Nowcasting Japanese GDP Using News Data and Machine Learning*

Yusuke Oh[1] and Mototsugu Shintani†[1]

[1]The University of Tokyo

This version: August 2024

Abstract

We conduct a nowcasting analysis on Japan's GDP using machine learning. We employ machine learning approaches because the estimation of the mixed-data sampling (MIDAS) models without parameter restriction, in the presence of text information extracted from newspapers potentially involves high-dimensional data. Based on the unrestricted MIDAS model with macroeconomic indicators, survey-based indicators, and text information, we find that text information helps improve nowcasting performance during the COVID-19 period. In addition, the forecasting combination of machine learning and dynamic factor model has the potential to outperform using either method alone.

Keywords: nowcasting, machine learning, mixed frequency data, news, text data

JEL Classification: C32, C53, E37, O53

*The authors would like to thank participants at the the 16th International Conference on Computational and Financial Econometrics (CFE 2022) for their helpful comments and discussions. Shintani gratefully acknowledges the financial support of the Grant-in-Aid for Scientific Research (20H01482).

†Corresponding author, E-mail: shintani@e.u-tokyo.ac.jp

1 Introduction

We employ machine learning methods to estimate an unrestricted mixed-data sampling model for nowcasting the growth rate of Japan's gross domestic product (GDP). Our set of predictors includes 'hard' data, consisting of basic monthly and quarterly macroeconomic variables, survey-based indicators ('soft' data), and news data constructed from text information in news articles. Since the target variable is quarterly data and the predictors are both monthly and quarterly data, our approach involves estimating mixed frequency data models. We use machine learning methods because they do not require any parametric restrictions on the weight function in mixed-data sampling models (U-MIDAS).

The machine learning approach is also advantageous for incorporating text information into the forecast model, as such predictors can be high-dimensional. This contrasts with the work of Hayashi and Tachi (2023), who use the maximum likelihood method to estimate the dynamic factor model (DFM) for nowcasting the growth rate of Japan's GDP based on 'hard' and 'soft' data.

There are two main methods for handling mixed-frequency data when estimating the model. In the first method, low-frequency data are converted into high-frequency data by augmenting the low-frequency observations. The model is then estimated using the high-frequency data. For example, in the case of stock and price variables, a hypothetical high-frequency observed value is generated by linear interpolation of observed values. For flow variables and rates of change, the values can be divided equally into each period. The maximum likelihood estimation of DFM, employed by Hayashi and Tachi (2023), falls under this method, as the high-frequency observations are treated as latent variables in the state space model and their values are estimated.

In the second method, high-frequency data are aggregated and all variables are expressed in terms of low-frequency data. The prediction using MIDAS models, originally proposed by Ghysels, Santa-Clara, and Valkanov (2005), falls under this method. In particular, we employ the unrestricted MIDAS (U-MIDAS) model, as considered by Foroni, Marcellino, and Schumacher (2015), and estimate the model using various types of ma-

chine learning. To nowcast the Japanese economy, the U-MIDAS model was used by Chikamatsu et al. (2021). However, they did not incorporate text information in their analysis.

Our paper is closely related to the literature on nowcasting and forecasting macroeconomic variables using text data. For example, Goshima et al. (2021) used newspaper article data and found that text information is useful in forecasting Japanese inflation. Ellingsen, Larsen, and Thorsrud (2022) found that news-based data contains valuable information not captured by hard economic data when forecasting US consumption. Kalamara et al. (2022) claimed that newspaper text data are useful in forecasting key macroeconomic variables in the UK.

Our analysis is also related to an increasing number of studies that employ machine learning methods in macroeconomic forecasting. For example, Diebold and Shin (2019) employed Lasso and Ridge regression and their extensions for forecasting Euro area GDP. Giannone, Lenza, and Primiceri (2021) applied a Bayesian forecasting framework that included Lasso, Ridge, and Elastic Net to hard macroeconomic data, among other series, to evaluate the usefulness of sparse modeling.

The effectiveness of ensemble machine learning methods based on regression trees, such as random forests, was emphasized by Medeiros et al. (2021) and Chen et al. (2022). Bai and Ng (2009) examined the effectiveness of boosting in forecasting inflation, interest rates, industrial production, employment, and the unemployment rate using a large set of US macroeconomic data.

