A Reservoir Computing Approach to Macroeconomic Modeling Using Data with Mixed Sampling Frequencies

Vivek Bhargava

Flow of the presentation

- Introduction
- Theoretical Foundation
- Modeling Approach
- Auxiliary Terminology
- Empirical Findings
- Conclusion

Introduction

- Importance of economic forecasting
- Types of economic forecasting
  - High frequency financial time-series forecasting
  - Low frequency macroeconomic predictions
- Major objectives of using mixed frequency data
  - Forecasting
  - Nowcasting
- Two key econometric methodology
  - Mixed Data Sampling (MIDAS) regressions
  - Reservoir Computing
- Key objective – to forecast low frequency variable using mixed frequency data

Introduction

- MIDAS regressions
  - Koenig et al. (2003) used distributed lag model to predict quarterly GDP using monthly IP and unemployment rate
  - Zadrozny (1990) used VARMA models on quarterly and monthly frequency data
  - Ghysels et al. (2006), Andreou et al. (2010) developed different weighting scheme that formed the basis for MIDAS regressions
  - Foroni et al. (2015) removed restriction on the weighting scheme and proposed Unrestricted-MIDAS methodology

Introduction

- Reservoir Computing (RC)
  - Artificial Neural Network (ANNs) vs Recurrent Neural Network (RNNs)
  - LeCun et al. (2012) introduced the backpropagation methodology - involves optimizing the weights by using the gradient of the loss function with respect to the weights.
  - RNNs face the problem of exponential vanishing of error gradients (Bengio et al., 1994)
  - Echo State Network (ESNs) were developed (Jaeger, 2001; Maass et al., 2002)

Theoretical Foundation

Reservoir Computing

- The key objective of the RC approach is to train an input signal $z_t \in \mathbb{R}^{N_Z}$ to obtain a desired output $y_t \in \mathbb{R}^{N_Y}$ and $x_t \in \mathbb{R}^{N_X}$ are the reservoir states, where $t \in \mathbb{Z}$ denotes the discrete time.

- The state-space system of a reservoir is defined by:

$$x_t = \mathbf{F}(x_{t-1}, z_t)$$

$$y_t = \mathbf{h}(x_t)$$

- $\mathbf{F} : \mathbb{R}^{N_x} \times \mathbb{R}^{N_z} \rightarrow \mathbb{R}^{N_x}$ as a mapping from the low dimensional input to a higher dimensional reservoir state x_t

- $\mathbf{h} : \mathbb{R}^{N_x} \rightarrow \mathbb{R}^{N_y}$ transforms a high dimensional reservoir state into the desired N_y dimensional output space

Theoretical Foundation

Reservoir Computing

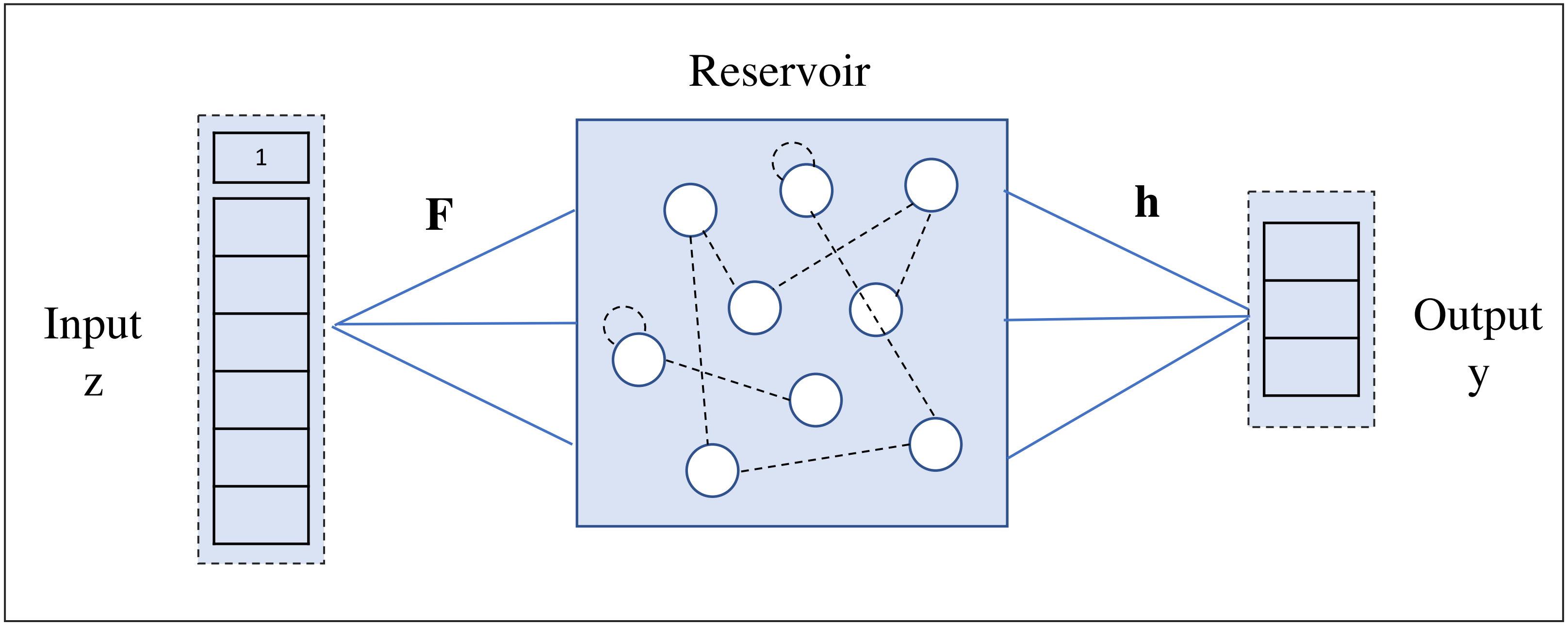


Figure 1: A schematic of a network for the reservoir computing approach (based on Gonon et al. (2020), Lukosevicius & Jaeger (2009))

Theoretical Foundation

Reservoir Computing – Echo State Network

- The RC system is extended to formally define the ESN:

$$\tilde{\mathrm{x}}_{\mathrm{t}} = \boldsymbol{\sigma}\left(\mathbf{A}\,\mathrm{x}_{\mathrm{t}-1} + \mathbf{C}\,[1; \mathrm{z}_{\mathrm{t}}]\right)$$

$$\mathrm{x}_{\mathrm{t}} = (1-\alpha)\mathrm{x}_{\mathrm{t}-1} + \alpha\,\tilde{\mathrm{x}}_{\mathrm{t}}$$

$$\mathrm{y}_{\mathrm{t}} = \mathbf{W}^{\mathbf{Out}}[1; \mathrm{x}_{\mathrm{t}}]$$

- $\mathbf{A} \in \mathbb{R}^{\mathrm{N_x}} \times \mathbb{R}^{\mathrm{N_x}}$ is the reservoir connection matrix and
- $\mathbf{C} \in \mathbb{R}^{\mathrm{N_x} \times (1+\mathrm{N_z})}$ is the input mask
- $\boldsymbol{\sigma}$ is the activation function, α is the leakage rate
- $\mathbf{W}^{\mathbf{Out}}$ is the linear readout mapping

Theoretical Foundation

ESN - Reservoir Properties

- Size of the reservoir
- Distribution of the weights
- Sparsity of the reservoir for **A**
- Spectral radius for **A**
- Scaling the input mask **C**
- Leaking rate

Theoretical Foundation

ESN – Training the network

- Preparing the states

$$\tilde{\mathbf{X}} = [\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \dots, \mathrm{x}_\mathrm{T}]$$

$$\mathbf{X} = [\mathbf{1}_\mathbf{T}; \tilde{\mathbf{X}}]$$

where $\mathrm{x}_\mathrm{t} \in \mathbb{R}^{\mathrm{N}_\mathrm{x}}$, $\tilde{\mathbf{X}} \in \mathbb{R}^{\mathrm{N}_\mathrm{x} \times \mathrm{T}}$ and $\mathbf{X} \in \mathbb{R}^{(1+\mathrm{N}_\mathrm{x}) \times \mathrm{T}}$

- The readouts are trained using ridge regularization:

$$\mathbf{W}^{\mathbf{Out}} = \mathbf{Y}^{\mathbf{Target}} \mathbf{X}^{\mathbf{T}} \left(\mathbf{X}\mathbf{X}^{\mathbf{T}} + \lambda \mathbf{I}\right)^{-\mathbf{1}}$$

where $\mathbf{Y}^{\mathbf{Target}} \in \mathbb{R}^{\mathrm{N}_\mathrm{y} \times \mathrm{T}}$ and λ is the ridge coefficient

Theoretical Foundation

MIDAS Regressions

- Let there be n different frequency data series labelled as {'A', 'Q', 'M',…} and the frequency interval defined as {'a', 'q', 'm',…}

- A simple aggregate model:

$$y_t^Q = \mu + \beta\, x_{t-1}^M + \epsilon_t$$

where $x_{t-1}^M = \sum_{j=1}^m x_{j,t-1}^M$ or $x_{t-1}^M = \frac{1}{m}\sum_{j=1}^m x_{j,t-1}^M$

- A Distributed Lag (DL) and Autoregressive DL (ADL) model:

$$y_t^Q = \mu + \sum_{j=1}^m \beta_j^M x_{j,t-1}^M + \epsilon_t$$

$$y_t^Q = \mu + \beta^Q y_{t-1}^Q + \sum_{j=1}^d \beta_j^D x_{j,t-1}^D + \epsilon_t$$

Theoretical Foundation

MIDAS Regressions

- The MIDAS model:

$$y_t^Q = \mu + \beta^Q y_{t-1}^Q + \beta^D \sum_{j=1}^{d} w_j(\theta^D) x_{j,t-1}^D + \epsilon_t$$

- where $\sum_{j=1}^{d} w_j(\theta^D) = 1$ is a weighting scheme.

- The weighting scheme can be parametrized using various polynomial function.

- The parameters (μ, β^Q, β^D, θ^D) are estimated using Nonlinear Least Square (NLS) estimation method.

Theoretical Foundation

MIDAS Regressions

- Parametrization of the polynomial weights:

Exponential Almon lag: $$\mathrm{w_j}(\theta_1, \theta_2) = \frac{\exp(\theta_1 j + \theta_2 j^2)}{\sum_{j=1}^{d} \exp(\theta_1 j + \theta_2 j^2)}$$

Beta lag: $$\mathrm{w_j}(\theta_1, \theta_2) = \frac{\mathbf{f}(j;\theta_1;\theta_2)}{\sum_{j=1}^{d} \mathbf{f}(j;\theta_1;\theta_2)}$$

- where

$$\mathbf{f}(j;\theta_1;\theta_2) = \frac{j^{\theta_1-1}(1-j)^{\theta_2-1}\Gamma(\theta_1+\theta_2)}{\Gamma(\theta_1)+\Gamma(\theta_2)}$$

$$\Gamma(\theta) = \int_0^{\infty} e^{-x}\, x^{\theta-1} dx$$

Modeling Approach

Model 1: Stacked Input Model

- Key idea – modify the input signal
- Stack high frequency variable according to the lowest frequency
- Consider N^{Q} quarterly and N^{M} monthly data series with $m = 3$
- Let $\mathrm{z}_{\mathrm{t}}^{\mathrm{Q}} \in \mathbb{R}^{\mathrm{N}_{\mathrm{z}}^{\mathrm{Q}}}$ and $\mathrm{z}_{\mathrm{t}}^{\mathrm{M}} \in \mathbb{R}^{\mathrm{N}_{\mathrm{z}}^{\mathrm{M}}}$ are the input signal for corresponding frequency where $\mathrm{N}_{\mathrm{z}}^{\mathrm{Q}} = \mathrm{N}^{\mathrm{Q}}$ and $\mathrm{N}_{\mathrm{z}}^{\mathrm{M}} = m \times \mathrm{N}^{\mathrm{M}}$
- Now constructing the input signal as

$$\mathrm{z}_{\mathrm{t}} = [\mathrm{z}_{\mathrm{t}}^{\mathrm{Q}}; \mathrm{z}_{\mathrm{t}}^{\mathrm{M}}]$$

with $\mathrm{z}_{\mathrm{t}} \in \mathbb{R}^{\mathrm{N}_{\mathrm{z}}}$ where $\mathrm{N}_{\mathrm{z}} = \mathrm{N}_{\mathrm{z}}^{\mathrm{Q}} + \mathrm{N}_{\mathrm{z}}^{\mathrm{M}}$

Modeling Approach

Model 1: Stacked Input Model

- State space equation for Model 1:

$$\tilde{\mathrm{x}}_{\mathrm{t}} = \boldsymbol{\sigma}\left(\mathbf{A}\,\mathrm{x}_{\mathrm{t}-1} + \mathbf{C}\,[1;\mathrm{z}_{\mathrm{t}}]\right)$$

$$\mathrm{x}_{\mathrm{t}} = (1-\alpha)\mathrm{x}_{\mathrm{t}-1} + \alpha\tilde{\mathrm{x}}_{\mathrm{t}}$$

- Preparing the states:

$$\widetilde{\mathbf{X}} = [\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \ldots, \mathrm{x}_{\mathrm{T}}]$$

$$\mathbf{X} = [\mathbf{1}_{\mathbf{T}}, \widetilde{\mathbf{X}}]$$

- The readouts are trained using ridge regularization:

$$\mathbf{W}^{\mathbf{Out}} = \mathbf{Y}^{\mathbf{Target}}\,\mathbf{X}^{\mathbf{T}}\left(\mathbf{X}\mathbf{X}^{\mathbf{T}} + \lambda\mathbf{I}\right)^{-\mathbf{1}}$$

Modeling Approach

Model 2: Synchronizing States

- A dedicated reservoir system for individual frequency.
- For quarterly-monthly data series, two reservoir systems are used
- Let N^{Q} and N^{M} be the number of different quarterly and monthly data series, with $t = \{1,2, \dots, T\}$ and $\tau = \{1,2, \dots, mT\}$ the corresponding discrete time index
- Let $\mathrm{z}_{\mathrm{t}}^{\mathrm{Q}} \in \mathbb{R}^{\mathrm{N}^{\mathrm{Q}}}$ and $\mathrm{z}_{\mathrm{t}}^{\mathrm{M}} \in \mathbb{R}^{\mathrm{N}^{\mathrm{M}}}$ are the input signal for corresponding frequency
- Additionally, $\mathrm{N}_{\mathrm{x}}^{\mathrm{Q}}$ and $\mathrm{N}_{\mathrm{x}}^{\mathrm{M}}$ are the number of neurons in the respective reservoir

