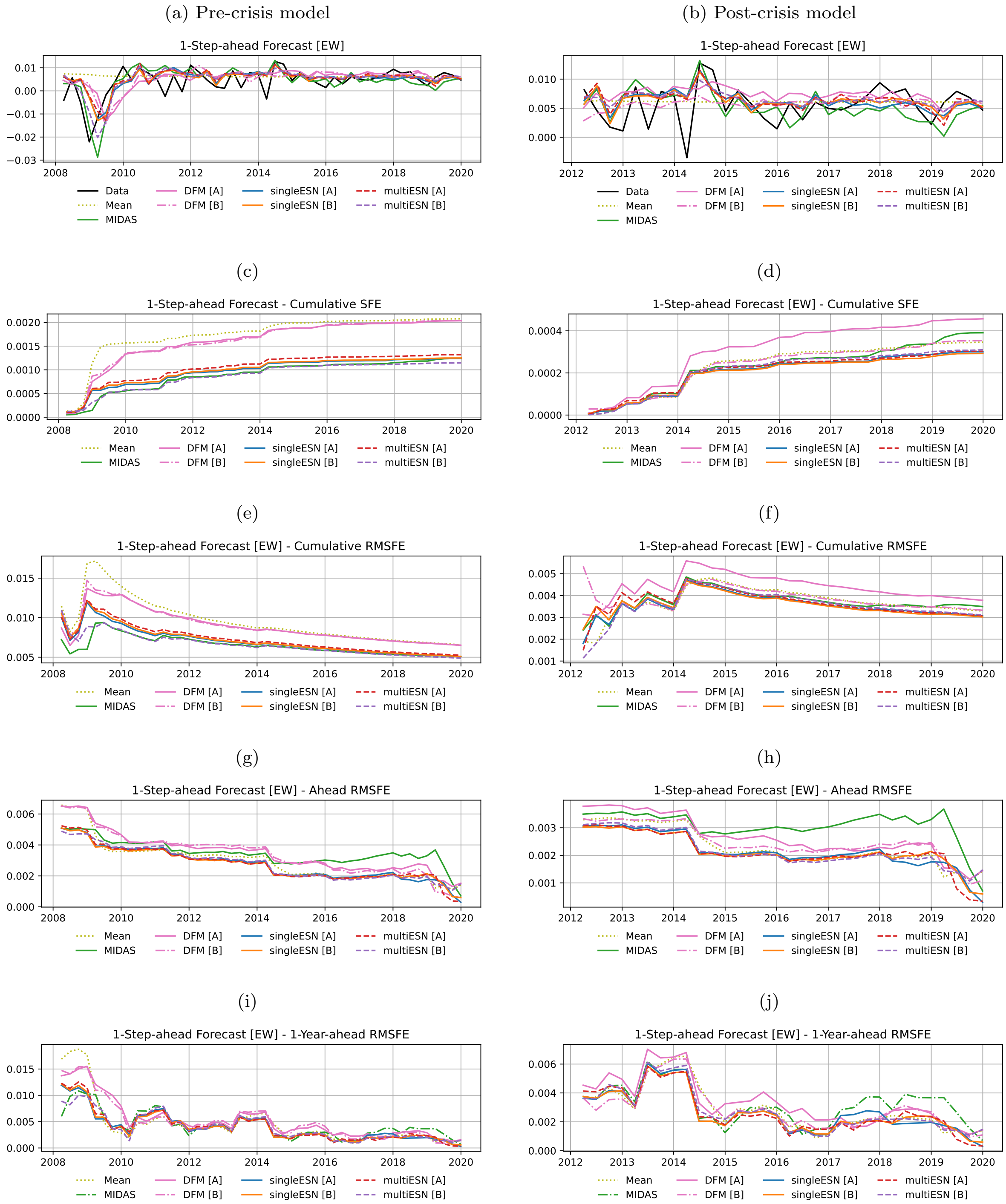
(g)

(h)

(i)

(j)

Figure 24: 1-Step-ahead GDP Forecasting, Expanding Window - Small-MD Dataset



(a) Pre-crisis model

(b) Post-crisis model

(c)

(d)