Furthermore, Kim and Swanson (2018), Gu, Kelly, and Xiu (2020), Maehashi and Shintani (2020), and Coulombe et al. (2022) conducted horse race analyses using various machine learning methods to forecast macroeconomic and financial variables.

The rest of the paper is organized as follows. Section 2 introduces the model for mixed frequency data and the machine learning methods used in the analysis. In Section 3, we describe the data and procedures for evaluating nowcast performance. Section 4 presents the main empirical results, followed by concluding remarks in Section 5.

2 Mixed frequency data and machine learning

2.1 Unrestricted MIDAS model

We consider the problem of predicting the quarterly target variable using monthly predictors. In practice, we need to incorporate the fact that the timing of the release differs across the predictors. This issue is often referred to as the ragged-edge data problem (see Bańbura, Giannone and Reichlin, 2011). For notational simplicity, we assume that the one predictor, x_t, becomes available exactly one month after the end of reference month, and the other predictor, z_t, is observed in the same month. Most of the hard data we use in the analysis are represented by the former type predictor x_t. In contrast, news data and financial market series can be represented by the latter type of predictor z_t. While we use multiple predictors with more than two types of data release timing in the empirical analysis, the following discussion can be extended with a simple modification.

Since the predictor is observed monthly, we denote the time index of predictor variable x_t and z_t to take multiples of 1/3. On the other hand, since the target variable is observed quarterly, we let its time subscript of y_t to take integer values. The mixed-data sampling (MIDAS) model for h-period-ahead forecast of y_t using z_t as a predictor is given by:

$$y_{t+h} = \mu_{h,0} + \beta_{h,1} \sum_{j=0}^{p_z-1} w_j(\theta) L^{j/3} z_t + \varepsilon_{t+h} \tag{1}$$

where $w_j(\theta) = e^{\theta_1 j + \cdots + \theta_q j^q} / (\sum_{i=0}^{p_z} e^{\theta_1 i + \cdots + \theta_q i^q})$ is the weighted moving average weight function, $\theta = (\theta_1, ...\theta_q)$ are the parameters of the weight function. For example, in Ghysels, Santa-Clara, and Valkanov (2005), $q = 2$ and the weight function $w_j(\theta_1, \theta_2) = e^{\theta_1 j + \theta_2 j^2} / (\sum_{i=0}^{p_z-1} e^{\theta_1 i + \theta_2 i^2})$ is employed.[1] Since the sum of the weights is given by $\sum_{i=0}^{p_z-1} w_j(\theta) = 1$, with the choice of $p_z = 3$ and $\theta_1 = \cdots = \theta_q = 0$, we can confirm x_t^Q corresponds to the 3-month average. In the MIDAS model, since the coefficients $(\mu_h, \beta_{h,1})$ of the (1) equation and the weight function parameters $\theta = (\theta_1, ..., \theta_q)$ are unknown, they are estimated simultaneously using nonlinear least squares. The benchmark MIDAS regression

[1] See also Ghysels and Valkanov (2006) and Clements and Galvão (2008).

model (1) can be extended to the model with the lagged dependent variables as additional predictors given by

$$y_{t+h} = \mu_h + \sum_{j=0}^{p_y-1} \phi_{h,j} L^j y_t + \beta_{h,1} \sum_{j=0}^{p_z-1} w_j(\theta) L^{j/3} z_t + \varepsilon_{t+h}. \tag{2}$$

Andreou, Ghysels and Kourtellos (2013) refer (2) to the ADL-MIDAS regression model.

By expanding equations (1) and (2), we obtain the unrestricted MIDAS (U-MIDAS) models given by

$$y_{t+h} = \mu_h + \sum_{j=0}^{p_z-1} \delta_{h,j} L^{j/3} z_t + \varepsilon_{t+h} \tag{3}$$

and

$$y_{t+h} = \mu_h + \sum_{j=0}^{p_y-1} \phi_{h,j} L^j y_t + \sum_{j=0}^{p_z-1} \delta_{h,j} L^{j/3} z_t + \varepsilon_{t+h}, \tag{4}$$

respectively.

Note that the number of parameters of the U-MIDAS model in equation (3) is $p_z + 1$ while the number of parameters of the MIDAS model in equation (1) is $q+2$. For the case of N monthly predictors, the number of parameters $Np_z + 1$ can become large for a large N. However, a U-MIDAS model with a large number of parameters can be estimated by using machine learning procedures explained in the next subsection.