Modeling Approach

Model 2: Synchronizing States

- States update dynamics for individual frequencies:

Quarterly Frequency:
$$\tilde{x}^{Q}{}_{t} = \boldsymbol{\sigma}\left(\mathbf{A}^{\mathbf{Q}}\, x^{Q}_{t-1} + \mathbf{C}^{\mathbf{Q}}[1; z^{Q}_{t}]\right)$$
$$x^{Q}_{t} = \left(1 - \alpha^{Q}\right)x^{Q}_{t-1} + \alpha^{Q}\, \tilde{x}^{Q}{}_{t}$$

Monthly Frequency:
$$\tilde{x}^{M}{}_{t} = \boldsymbol{\sigma}\left(\mathbf{A}^{\mathbf{M}}\, x^{M}_{t-1} + \mathbf{C}^{\mathbf{M}}[1; z^{M}_{t}]\right)$$
$$x^{M}_{t} = \left(1 - \alpha^{M}\right)x^{M}_{t-1} + \alpha^{M}\, \tilde{x}^{M}{}_{t}$$

Modeling Approach

Model 2A: Synchronizing States at Low Frequency

- Updated states from individual reservoir system:

$$\widetilde{\mathbf{X}}^{\mathbf{Q}} = \left[\mathrm{x}_1^{\mathrm{Q}}, \mathrm{x}_2^{\mathrm{Q}}, \mathrm{x}_3^{\mathrm{Q}}, \dots, \mathrm{x}_{\mathrm{T}}^{\mathrm{Q}}\right]$$

$$\widetilde{\mathbf{X}}^{\widetilde{\mathbf{M}}} = [\mathrm{x}_1^{\mathrm{M}}, \mathrm{x}_2^{\mathrm{M}}, \mathrm{x}_3^{\mathrm{M}}, \dots, \mathrm{x}_{\tau}^{\mathrm{M}}]$$

where $\widetilde{\mathbf{X}}^{\mathbf{Q}} \in \mathbb{R}^{\mathrm{N}_{\mathrm{x}}^{\mathrm{Q}} \times \mathrm{T}}$ and $\widetilde{\mathbf{X}}^{\widetilde{\mathbf{M}}} \in \mathbb{R}^{\mathrm{N}_{\mathrm{x}}^{\mathrm{M}} \times m\mathrm{T}}$

- Every m^{th} state vector from $\widetilde{\mathbf{X}}^{\widetilde{\mathbf{M}}}$ is collected in $\widetilde{\mathbf{X}}^{\mathbf{M}} \in \mathbb{R}^{\mathrm{N}_{\mathrm{x}}^{\mathrm{M}} \times \mathrm{T}}$

- Preparing the states matrix as $\mathbf{X} \in \mathbb{R}^{(1+\mathrm{N}_{\mathrm{x}}^{\mathrm{M}}+\mathrm{N}_{\mathrm{x}}^{\mathrm{M}}) \times \mathrm{T}}$:

$$\mathbf{X} = [\mathbf{1}_{\mathbf{T}}, \widetilde{\mathbf{X}}^{\mathbf{Q}}, \widetilde{\mathbf{X}}^{\mathbf{M}}]$$

Modeling Approach

Model 2A: Synchronizing States at Low Frequency

- Setting up the target matrix:

$$\mathbf{Y}^{\mathbf{Target}} = [\mathrm{y}_2, \mathrm{y}_3, \mathrm{y}_4, \dots, \mathrm{y}_{\mathrm{T}+1}]$$

where $\mathbf{Y}^{\mathbf{Target}} \in \mathbb{R}^{\mathrm{N_y} \times \mathrm{T}}$ and $\mathrm{y_t} \in \mathbb{R}^{\mathrm{N_y}}$

- The readouts are trained using ridge regularization:

$$\mathbf{W}^{\mathbf{Out}} = \mathbf{Y}^{\mathbf{Target}}\, \mathbf{X}^{\mathbf{T}}\left(\mathbf{X}\mathbf{X}^{\mathbf{T}} + \lambda \mathbf{I}\right)^{-\mathbf{1}}$$

Modeling Approach

Model 2B: Synchronizing States at High Frequency

- Updated states from individual reservoir system:

$$\widetilde{\mathbf{X}}^{\mathbf{Q}} = \left[\mathrm{x}_1^{\mathrm{Q}}, \mathrm{x}_2^{\mathrm{Q}}, \mathrm{x}_3^{\mathrm{Q}}, \dots, \mathrm{x}_{\mathrm{T}}^{\mathrm{Q}}\right]$$

$$\widetilde{\mathbf{X}}^{\widetilde{\mathbf{M}}} = [\mathrm{x}_1^{\mathrm{M}}, \mathrm{x}_2^{\mathrm{M}}, \mathrm{x}_3^{\mathrm{M}}, \dots, \mathrm{x}_{\tau}^{\mathrm{M}}]$$

- Every state vector from $\widetilde{\mathbf{X}}^{\mathbf{Q}}$ is repeated m times to obtain the expanded quarterly state matrix $\widetilde{\mathbf{X}}^{\widetilde{\mathbf{Q}}} \in \mathbb{R}^{\mathrm{N}_{\mathrm{x}}^{\mathrm{Q}} \times m\mathrm{T}}$

$$\widetilde{\mathbf{X}}^{\mathbf{Q}} = \left[[\mathrm{x}_1^{\mathrm{Q}}, \mathrm{x}_1^{\mathrm{Q}}, \mathrm{x}_1^{\mathrm{Q}}], [\mathrm{x}_2^{\mathrm{Q}}, \mathrm{x}_2^{\mathrm{Q}}, \mathrm{x}_2^{\mathrm{Q}}], \dots, [\mathrm{x}_{\mathrm{T}}^{\mathrm{Q}}, \mathrm{x}_{\mathrm{T}}^{\mathrm{Q}}, \mathrm{x}_{\mathrm{T}}^{\mathrm{Q}}]\right]$$

- Preparing the states matrix as $\mathbf{X} \in \mathbb{R}^{(1+\mathrm{N}_{\mathrm{x}}^{\mathrm{M}}+\mathrm{N}_{\mathrm{x}}^{\mathrm{M}})\times\mathrm{T}}$:

$$\mathbf{X} = [\mathbf{1}_{\mathbf{mT}}, \widetilde{\mathbf{X}}^{\widetilde{\mathbf{Q}}}, \widetilde{\mathbf{X}}^{\widetilde{\mathbf{M}}}]$$

Modeling Approach

Model 2B: Synchronizing States at High Frequency

- Setting up the target matrix:

$$\mathbf{Y}^{\mathbf{Target}} = \left[[\mathrm{y}_2, \mathrm{y}_2, \mathrm{y}_2], [\mathrm{y}_3, \mathrm{y}_3, \mathrm{y}_3], \ldots, [\mathrm{y}_{\mathrm{T}+1}, \mathrm{y}_{\mathrm{T}+1}, \mathrm{y}_{\mathrm{T}+1}]\right]$$

where $\mathbf{Y}^{\mathbf{Target}} \in \mathbb{R}^{\mathrm{N_y} \times \mathrm{mT}}$ and $\mathrm{y_t} \in \mathbb{R}^{\mathrm{N_y}}$

- The readouts are trained using ridge regularization:

$$\mathbf{W}^{\mathbf{Out}} = \mathbf{Y}^{\mathbf{Target}} \mathbf{X}^{\mathrm{T}} \left(\mathbf{X}\mathbf{X}^{\mathrm{T}} + \lambda \mathbf{I}\right)^{\mathbf{-1}}$$

- The low frequency predicted values are obtained by taking the mean:

$$\hat{\mathrm{y}}_{\mathrm{t}}^{\mathrm{Q}} = \frac{1}{\mathrm{m}} \sum_{j=1}^{m} \hat{\mathrm{y}}_{j,\mathrm{t}}$$

Modeling Approach

Model 3: Unstacked Input Model

- Modify the input signal according to the higher frequency
- The two input matrices for quarterly and monthly data series are defined as:

$$\mathbf{Z^{Q}} = [\mathrm{z}_1^{\mathrm{Q}}, \mathrm{z}_2^{\mathrm{Q}}, \mathrm{z}_3^{\mathrm{Q}}, \dots, \mathrm{z}_{\mathrm{T}}^{\mathrm{Q}}]$$

$$\mathbf{Z^{\widetilde{M}}} = [\mathrm{z}_1^{\mathrm{M}}, \mathrm{z}_2^{\mathrm{M}}, \mathrm{z}_3^{\mathrm{M}}, \dots, \mathrm{x}_{\tau}^{\mathrm{M}}]$$

where $\mathbf{Z^{Q}} \in \mathbb{R}^{\mathrm{N^{Q}} \times \mathrm{T}}$ and $\mathbf{Z^{\widetilde{M}}} \in \mathbb{R}^{\mathrm{N^{M}} \times m\mathrm{T}}$

- The input signals in the quarterly data are repeated to synchronize at a data level according to the high frequency monthly data, such that:

$$\mathbf{Z^{\widetilde{Q}}} = \left[[\mathrm{z}_1^{\mathrm{Q}}, \mathrm{z}_1^{\mathrm{Q}}, \mathrm{z}_1^{\mathrm{Q}}], [\mathrm{z}_2^{\mathrm{Q}}, \mathrm{z}_2^{\mathrm{Q}}, \mathrm{z}_2^{\mathrm{Q}}], \dots, [\mathrm{z}_{\mathrm{T}}^{\mathrm{Q}}, \mathrm{z}_{\mathrm{T}}^{\mathrm{Q}}, \mathrm{z}_{\mathrm{T}}^{\mathrm{Q}}]\right]$$

Modeling Approach

Model 3: Unstacked Input Model

- A final input matrix is constructed as $\mathbf{Z} \in \mathbb{R}^{(N^Q+N^M)\times mT}$

$$\mathbf{Z} = [\mathbf{Z}^{\mathbf{Q}}; \mathbf{Z}^{\mathbf{M}}]$$

- The states are updates as:

$$\tilde{x}_t = \boldsymbol{\sigma}\left(\mathbf{A}\, x_{t-1} + \mathbf{C}\,[1; z_t]\right)$$

$$x_t = (1-\alpha)x_{t-1} + \alpha\tilde{x}_t$$

with $z_t \in \mathbb{R}^{N_z}$ where $N_z = N^Q + N^M$

Modeling Approach

Model 3: Unstacked Input Model

- Preparing the states:

$$\widetilde{\mathbf{X}} = [\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \dots, \mathrm{x}_{m\mathrm{T}}]$$

$$\mathbf{X} = [\mathbf{1}_{m\mathbf{T}}, \widetilde{\mathbf{X}}]$$

- Setting up the target matrix:

$$\mathbf{Y}^{\mathbf{Target}} = \big[[\mathrm{y}_2, \mathrm{y}_2, \mathrm{y}_2], [\mathrm{y}_3, \mathrm{y}_3, \mathrm{y}_3], \dots, [\mathrm{y}_{\mathrm{T}+1}, \mathrm{y}_{\mathrm{T}+1}, \mathrm{y}_{\mathrm{T}+1}]\big]$$

where $\mathbf{Y}^{\mathbf{Target}} \in \mathbb{R}^{\mathrm{N_y} \times \mathrm{mT}}$ and $\mathrm{y_t} \in \mathbb{R}^{\mathrm{N_y}}$

- The readouts are trained using ridge regularization method

- The low frequency predicted values are obtained by taking the mean:

$$\hat{\mathrm{y}}_{\mathrm{t}}^{\mathrm{Q}} = \frac{1}{\mathrm{m}} \sum_{j=1}^{m} \hat{\mathrm{y}}_{j,\mathrm{t}}$$

Auxiliary Terminology

Hyperparameter Optimization

- Key hyperparameters to be tuned
  - Size of the reservoir: N_x
  - Leakage rate: α
  - Scaling input mask: bias $\mathbf{b}$ and γ
  - Spectral radius for the reservoir connections: ρ
  - Regularization coefficient: λ
- Nested optimization
  - Grid approach for the discrete values
  - Continuous optimization for the non-discrete reservoir parameters

Auxiliary Terminology

Hyperparameter Optimization – Fixed vs Expanding Window

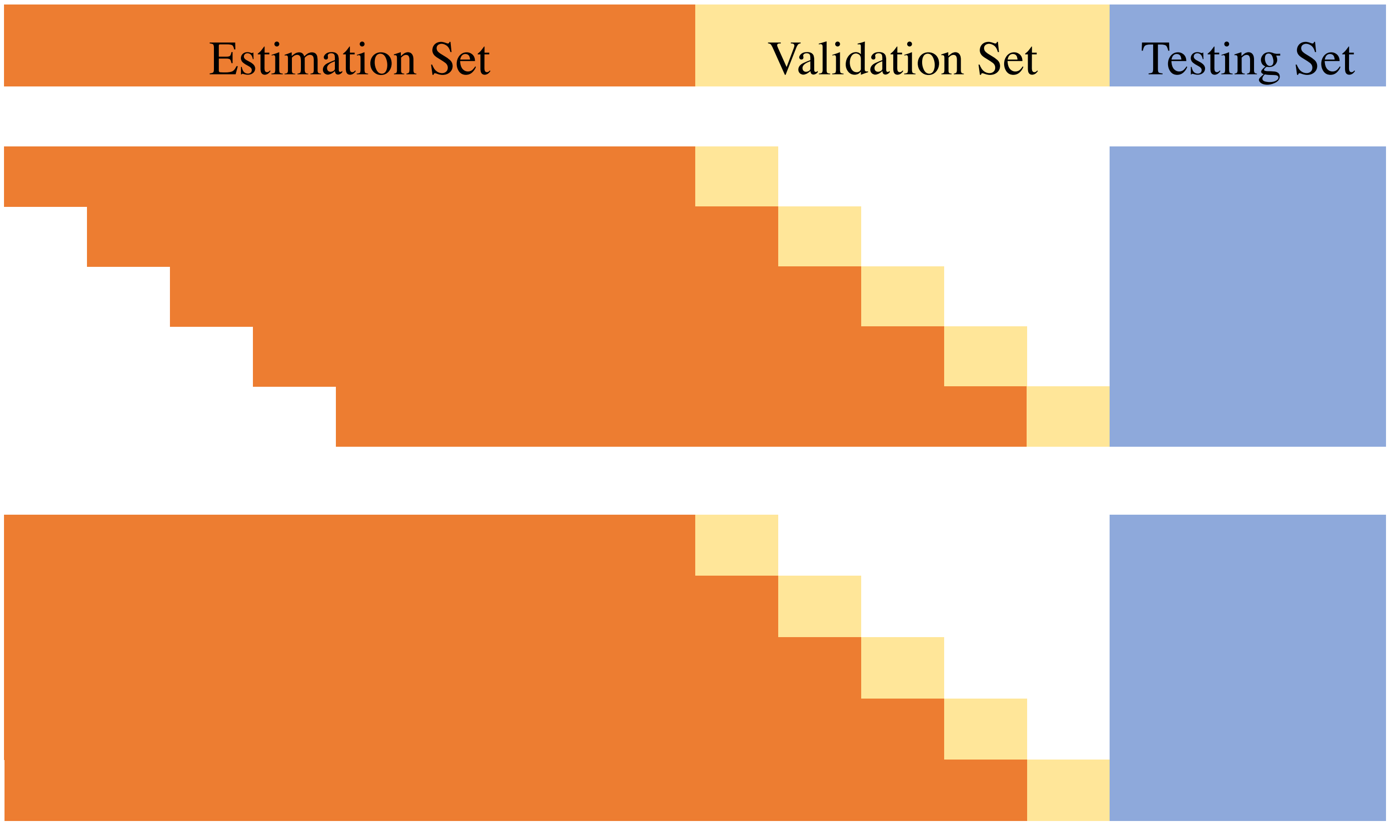


Figure 2: The above schematic shows the distribution of total data into training and testing dataset. The training dataset is further divided into estimation and validation set to optimize the reservoir dynamics using k-fold cross validation. The split for the three groups is taken as 50%, 30% and 20% respectively.