(e)

(f)

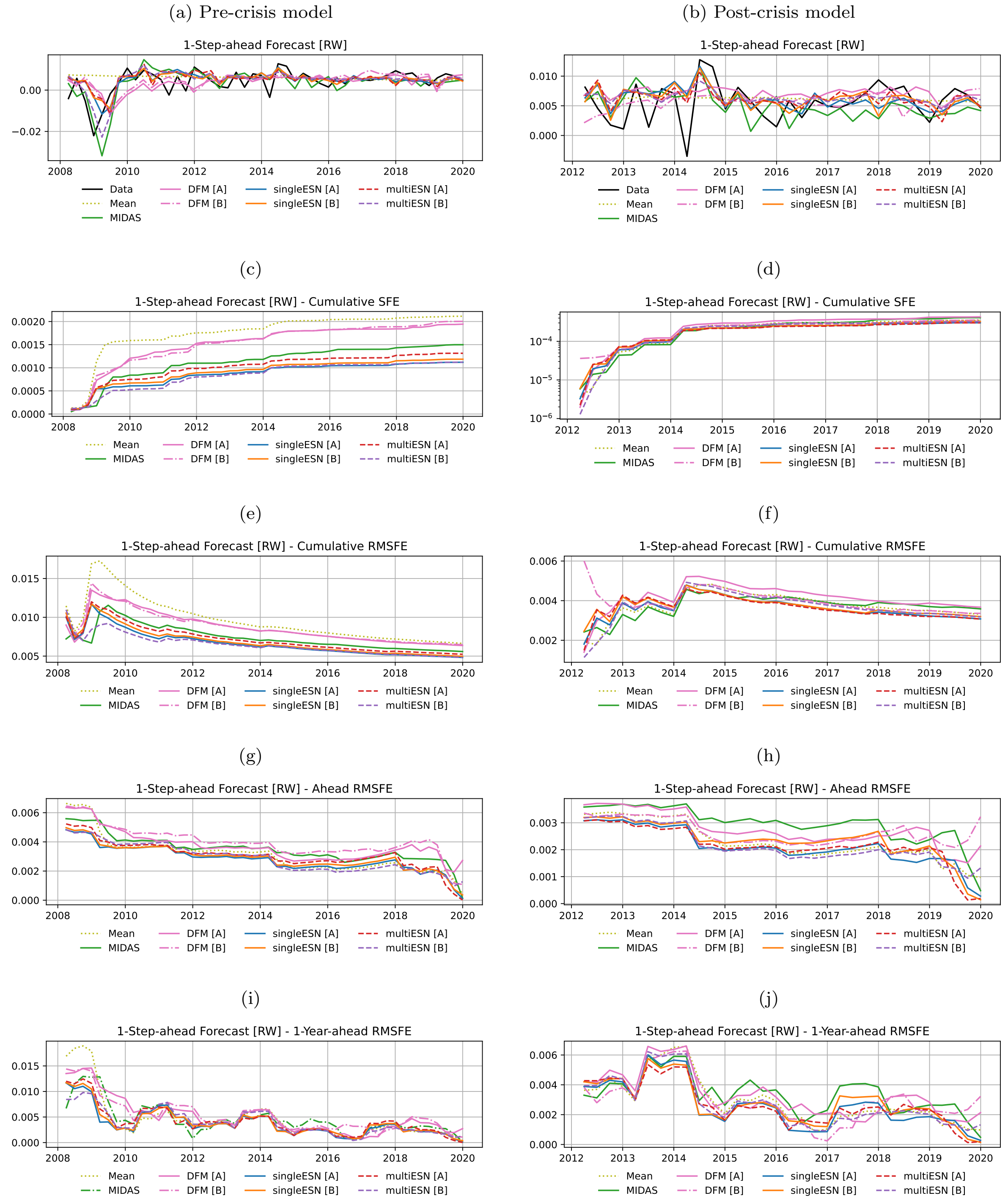
(g)

(h)

(i)

(j)

Figure 25: 1-Step-ahead GDP Forecasting, Rolling Window - Small-MD Dataset



(a) Pre-crisis model

(b) Post-crisis model

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)