It should also be note that for the same target quarterly GDP, available observations differ depending on whether the timing of nowcast (or forecast) is either at the first, second or third month of a quarter. Such a different information structures, along with the ragged-edge problem of additional predictor x_t, can be incorporated by adjusting the index j on the coefficient of each predictor in (3) and (4) to begin with $\ell(> 0)$ instead of 0 where ℓ represents the information delay.

In summary, we consider the following three types of the U-MIDAS model to construct the nowcast ($h = 0$) of Japanese GDP. Without the loss of generality, predictors are denoted by y_t for the lagged quarterly GDP; x_t for monthly variables with one month information delay (most hard data); and z_t for monthly variables with no information delay (news data and policy rate).

1. M1 (End-of-month 1) type model

$$y_{t+h} = \mu_h + \sum_{j=2}^{p_y+1} \phi_{h,j} L^j y_t + \sum_{j=3}^{p_x+2} \delta_{h,j} L^{j/3} x_t + \sum_{j=2}^{p_z+1} \gamma_{h,j} L^{j/3} z_t + \varepsilon_{t+h} \tag{5}$$

2. M2 (End-of-Month 2) type model

$$y_{t+h} = \mu_h + \sum_{j=1}^{p_y} \phi_{h,j} L^j y_t + \sum_{j=2}^{p_x+1} \delta_{h,j} L^{j/3} x_t + \sum_{j=1}^{p_z} \gamma_{h,j} L^{j/3} z_t + \varepsilon_{t+h} \tag{6}$$

3. M3 (End-of-Month 3) type model

$$y_{t+h} = \mu_h + \sum_{j=1}^{p_y} \phi_{h,j} L^j y_t + \sum_{j=1}^{p_x} \delta_{h,j} L^{j/3} x_t + \sum_{j=0}^{p_z-1} \gamma_{h,j} L^{j/3} z_t + \varepsilon_{t+h} \tag{7}$$

Here, M1 type model is estimated only using observations from February 1, May 1, August 1, and November 1 in each year. Similarly, M2 type model uses observations from March 1, June 1, September 1, and December 1 in each year, while M3 type model uses observations from January 1, April 1, July 1, and October 1 in each year. For the lag length parameters, we impose the following restrictions: $p_z \geq 1$ for M1 type model; $p_x \geq 1$ and $p_z \geq 2$ for M2 type model; and $p_x \geq 2$ and $p_z \geq 3$ for M3 type model.

2.2 Machine learning

We consider a set of machine learning (ML) methods to estimate unrestricted MIDAS (U-MIDAS) models. We divide the machine learning methods available for macroeconomic forecasting into four major types: (i) regularized least squares estimators, (ii) support vector regression, (iii) tree-based methods, and (iv) neural networks. Below, we describe each type separately.

2.2.1 Regularized least squares estimator

The first type is regularized least squares estimators, which add a penalty term, or regularization term, to the objective function to prevent overfitting. In this paper, we focus

on the elastic net estimator.

Consider a simple linear regression model with a target variable y_{t+h} and predictors $X_t = [x_{1t}, x_{2t}, ...x_{Nt}]$

$$y_{t+h} = \beta_0 + \sum_{i=1}^{N} \beta_i x_{it} + \varepsilon_{t+h} \tag{8}$$

Minimizing the residual sum of squares yields the OLS estimator of $\beta = [\beta_0, \beta_1, ...\beta N]$ given by

$$\widehat{\beta}_{OLS} = \arg\min \sum_{t=1}^{T} \left(y_{t+h} - \beta_0 - \sum_{i=1}^{N} \beta_i x_{it} \right)^2. \tag{9}$$

The elastic net estimator uses a linear combination of the L_1 norm $\sum_{i=1}^{N} |\beta_i|$ and the L_2 norm, defined as follows:

$$\widehat{\beta}_{enet} = \arg\min \left[\sum_{t=1}^{T} \left(y_{t+h} - \beta_0 - \sum_{i=1}^{N} \beta_i x_{it} \right)^2 + \omega\lambda \sum_{i=1}^{N} |\beta_i| + (1-\omega)\lambda \sum_{i=1}^{N} \beta_i^2 \right]. \tag{10}$$

The additional adjustment parameter ω controls the relative weight between the L_1 norm penalty and the L_2 norm penalty. Two adjustment parameters, ω and λ, can be selected simultaneously to minimize the MSE calculated by K-fold cross-validation.