Auxiliary Terminology

Training Methodology

- The network is re-trained using optimized hyperparameters
- Training approach
  - Fixed window training
  - Rolling window training
- Iterative forecasting approach used to make h – step ahead forecasts

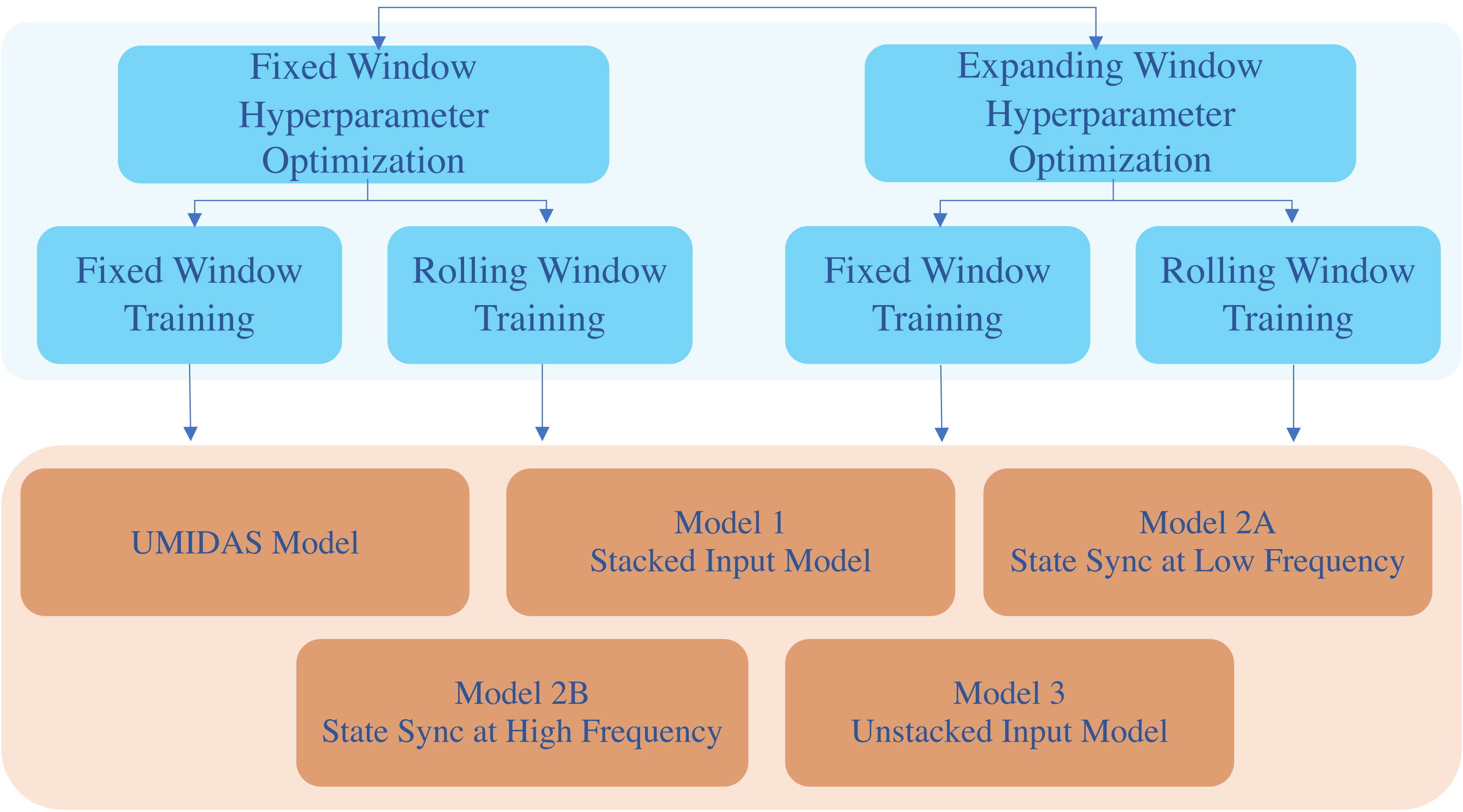
Auxiliary Terminology

Data Description

- Type of mixed frequencies used – quarterly and monthly time series
- Quarterly data – Real GDP growth rates
- Monthly data – Core Price Index (CPI) growth and unemployment rate
- Time period – Q1 1985 to Q1 2011
- $\mathrm{T}_{\mathrm{Total}} = 105$ (quarterly periods)
- Timing convention

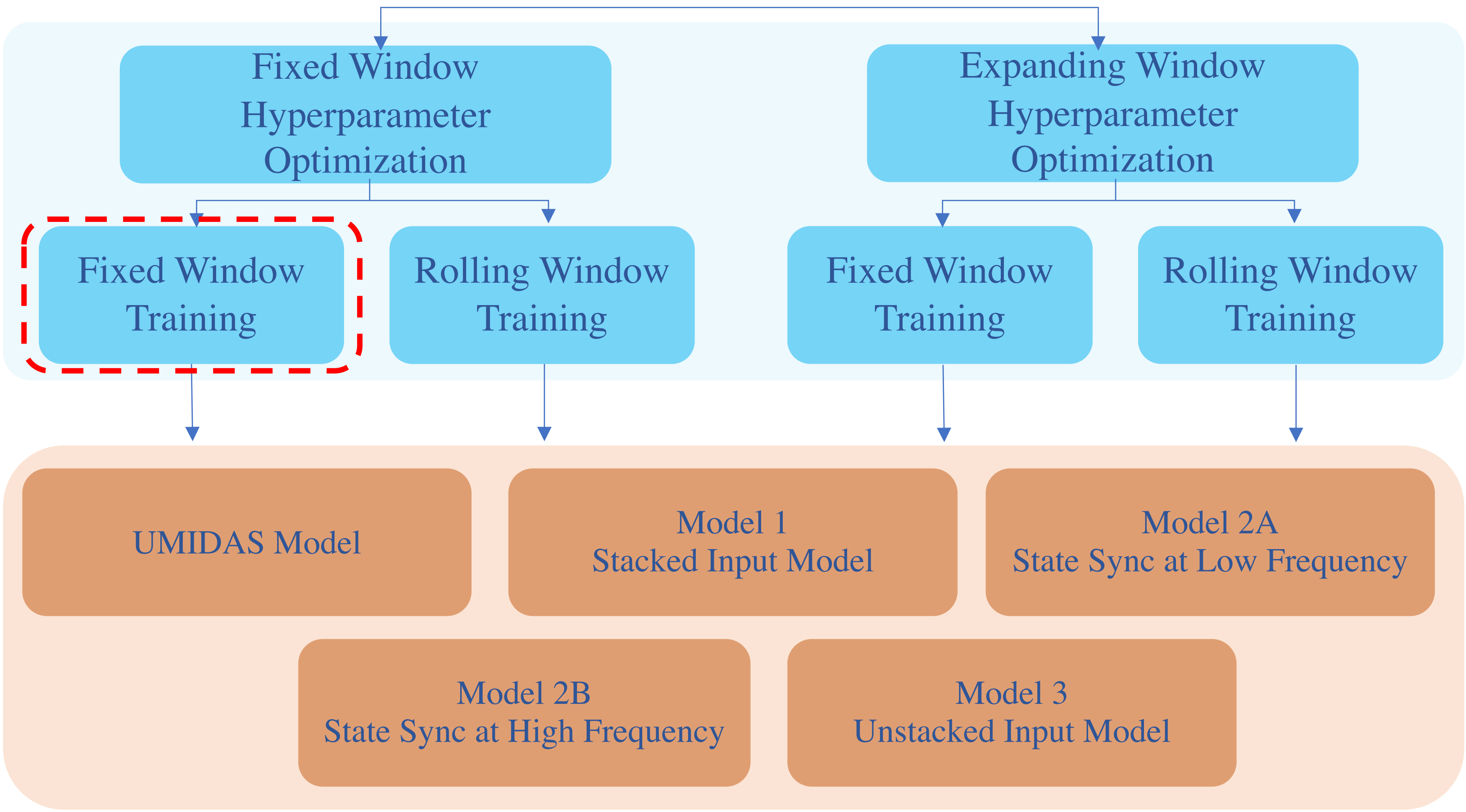
Empirical Findings

Structure of results



Empirical Findings

Structure of results



Empirical Findings

UMIDAS Model

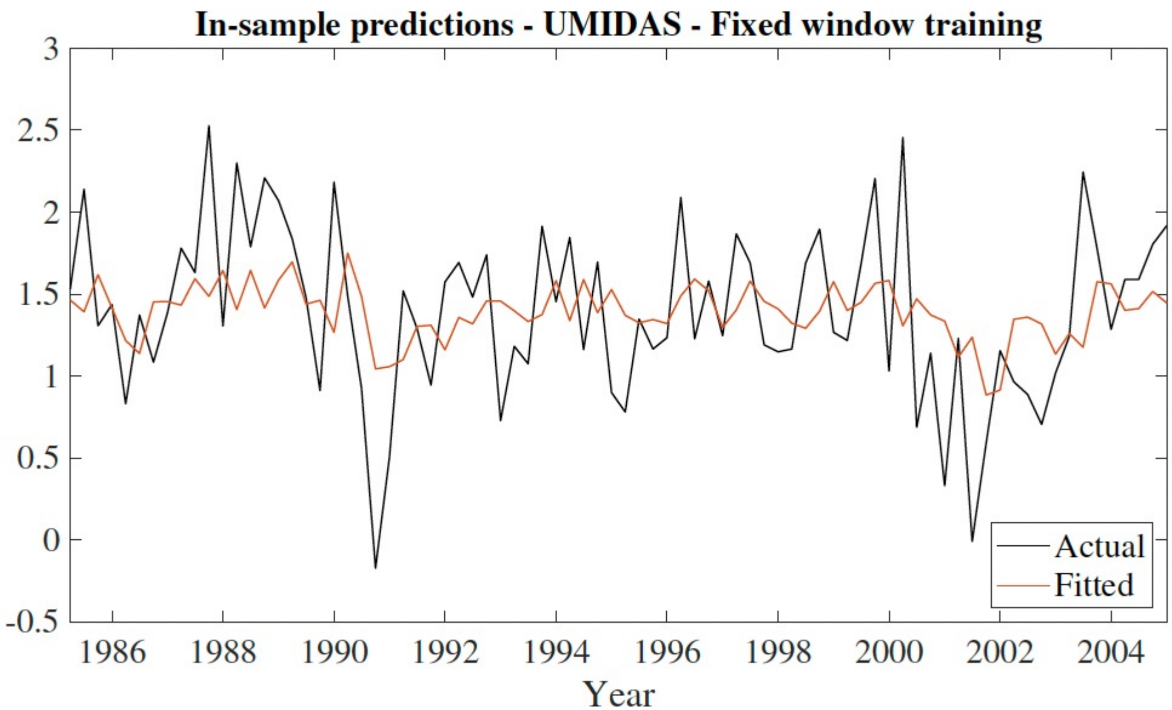


Figure 3: In-sample predictions of UMIDAS model showing the 1-step ahead fitted value for the targeted quarterly GDP growth rate. The values are obtained using fixed window training.

Empirical Findings

UMIDAS Model

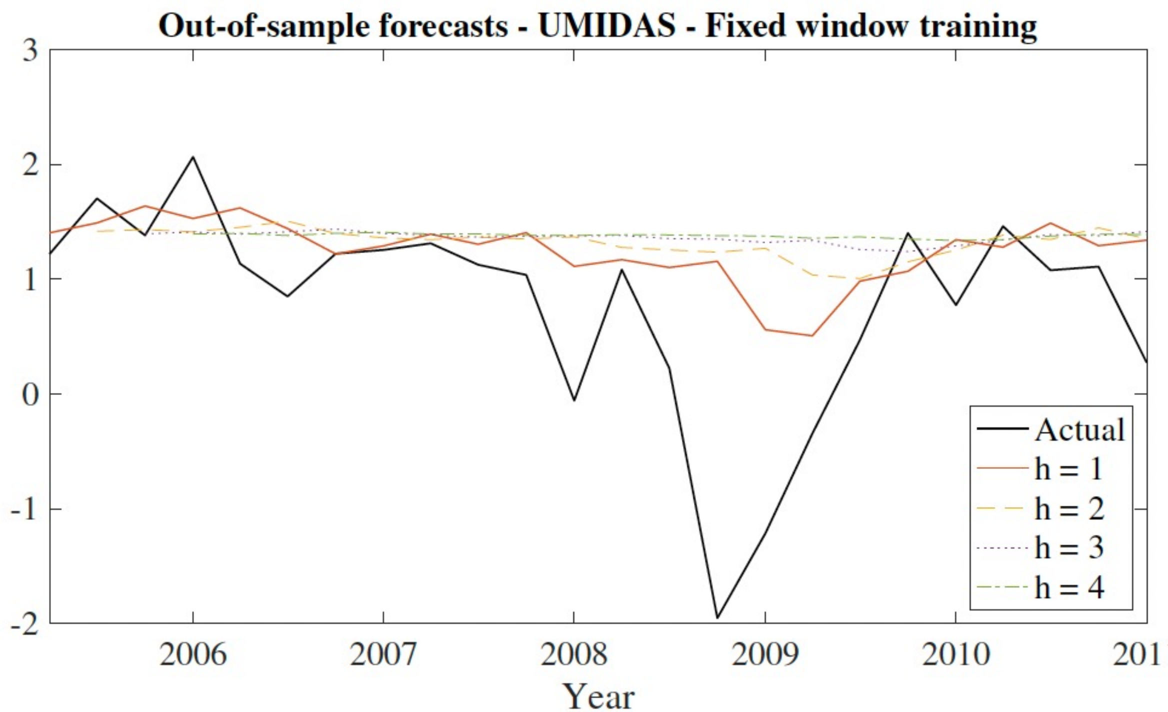


Figure 4: Out-of-sample forecasts of UMIDAS model showing the forecasted value for the targeted quarterly GDP growth rate. These are h-step ahead forecasts using fixed window training.

Empirical Findings

UMIDAS Model

- Out of sample performance of the UMIDAS model:

	MSE: h=1	MSE: h=2	MSE: h=3	MSE: h=4
UMIDAS	0.79	1.07	1.23	1.32

Table 1: Out-of-sample MSE performance of UMIDAS for $h \in \{1, 2, 3, 4\}$ step ahead forecasts. The MSE values are calculated for the quarterly GDP growth rates.

- Closely follow the dynamics with an autoregressive form
- Magnitude of movement is smaller
- Predictions are flat and large movements are not easily adjusted

Empirical Findings

Model 1 – Stacked Input Model

Neurons	**Validation Error**	γ	ρ	α	λ	b
15	0,23	3.46	1.43	0.14	0.06	8.49
20	0.28	0.07	0.62	0.49	0.08	11.72
25	0.23	-1.16	2.44	0.33	24.92	2.04
30	0.24	1.64	3.56	0.18	0.03	16.95
50	0.31	0.65	0.86	0.24	0.00	0.84
100	0.29	0.17	0.44	0.30	0.21	12.40

Table 2: Reservoir Dynamics for Model 1 using fixed window hyperparameter optimization. Based on the size of the reservoir, the table contains the optimized value of hyperparameters that are tuned using fixed window optimization technique. A *5-fold* cross validation approach is used to select the size of the reservoir based on the lowest validation error.

Empirical Findings

Model 1 – Stacked Input Model

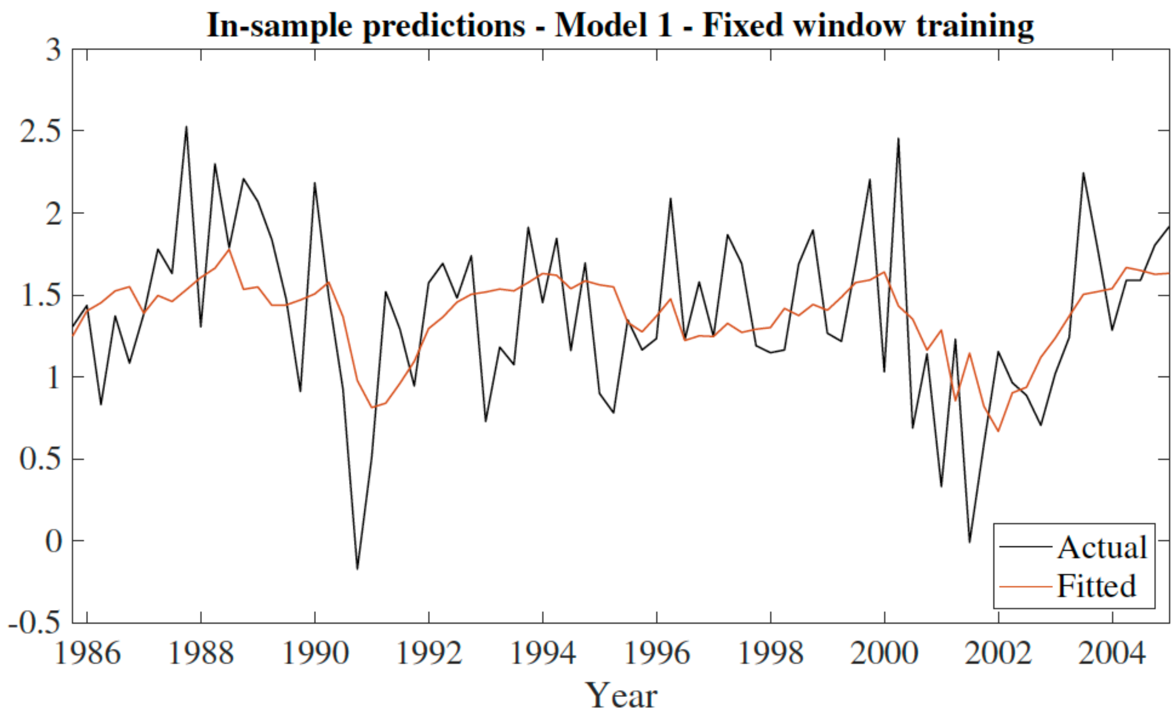


Figure 5: In-sample predictions of Model 1 using stacked input showing the fitted value for the targeted quarterly GDP growth rate. The network is trained only once on the training data. For Model 1, the monthly data are stacked according to quarterly frequency. The reservoir has $N_x = 15$ and $\alpha = 0.14$.

Empirical Findings

Model 1 – Stacked Input Model

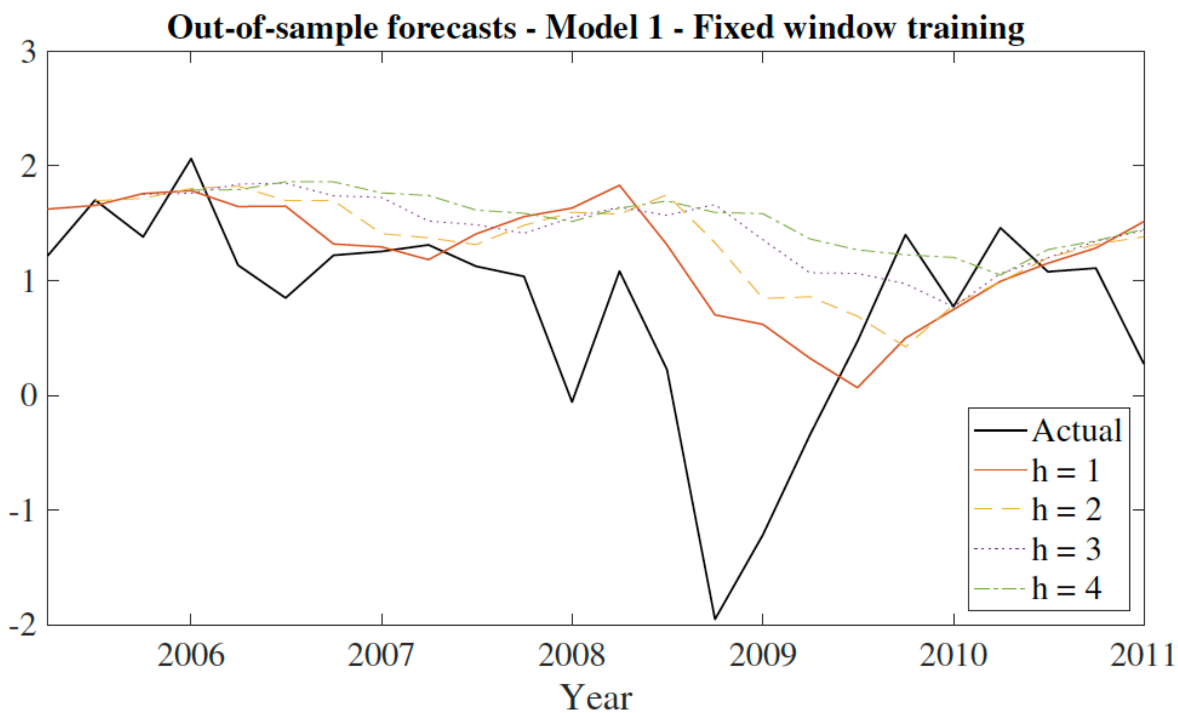


Figure 6: Out-of-sample forecasts of Model 1 model showing the forecasted values for the targeted quarterly GDP growth rate. These are h-step ahead forecasts using fixed window training, hence the network is trained only once on the training data. For Model 1, the monthly data are stacked according to quarterly frequency. The reservoir has $N_x = 15$ and $\alpha = 0.14$

Empirical Findings

Model 1 – Stacked Input Model

- Out of sample performance of the Model 1:

	MSE h=1	MSE h=2	MSE h=3	MSE h=4
Model 1	0.82	1.13	1.41	1.60
UMIDAS	0.79	1.07	1.23	1.32

Table 3: Out-of-sample MSE performance of Model 1 benchmarked against UMIDAS model for $h \in \{1, 2, 3, 4\}$ step ahead forecasts using fixed window training.

- Fitted values are smoother compared to the UMIDAS model
- The co-movement for crests and troughs are more pronounced
- Similarly, the crisis dip is captured well in the out of sample forecast
- The h-step ahead forecasts are slightly more pronounced

Empirical Findings

Model 2A: Low Frequency State Sync

Neurons	Validation Error	λ	ρ^Q	α^Q	b^Q	ρ^M	α^M	b^M
10	0.17	0.01	2.11	0.29	0.28	1.92	0.07	6.70
15	0.22	0.10	0.25	0.54	4.22	1.78	0.12	1.86
20	0.17	0.08	-2.64	0.26	0.45	3.49	0.19	1.90
25	0.18	0.02	0.84	0.39	0.08	1.68	0.42	0.75
30	0.23	0.19	0.00	0.53	-0.08	1.44	0.12	0.91
50	0.20	0.08	0.74	0.40	1102.36	1.21	0.30	2.51
100	0.21	0.00	0.00	0.13	0.00	0.52	0.23	2.95

Table 4: Reservoir Dynamics for Model 2A using fixed window hyperparameter optimization. Based on the size of the reservoir, the table contains the optimized value of hyperparameters that are tuned using fixed window optimization technique. A *5-fold* cross validation approach is used to select the size of the reservoir based on the lowest validation error.

Empirical Findings

Model 2A: Low Frequency State Sync

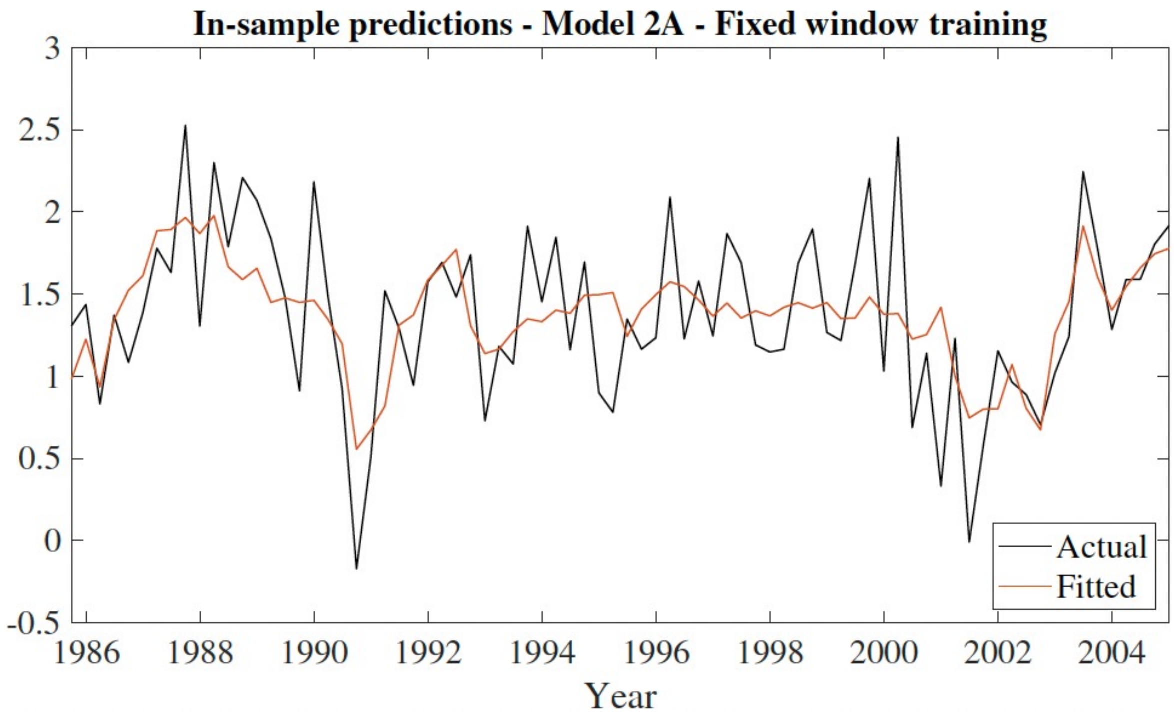


Figure 7: In-sample predictions of Model 2A showing the fitted value for the targeted quarterly GDP growth rate. For Model 2A, the states are synced according to the quarterly frequency. The optimized hyperparameters are $\mathrm{N_x} = 20$, $\alpha^Q = 0.25$ and $\alpha^M = 0.18$.