By combining the penalties of the L_1 and L_2 norms, elastic net can handle multicollinearity well by shrinking coefficients and also perform variable selection by setting some coefficients to zero.

2.2.2 Support Vector Regression (SVR)

Support Vector Regression (SVR) is designed to fit a regression model within an allowable error margin ϵ, while maximizing model fit. It employs kernel functions, typically the Radial Basis Function (RBF), to handle nonlinear relationships. The model can be described by:

$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x, x_i) + b \tag{11}$$

where α_i and α_i^* are dual coefficients, $K(x, x_i)$ is the kernel function, and b is the bias. The RBF kernel is defined as:

$$K(x, x_i) = \exp(-\gamma \|x - x_i\|^2) \tag{12}$$

where γ determines the kernel's width, controlling the influence of each support vector.

SVR's objective is to minimize:

$$\frac{1}{2}\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)^2 K(x_i, x_i) + C\sum_{i=1}^{n}\max(0, |y_i - f(x_i)| - \epsilon) \tag{13}$$

where C is the regularization parameter. This configuration penalizes deviations larger than ϵ, ensuring the model is both flat and accurate within the epsilon margin.

SVR provides a robust alternative to traditional regression methods, particularly useful for datasets with complex nonlinear relationships or when the number of predictors exceeds the number of observations.

2.2.3 Random Forest and Boosting

The third type of ML methods is regression trees combined with ensemble learning. A regression tree with M terminal nodes can be written as

$$y_{t+h} = \sum_{m=1}^{M} \theta_m \mathbf{1}_{\{X_t \in R_m\}} + \varepsilon_{t+h} \tag{14}$$

where $\mathbf{1}_{\{\cdot\}}$ is the indicator function, R_m is the partition of the range of X_t, and θ_m is the mean of y_t conditional on $X_t \in R_m$.

Regression trees accommodate non-linearities but are prone to overfitting. Ensemble methods mitigate these issues and are useful in macroeconomic forecasting due to their ability to capture complex relationships and robustness to outliers in economic data.

Random forests, introduced by Breiman (2001), use bootstrap aggregating (bagging) and reduce correlation between tree predictions. The final forecast is the average of all bootstrap forecasts: $B^{-1}\sum_{b=1}^{B}\widehat{y}_{t+h}^{(b)}$, where only a subset of predictors are used at each node.

Boosting, introduced by Schapire (1990) and Freund (1995), estimates models sequen-

tially. At each stage s, the model is updated as:

$$g_s(X_t) = g_{s-1}(X_t) + \eta \sum_{m=1}^{M_s} \theta_{sm} \mathbf{1}_{\{X_t \in R_{sm}\}} \tag{15}$$

where η is the learning rate and the new tree is estimated using the previous residual.

We employ LightGBM for its computational efficiency, which is particularly beneficial when dealing with large-scale economic datasets.

2.2.4 Neural Networks

The fourth type covers models based on neural networks: Multi-Layer Perceptrons (MLPs), Convolutional Neural Networks (CNNs), and Long Short-Term Memory networks (LSTMs). MLPs are basic neural networks with interconnected layers. CNNs, adapted from image processing, use convolutional layers to identify important features in time series. LSTMs address long-term dependencies in sequence data.

For an MLP with one intermediate layer, the h-period-ahead forecast model is:

$$y_{t+h} = \sum_{j=1}^{q} \theta_j h_j(X_t) + b + \varepsilon_{t+h} \tag{16}$$

$$h_j(X_t) = \sigma(w_j' X_t + b_j) \tag{17}$$

where X_t and w_j are $N \times 1$ vectors, σ is the activation function, h_j is the hidden unit, and q is the number of hidden units.