Empirical Findings

Model 2A: Low Frequency State Sync

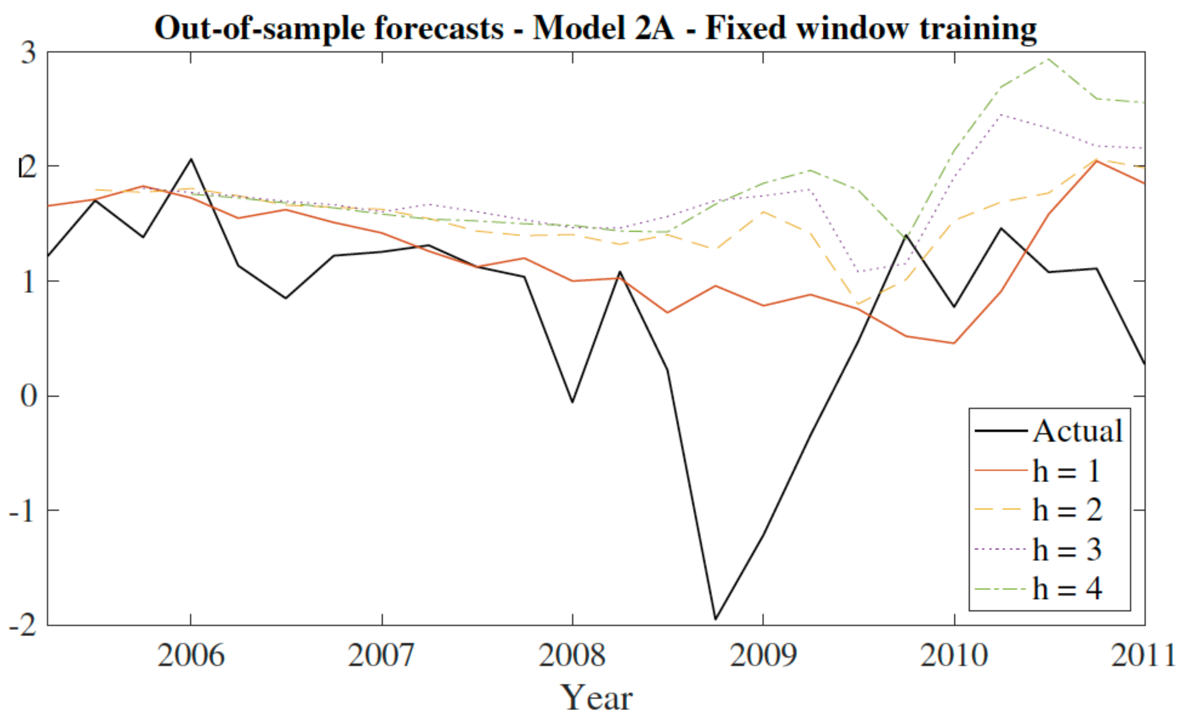


Figure 8: Out-of-sample forecasts of Model 2A model showing the forecasted values for the targeted quarterly GDP growth rate. These are h-step ahead forecasts using fixed window training, hence the network is trained only once on the training data. For Model 2A, the states are synced according to the quarterly frequency. The optimized hyperparameters are $N_x = 20$, $\alpha^Q = 0.25$ and $\alpha^M = 0.18$.

Empirical Findings

Model 2A: Low Frequency State Sync

- Out of sample performance of the Model 2A:

	MSE h=1	MSE h=2	MSE h=3	MSE h=4
Model 2A	0.90	1.39	1.91	2.36
UMIDAS	0.79	1.07	1.23	1.32

Table 5: Out-of-sample MSE performance of Model 2A benchmarked against UMIDAS model for $h \in \{1, 2, 3, 4\}$ step ahead forecasts using fixed window training.

- Fitted values are smoother and trace the actual training data closely
- Difference in leakage rate for the two reservoirs
- Magnitude of movement is smaller in the fitted data
- Out-of-sample performance deteriorates as forecasting horizon increase
- Not able to adjust to the big financial drop after Q4 2008

Empirical Findings

Model 2B: Low Frequency State Sync

Neurons	Validation Error	λ	ρ^Q	α^Q	b^Q	ρ^M	α^M	b^M
10	0.19	0.00	0.15	0.39	-0.41	2.50	0.12	4.54
15	0.12	0.00	-1.06	0.48	-0.12	3.62	0.17	3.58
20	0.16	0.01	-1.24	0.34	1.60	4.66	0.19	2.55
25	0.21	0.18	1.29	0.17	28.32	1.19	0.33	0.37
30	0.23	0.16	-0.03	0.09	0.06	1.96	0.29	6.08
50	0.19	0.01	0.01	0.06	0.89	1.52	0.22	2.23
100	0.24	0.11	-0.81	0.91	-0.18	0.11	0.20	2.58

Table 6: Reservoir Dynamics for Model 2B using fixed window hyperparameter optimization. Based on the size of the reservoir, the table contains the optimized value of hyperparameters that are tuned using fixed window optimization technique. A *5-fold* cross validation approach is used to select the size of the reservoir based on the lowest validation error.

Empirical Findings

Model 2B: High Frequency State Sync

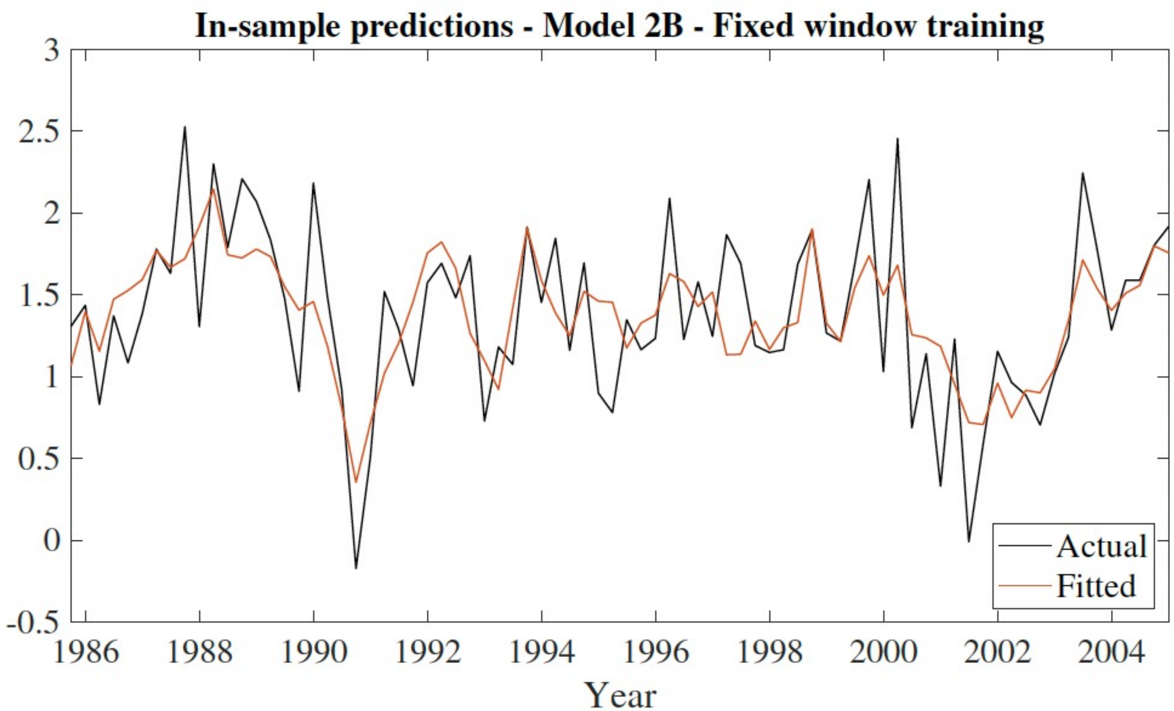


Figure 9: In-sample predictions of Model 2B showing the fitted value for the targeted quarterly GDP growth rate. For Model 2B, the states are synced according to the monthly frequency. The optimized hyperparameters are $N_x = 15$, $\alpha^Q = 0.47$ and $\alpha^M = 0.16$.

Empirical Findings

Model 2B: High Frequency State Sync

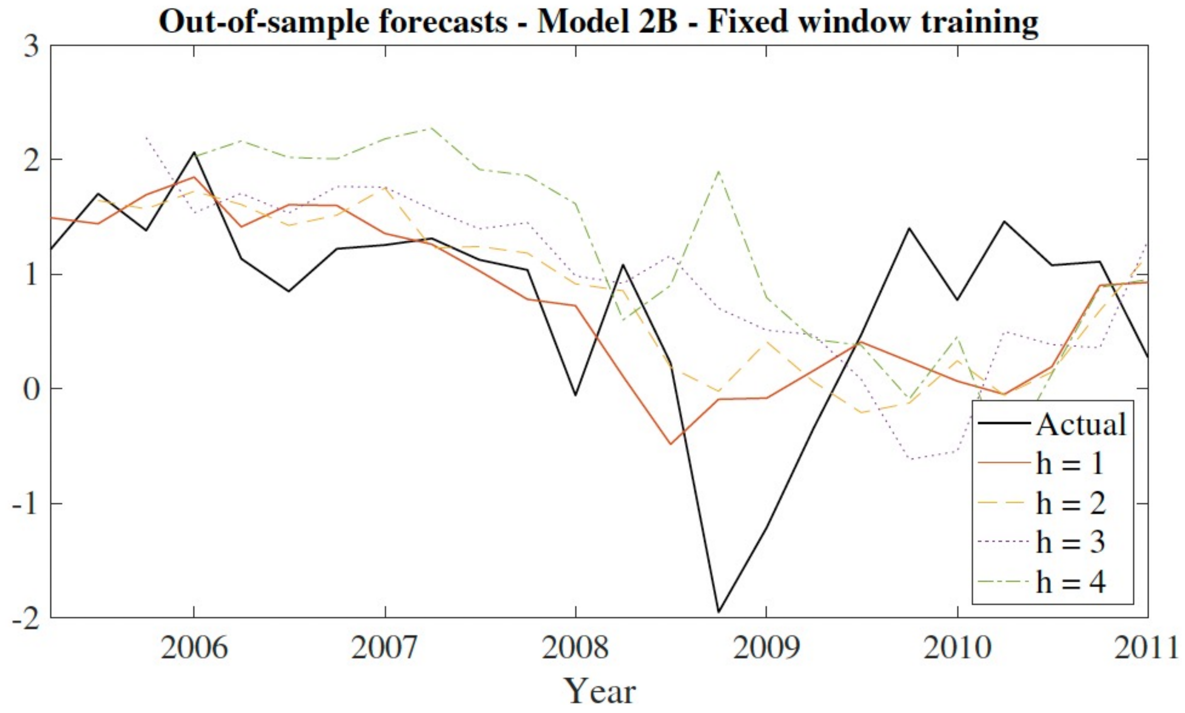


Figure 10: Out-of-sample forecasts of Model 2B model showing the forecasted values for the targeted quarterly GDP growth rate. These are h-step ahead forecasts using fixed window training, hence the network is trained only once on the training data. For Model 2B, the states are synced according to the monthly frequency. The optimized hyperparameters are $N_x = 15$, $\alpha^Q = 0.47$ and $\alpha^M = 0.16$.

Empirical Findings

Model 2B: High Frequency State Sync

- Out of sample performance of the Model 2B:

	MSE h=1	MSE h=2	MSE h=3	MSE h=4
Model 2B	0.57	0.69	1.10	1.77
UMIDAS	0.79	1.07	1.23	1.32

Table 7: Out-of-sample MSE performance of Model 2B benchmarked against UMIDAS model for $h \in \{1, 2, 3, 4\}$ step ahead forecasts using fixed window training.

- Economic intuition - Fitted data very close to large dips and spikes
- Low penalization hence less smooth curve
- Out-of-sample performance is relatively better for shorter horizons
- The leakage rate confirms the assertion made in Model 2A

Empirical Findings

Model 3 – Unstacked Input Model

Neurons	Validation Error	γ	ρ	α	λ	b
10	0.21	1.05	1.32	0.76	0.02	0.09
15	0.21	11.40	2.00	0.06	0.02	53.86
20	0.17	1.28	1.66	0.74	1.77	-1.44
25	0.23	0.67	0.50	0.44	0.01	-0.70
30	0.23	1.78	1.02	0.79	0.08	2.62
50	0.20	1.94	1.80	0.31	2.50	1.00

Table 8: Reservoir Dynamics for Model 3 using fixed window hyperparameter optimization. Based on the size of the reservoir, the table contains the optimized value of hyperparameters that are tuned using fixed window optimization technique. A *5-fold* cross validation approach is used to select the size of the reservoir based on the lowest validation error.

Empirical Findings

Model 3: Unstacked Input Model

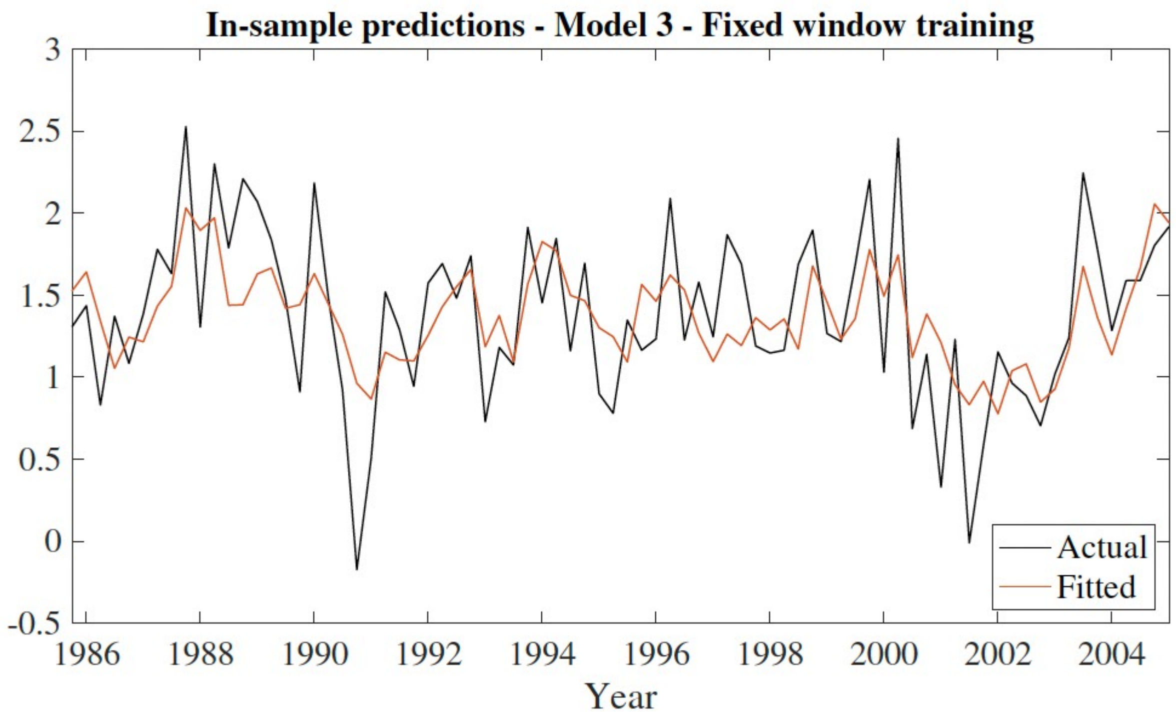


Figure 11: In-sample predictions of Model 3 showing the fitted value for the targeted quarterly GDP growth rate. For Model 3, the quarterly value is repeated using m = 3 frequency interval to synchronize with the monthly data. The optimized hyperparameters are $N_x = 20$ and $\alpha = 0.74$.

Empirical Findings

Model 3: Unstacked Input Model

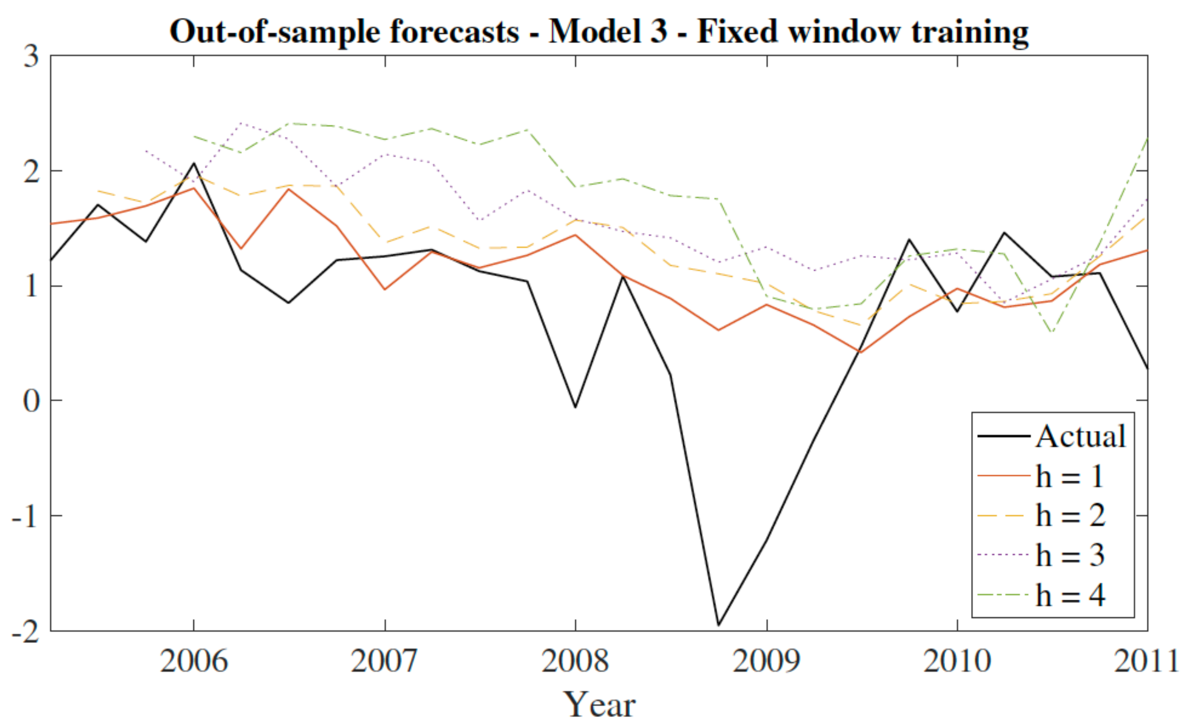


Figure 12: Out-of-sample forecasts of Model 3 model showing the forecasted values for the targeted quarterly GDP growth rate. These are h-step ahead forecasts using fixed window training, hence the network is trained only once on the training data. The quarterly value is repeated using m = 3 frequency interval to synchronize with the monthly data. The optimized hyperparameters are $N_x = 20$ and $\alpha = 0.74$.

Empirical Findings

Model 3: Unstacked Input Model

- Out of sample performance of the Model 3:

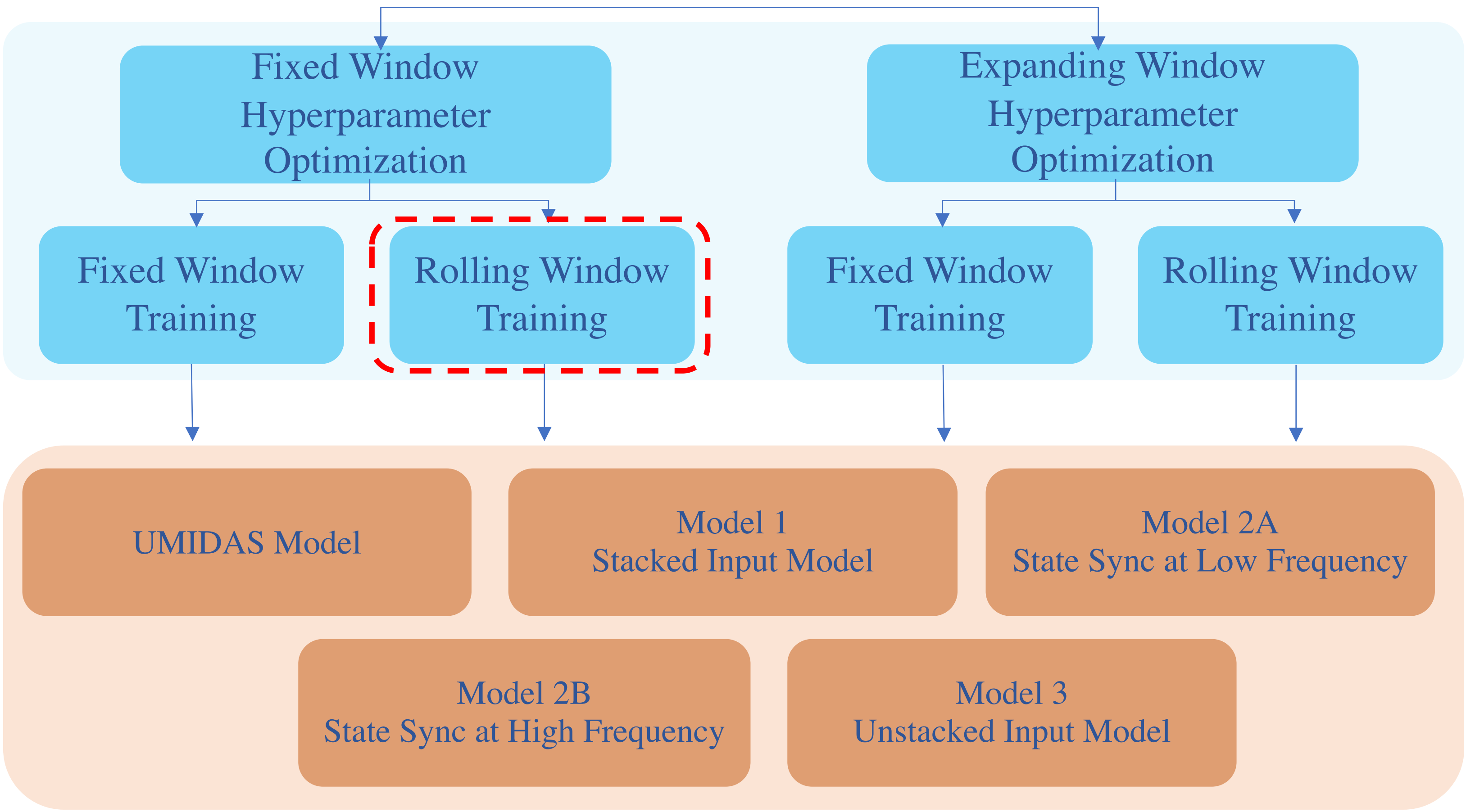
	MSE h=1	MSE h=2	MSE h=3	MSE h=4
Model 3	0.74	1.03	1.51	1.95
UMIDAS	0.79	1.07	1.23	1.32

Table 9: Out-of-sample MSE performance of Model 3 benchmarked against UMIDAS model for $h \in \{1, 2, 3, 4\}$ step ahead forecasts using fixed window training.

- The co-movement corresponding to actual data is sharp
- The leakage rate is higher which is counter-intuitive
- Out-of-sample performance is relatively better for shorter horizons
- Amplification in forecasts for higher forecasting horizon

Empirical Findings

Structure of results



Empirical Findings

UMIDAS Model

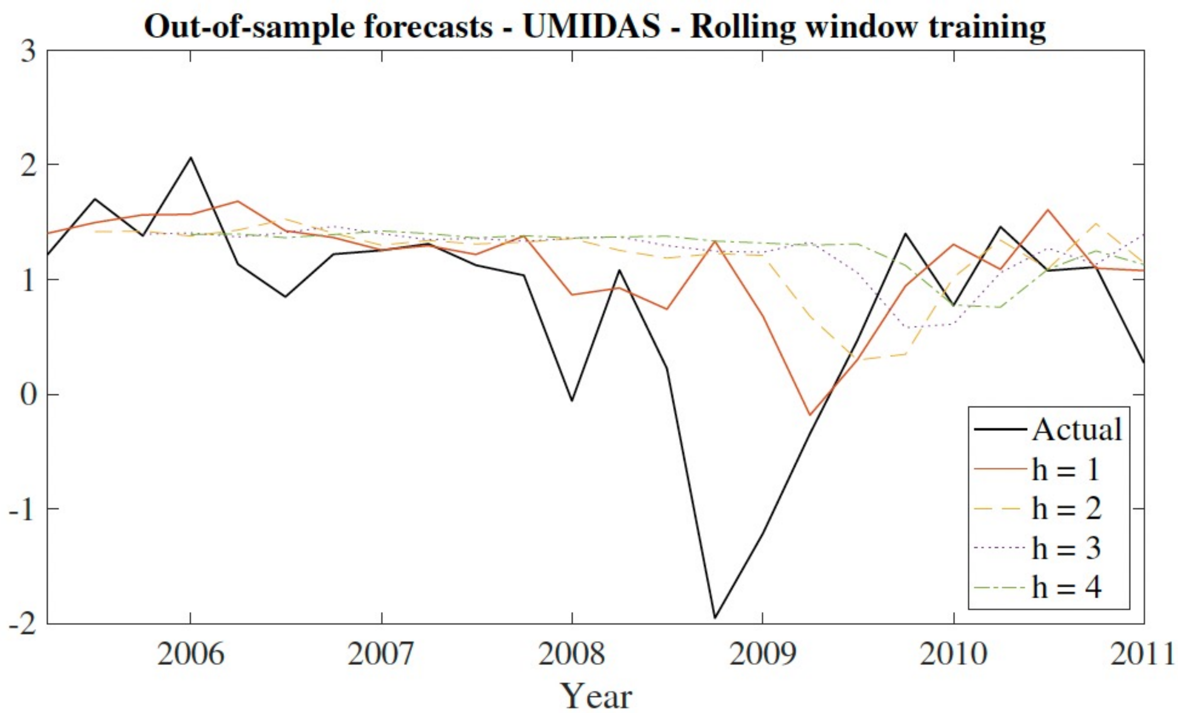


Figure 13: Out-of-sample forecasts of UMIDAS model showing the forecasted value for the targeted quarterly GDP growth rate. These are h-step ahead forecasts using rolling window training, hence the readouts are re-estimated in each iteration.

Empirical Findings

UMIDAS Model

- Out of sample performance of the UMIDAS model:

	MSE: h=1	MSE: h=2	MSE: h=3	MSE: h=4
UMIDAS	0.76	1.01	1.17	1.26

Table 10: Out-of-sample MSE performance of UMIDAS for $h \in \{1, 2, 3, 4\}$ step ahead forecasts. The MSE values are calculated for the quarterly GDP growth rates.

- The performance is relatively better than fixed window training
- Period after Q4 2008 is well adjusted in the out-of-sample forecasts
- Values for higher forecasting horizon is in closer to the actual value

Empirical Findings

Model 1 – Stacked Input Model

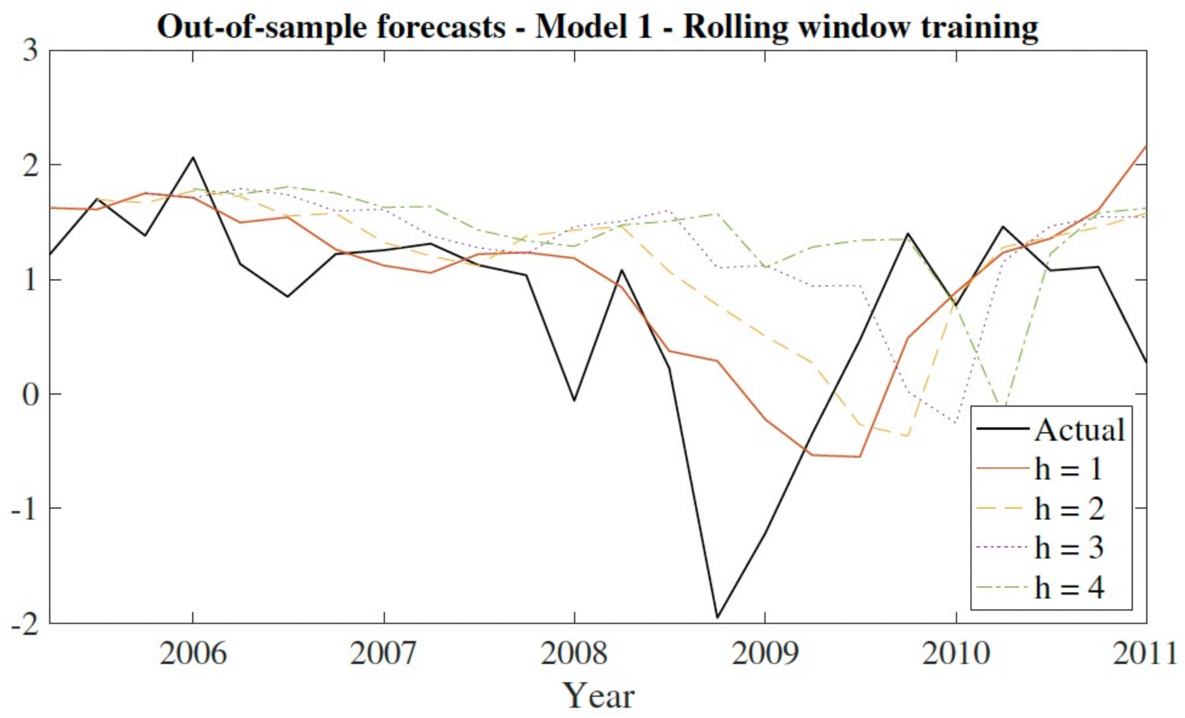


Figure 14: Out-of-sample forecasts of Model 1 model showing the forecasted values for the targeted quarterly GDP growth rate. These are h-step ahead forecasts using rolling window training, hence the readouts are re-estimated in each iteration. For Model 1, the monthly data are stacked according to quarterly frequency. The reservoir has $N_x = 15$ and $\alpha = 0.14$

Empirical Findings

Model 1 – Stacked Input Model

- Out of sample performance of the Model 1:

	MSE h=1	MSE h=2	MSE h=3	MSE h=4
Model 1	0.61	0.90	1.27	1.5
UMIDAS	0.76	1.01	1.17	1.26

Table 11: Out-of-sample MSE performance of Model 1 benchmarked against UMIDAS model for $h \in \{1, 2, 3, 4\}$ step ahead forecasts using rolling window training.