A $n+1$-layer MLP (with n intermediate layers) is given by:

$$y_{t+h} = \theta^{(n)\prime} h^{(n)} + b^{(n)} + \varepsilon_{t+h} \tag{18}$$

$$\begin{aligned} h^{(n)} &= [\sigma(\theta_1^{(n-1)\prime} h^{(n-1)} + b_1^{(n-1)}), ..., \sigma(\theta_{q_n}^{(n-1)\prime} h^{(n-1)} + b_{q_n}^{(n-1)})]' \\ &\vdots \\ h^{(1)} &= [\sigma(w_1' X_t + b_1), ..., \sigma(w_{q_1}' X_t + b_{q_1})]' \end{aligned}$$

where $h^{(\ell)}$ is the vector of hidden units in the ℓ-th layer, $\theta_k^{(\ell)}$ are weight vectors, and $b_k^{(\ell)}$ are bias terms.

Common activation functions are sigmoid $\sigma(z) = \frac{1}{1+e^{-z}}$, tanh $\sigma(z) = \frac{e^z-e^{-z}}{e^z+e^{-z}}$, and ReLU $\sigma(z) = \max(0, z)$, where z is the input.[2]

CNNs replace the MLP's intermediate layer with convolutional and pooling layers. The convolutional layer applies a locally weighted sum using a filter, while the pooling layer reduces the output dimension. This structure is particularly useful for analyzing time series data like macroeconomic variables.

Let us set the forecast horizon h at 1 and assume that monthly time series $\{y_{t+1}, x_t\}_{t=1}^T$ is available, where y_{t+1} is a target variable and x_t is a single predictor. By introducing nonlinearity into a simple distributed lag model of order 12, $y_{t+1} = \mu + \sum_{j=0}^{11} \delta_j x_{t-j} + \varepsilon_{t+1}$, and extending to MLP with q number of hidden units, we obtain $y_{t+1} = \sum_{j=1}^{q} \theta_j h_j + b + \varepsilon_{t+1}$ where $h_j = \sigma(w_j' X_t + b_j)$ is a hidden unit and $X_t = (x_t, x_{t-1}, ., x_{t-11})'$ is the inputs. Since the total number of parameters is $q \times (12 + 1) + q + 1$, the number of parameters in MLP tends to be large even when the number of units is not so large. Let us now replace the intermediate layer of this MLP with the convolutional and pooling layers of the CNN. In the convolution layer, the locally weighted moving average of the 12 lag variables is normalized by the activation function. The weight of this local weighted average is called the filter, and its length is called the filter size. The usual filter size is an odd number, for example, for 3 months, the weighted moving average is calculated using the weights $w = (w_1, w_2, w_3)'$ (and the bias term b). In this case, the filter will be out of the data range at the endpoints, but this is handled by substituting 0 or not calculating a weighted moving average. For example, when the observed value of the target variable y_{13} at $t = 12$ and the observed value of the predictor variable $X_{12} = (x_{12}, x_{11}, ..., x_1)'$, we have $\{h_1^{(c)}, h_2^{(c)}, ..., h_{10}^{(c)}\} = \{\sigma(w'(x_{12}, x_{11}, x_{10})' + b), \sigma(w'(x_{11}, x_{10}, x_9)' + b), ..., \sigma(w'(x_3, x_2, x_1)' + b)\}$ if the endpoints are not calculated. Dimension of the 10 output values of the convolution layer are reduced in the sub-

[2]Traditionally, sigmoid functions were used for shallow networks. However, in deep learning, sigmoid functions can lead to vanishing gradients. Tanh functions reduce this problem, while ReLU functions avoid it entirely.

sequent pooling layer. For example, in the case of 5-month max pooling, we have $\{h_1^{(p)}, h_2^{(p)}\} = \{\max(h_1^{(c)}, h_2^{(c)}, h_3^{(c)}, h_4^{(c)}, h_5^{(c)}), \max(h_6^{(c)}, h_7^{(c)}, h_8^{(c)}, h_9^{(c)}, h_{10}^{(c)})\}$. Likewise, for 5-month average pooling, we have$\{h_1^{(p)}, h_2^{(p)}\} = \{5^{-1}\sum_{j=1}^{5} h_j^{(c)}, 5^{-1}\sum_{j=6}^{10} h_j^{(c)}\}$. The two output values of the pooling layer are combined into the observed target variable in the subsequent fully-connected layer by $y_{13} = \sum_{j=1}^{2} \theta_j h_j^{(p)} + b_0 + \varepsilon_{13}$. When training CNNs, parameters are estimated by back propagation as in MLP using observed values from $t = 12$ to $t = T$. As described above, convolution using only current and past values in terms of the objective variable is called causal convolution. In this CNN, the total number of parameters to be estimated is $3 + 1 + 2 + 1 = 7$, because the parameters are the weight of one filter w and the bias term b in the convolution layer, the weight of all coupling layers $\{\theta_j\}_{j=1}^{2}$ and the bias term b_0 (There are no parameters to be estimated in the pooling layer). This is a significant reduction compared to the number of parameters in the MLP when $q = 3$, for example, which is $3 \times (12 + 1) + 3 + 1 = 43$.