Empirical Findings

Model 2A: Low Frequency State Sync

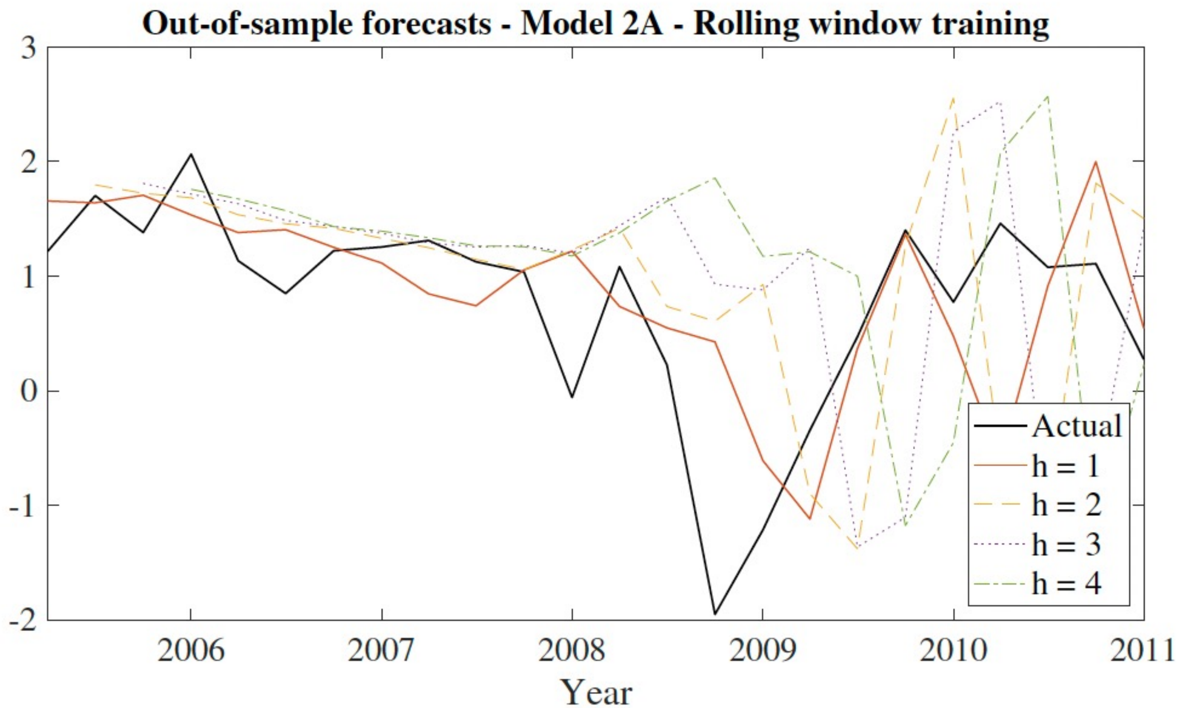


Figure 15: Out-of-sample forecasts of Model 2A model showing the forecasted values for the targeted quarterly GDP growth rate. These are h-step ahead forecasts using rolling window training, hence the readouts are re-estimated in each iteration. For Model 2A, the states are synced according to the quarterly frequency. The optimized hyperparameters are $N_x = 20$, $\alpha^Q = 0.25$ and $\alpha^M = 0.18$.

Empirical Findings

Model 2A: Low Frequency State Sync

- Out of sample performance of the Model 2A:

	MSE h=1	MSE h=2	MSE h=3	MSE h=4
Model 2A	0.63	1.38	1.96	2.10
UMIDAS	0.76	1.01	1.17	1.26

Table 12: Out-of-sample MSE performance of Model 2A benchmarked against UMIDAS model for $h \in \{1, 2, 3, 4\}$ step ahead forecasts using fixed window training.

Empirical Findings

Model 2B: High Frequency State Sync

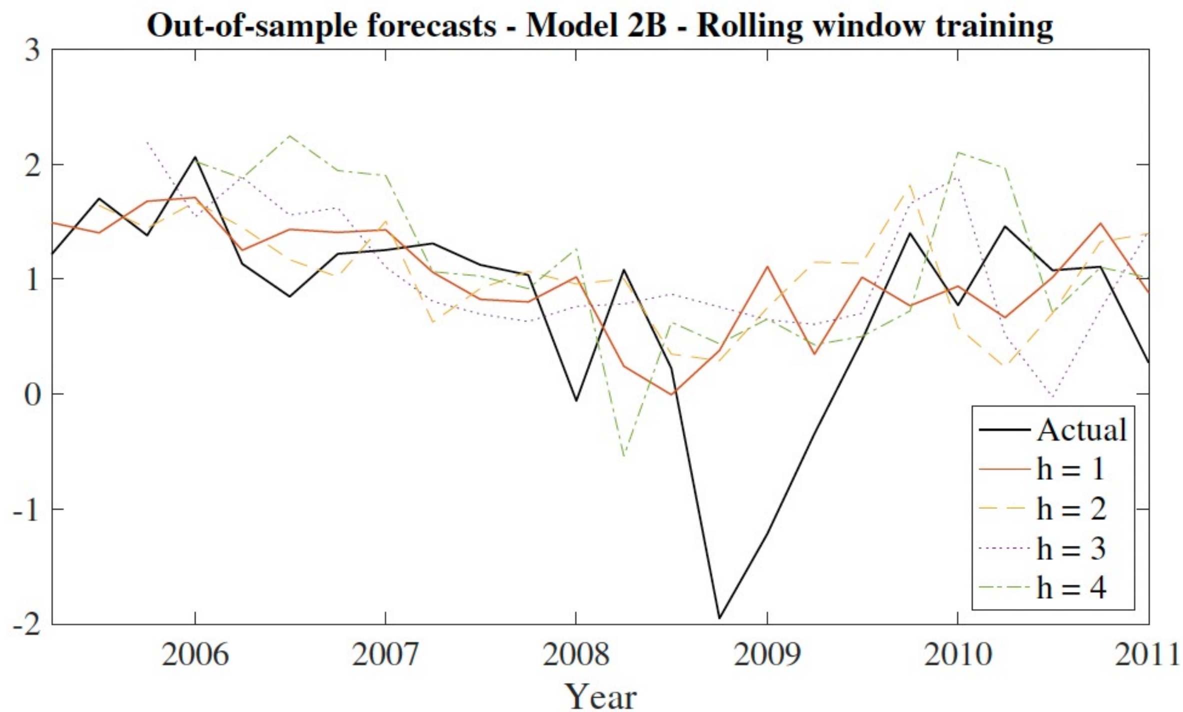


Figure 16: Out-of-sample forecasts of Model 2B model showing the forecasted values for the targeted quarterly GDP growth rate. These are h-step ahead forecasts using rolling window training, hence the readouts are re-estimated in each iteration. For Model 2B, the states are synced according to the monthly frequency. The optimized hyperparameters are $N_x = 15$, $\alpha^Q = 0.47$ and $\alpha^M = 0.16$.

Empirical Findings

Model 2B: High Frequency State Sync

- Out of sample performance of the Model 2B:

	MSE: h=1	MSE: h=2	MSE: h=3	MSE: h=4
Model 2B	0.67	0.72	0.94	1.022
UMIDAS	0.76	1.01	1.17	1.26

Table 13: Out-of-sample MSE performance of Model 2B benchmarked against UMIDAS model for $h \in \{1, 2, 3, 4\}$ step ahead forecasts using fixed window training.

Empirical Findings

Model 3: Unstacked Input Model

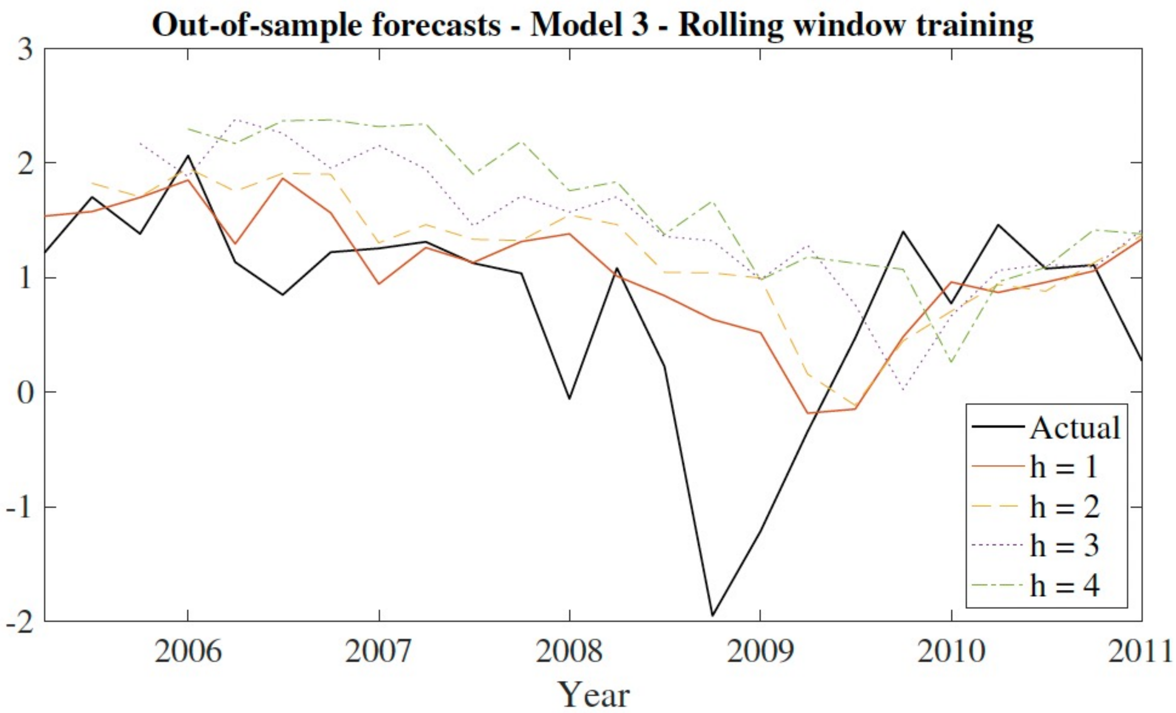


Figure 17: Out-of-sample forecasts of Model 3 model showing the forecasted values for the targeted quarterly GDP growth rate. These are h-step ahead forecasts using rolling window training, hence the readouts are re-estimated in each iteration. The quarterly value is repeated using m = 3 frequency interval to synchronize with the monthly data. The optimized hyperparameters are $N_x = 20$ and $\alpha = 0.74$.

Empirical Findings

Model 3: Unstacked Input Model

- Out of sample performance of the Model 3:

	MSE h=1	MSE h=2	MSE h=3	MSE h=4
Model 3	0.69	0.97	1.47	1.75
UMIDAS	0.76	1.01	1.17	1.26

Table 14: Out-of-sample MSE performance of Model 3 benchmarked against UMIDAS model for $h \in \{1, 2, 3, 4\}$ step ahead forecasts using fixed window training.

Empirical Summary

Fixed Window Training				
	h=1	**h=2**	**h=3**	**h=4**
Model 1	0.83	1.14	1.41	1.60
Model 2A	0.90	1.40	1.92	2.36
Model 2B	**0.57**	**0.69**	**1.10**	1.77
Model 3	**0.75**	**1.04**	1.51	1.96
UMIDAS	0.79	1.07	1.23	1.32

Rolling Window Training				
	h=1	**h=2**	**h=3**	**h=4**
Model 1	**0.61**	**0.90**	1.27	1.51
Model 2A	**0.63**	1.38	1.96	2.10
Model 2B	**0.67**	**0.73**	**0.94**	**1.00**
Model 3	**0.69**	**0.98**	1.48	1.75
UMIDAS	0.76	1.02	1.17	1.26

Table 15: A model wise out-of-sample performance for the two training approaches using reservoir dynamics tuned from fixed window hyperparameter optimization.

Conclusion

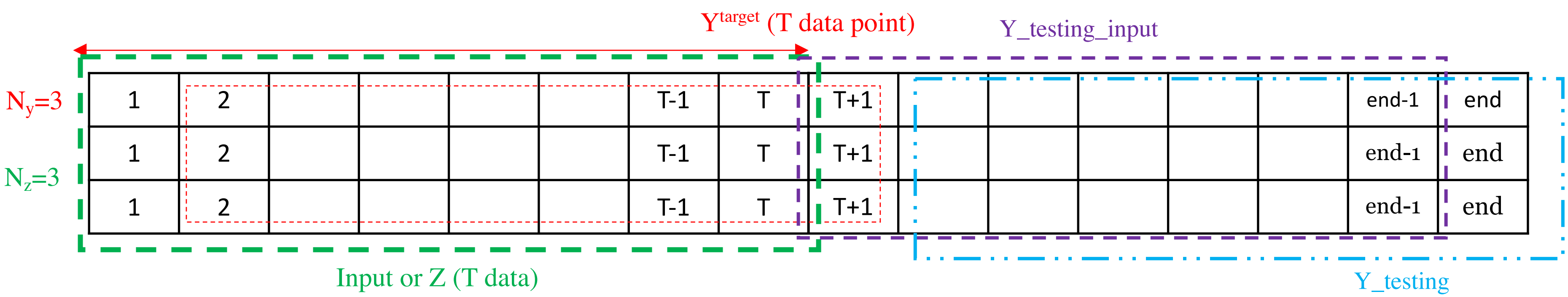
- Proposed 4 different models to incorporate mixed frequency data in ESNs
- Preference to operate at higher frequencies
  - Models operating at higher frequency have relatively better performance
- Potentially capture the explanatory power of variables used
- Extension of models to Nowcasting literature
  - Alignment of data by shifting the low frequency period ahead
  - Using HF data in the same LF period that we are forecasting

Limitations and potential research opportunities

- Effectively tuning the reservoir dynamics
    - Quantum of data
    - Hyperparameter optimization: Fixed vs Expanding
- Extending the models to datasets with larger frequency interval
- Accommodating for higher lags in the input signal
- Adding ‘depth’ in the reservoirs

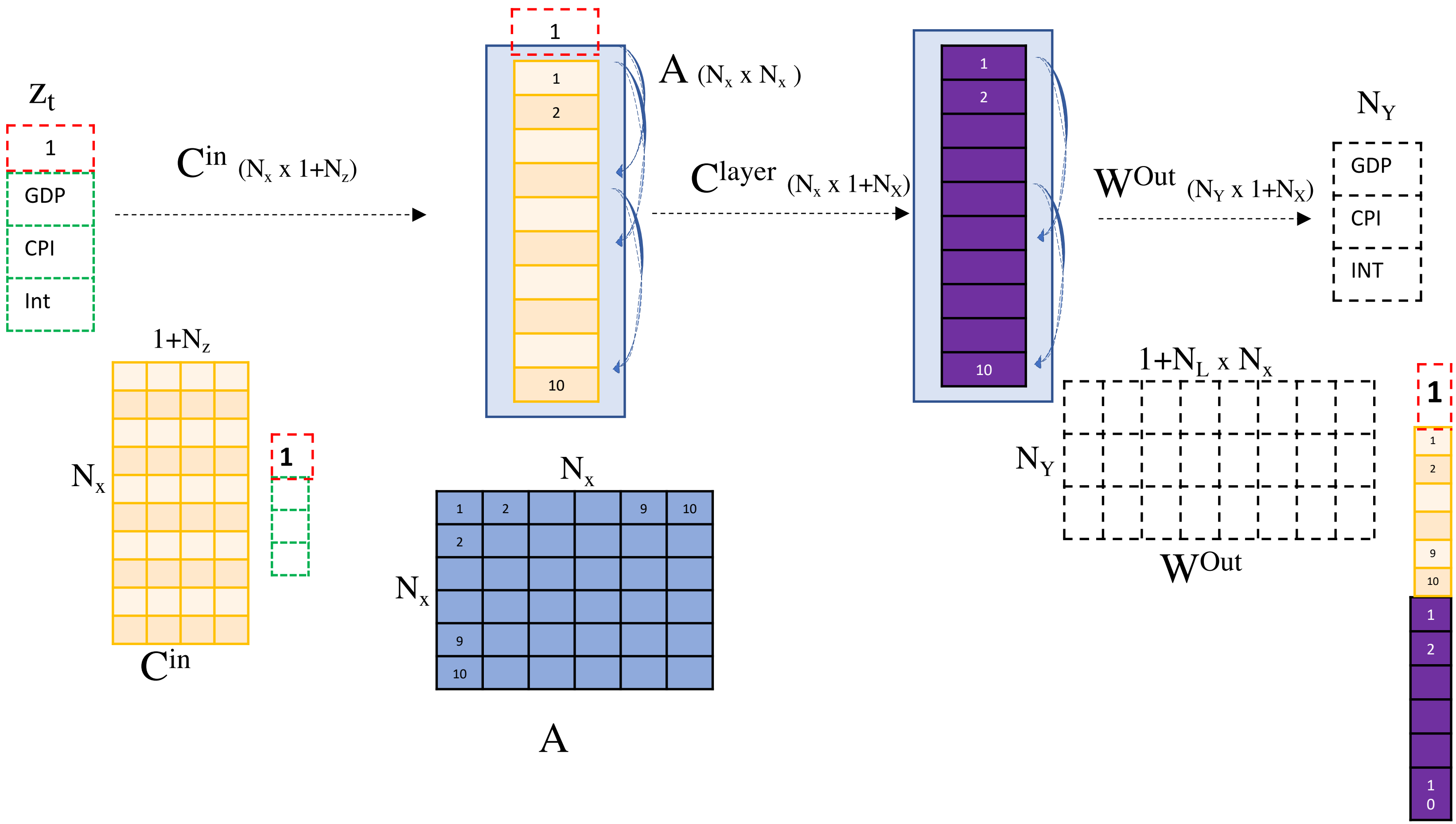
Appendix

Schematic of the network



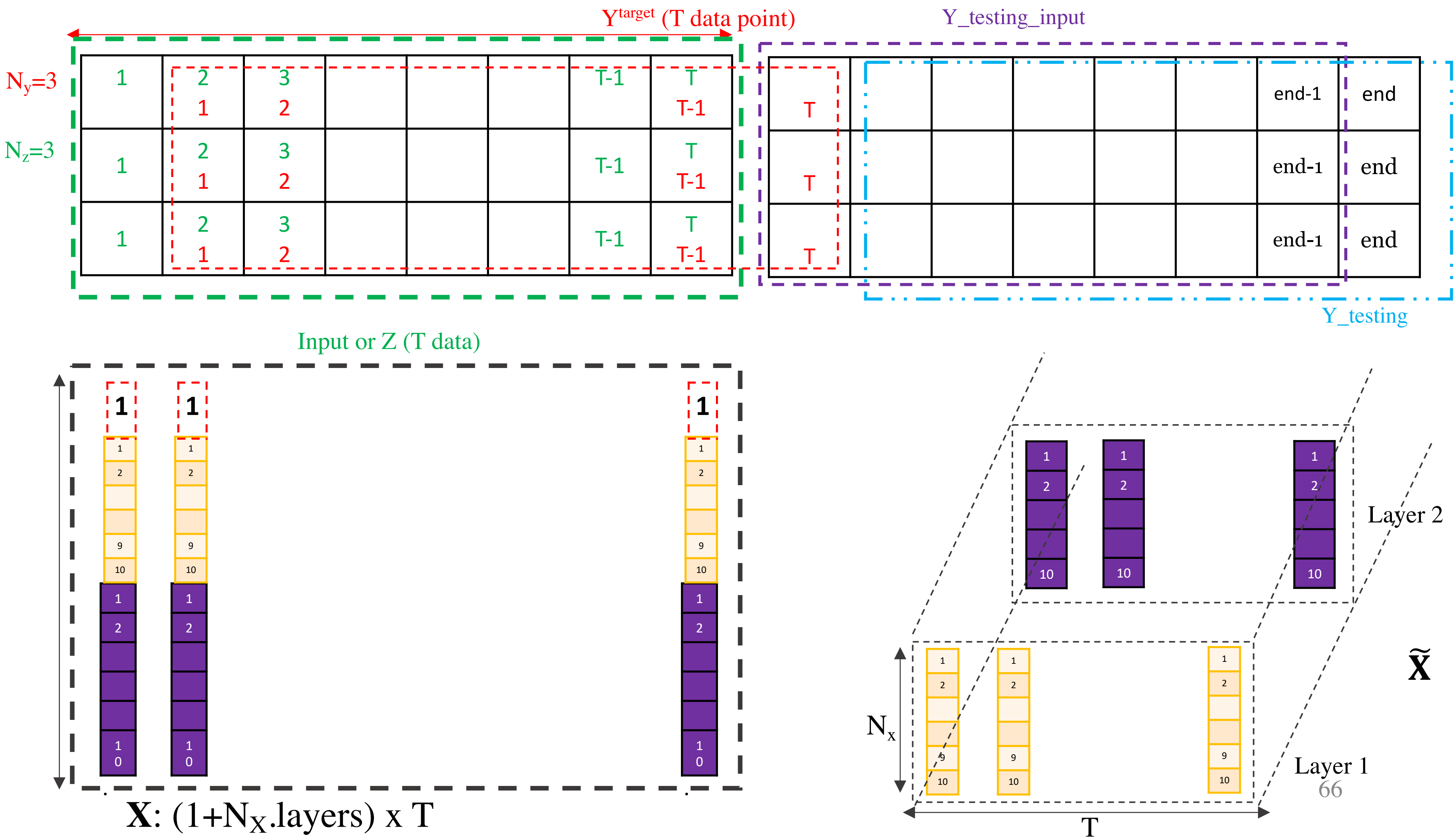
Appendix

Schematic of the network



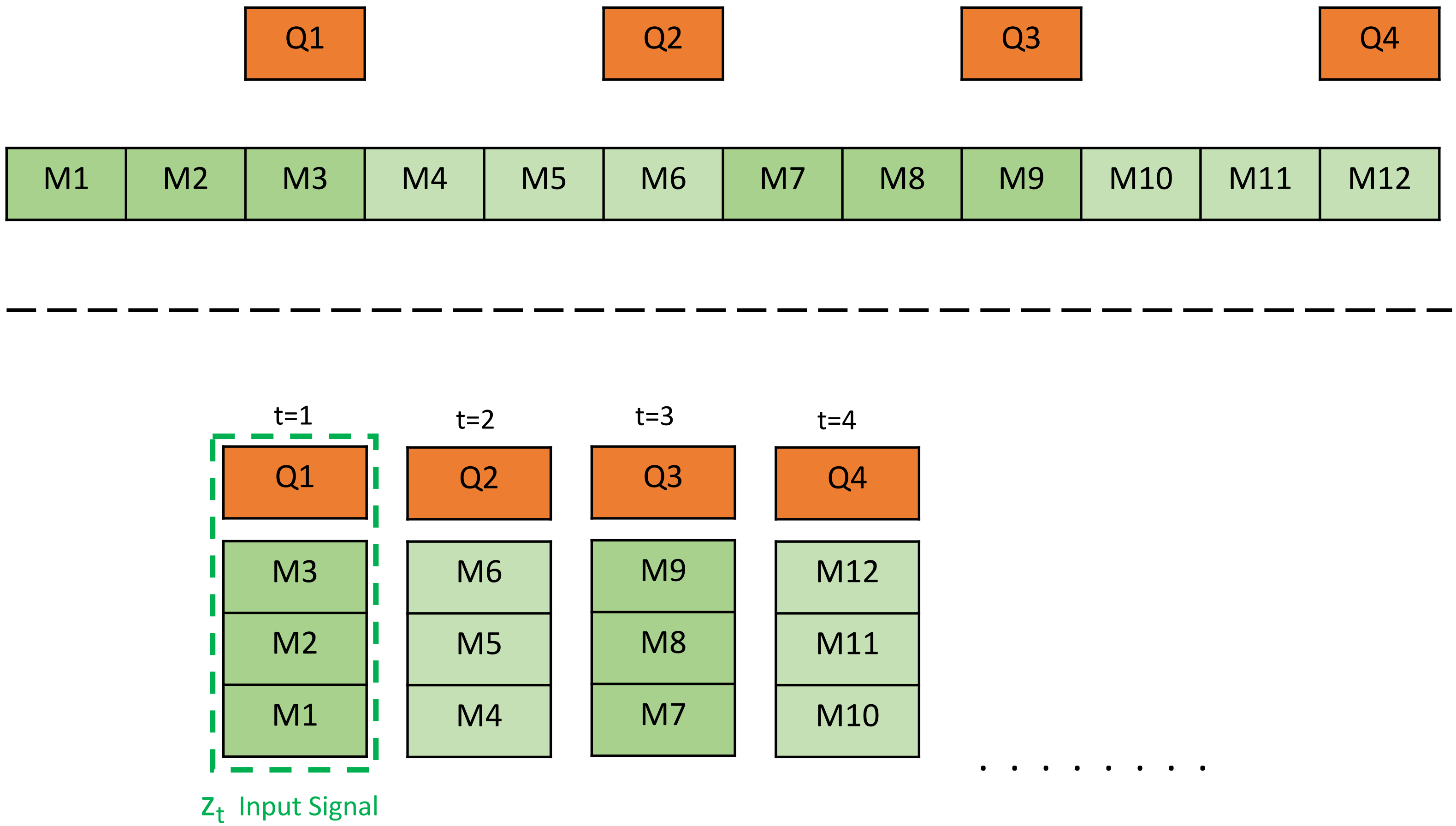
Appendix

Understanding the states of the reservoir



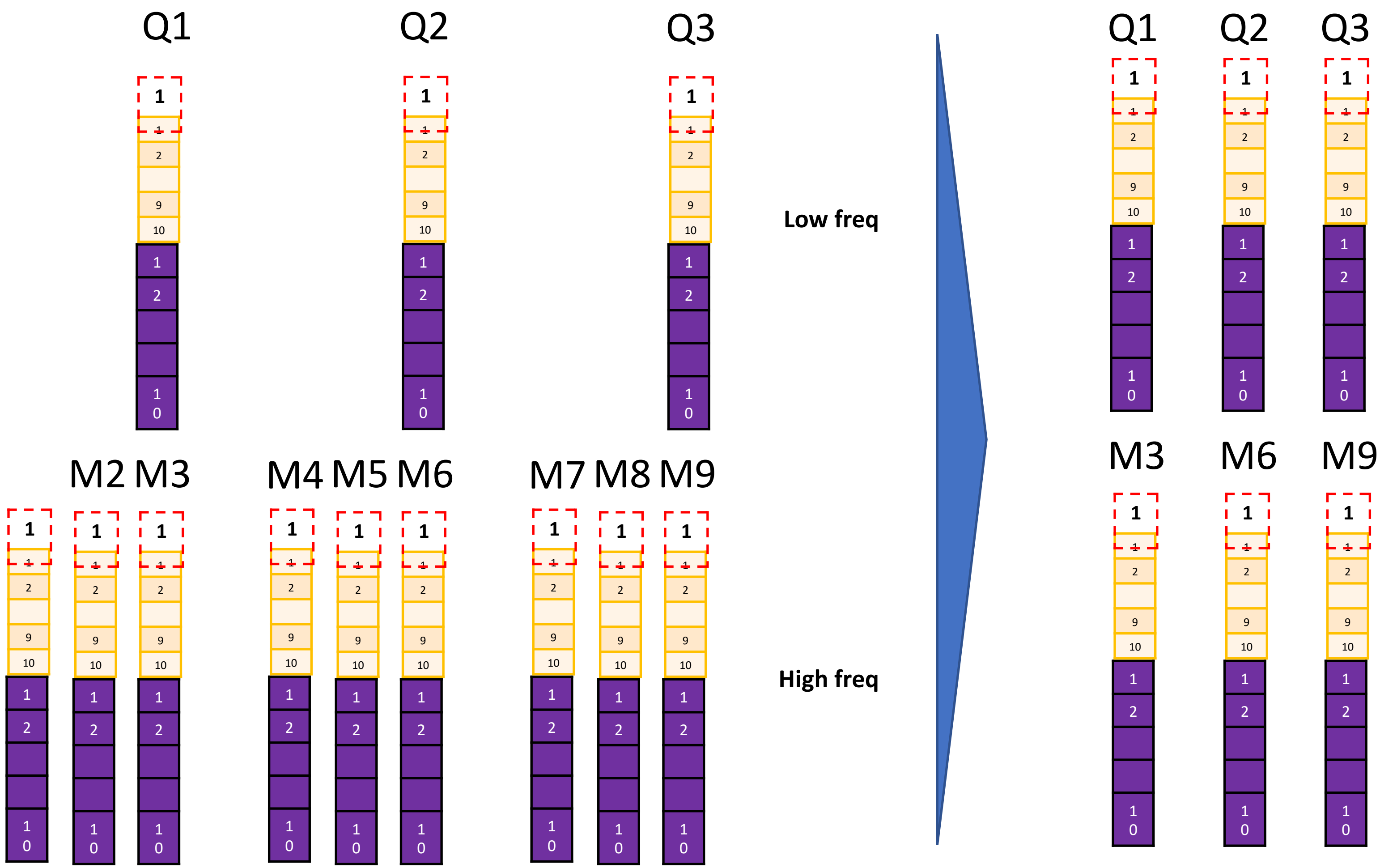
Appendix

Model 1



Appendix

Model 2A



Appendix

Model 3