As a third neural network, we consider the recurrent neural network (RNN). Using a simple RNN, the forecast model can be given by

$$y_{t+h} = \sum_{j=1}^{q} \theta_j h_{jt} + b + \varepsilon_{t+h} \tag{19}$$

$$h_{jt} = \sigma(w_j' X_t + \sum_{k=1}^{q} \theta_{jk} h_{kt-1} + b_j) \tag{20}$$

where $\{h_{jt-1}\}_{j=1}^{q}$ represents the q hidden units of the middle layer in period t and σ is the activation function. The time-series structure is introduced into the RNN by the adding past information $\{h_{kt-1}\}_{k=1}^{q}$ along with the usual $N \times 1$ predictor variable X_t as input to the intermediate layer (20). Hochreiter and Schmidhuber (1997) introduced a gating mechanism called long short-term memory (LSTM) to RNNs. The state of the LSTM is represented by an intermediate layer h_t and a memory cell C_t, and the information flow is controlled by each gate. For simplicity, consider the case of one-dimensional intermediate

layer so that the input gate, forget gate, and output gate can be written as

$$i_t = \sigma_g(w_i'(X_t', h_{t-1})' + b_i) \tag{21}$$

$$f_t = \sigma_g(w_f'(X_t', h_{t-1})' + b_f) \tag{22}$$

$$o_t = \sigma_g(w_o'(X_t', h_{t-1})' + b_o) \tag{23}$$

where h_t is the (scalar) hidden unit of the intermediate layer in period t, b_i, b_f, and b_o are bias terms, w_i, w_f, and w_o are $(N+1)\times 1$ vectors of weights, σ_g is the gate activation function, usually a sigmoid function. If we also assume one-dimensional memory cell, the current C_t is given by

$$C_t = f_t \times C_{t-1} + i_t \times \sigma(w_c'(X_t', h_{t-1}) + b_c) \tag{24}$$

and the current h_t is given by

$$h_t = o_t \times \sigma(C_t). \tag{25}$$

The forecast model is given by $y_{t+h} = \theta h_t + b + \varepsilon_{t+h}$ where b_c and b are bias terms, w_c is a $(N+1)\times 1$ vector of weights, θ is a scalar weight, and σ is an activation function, usually the tanh function. In (24), how much new information $\sigma(w_c'(X_t', h_{t-1})' + b_c)$ is added to the storage cell C_t is controlled by the input gate i_t and how much past information C_{t-1} is left is controlled by the forget gate f_t. The output (25) expression of the intermediate layer is also controlled by the output gate o_t. Intuitively, by keeping the values of these gates within appropriate ranges, the model can avoid the vanishing gradient problem.

2.3 Dynamic factor models

The performance of nowcasting and forecasting based on machine learning methods is compared with those based on DFM. Following Mariano and Murasawa (2003), Giannone, Reichlin and Small (2008), Bańbura and Modugno (2014) and Luciani et al. (2018) among others, we estimate DFM by the maximum likelihood method allowing for mixed frequency data. To review this model, for simplicity, assume that the two series of monthly

data $\{y^*_{t_m}, x_{t_m}\}$, with a new monthly time index $t_m = 1, ...T_m$, are described by a DFM given by

$$\begin{bmatrix} y^*_{t_m} \\ x_{t_m} \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} f_{t_m} + \begin{bmatrix} e_{1t_m} \\ e_{2t_m} \end{bmatrix}$$

where λ_i, for $i = 1, 2$, is a factor loading, f_{t_m} is a scalar common factor which follows an AR model

$$f_{t_m} = \sum_{j=1}^{p_f} \phi_j f_{t_m - j} + \varepsilon_{t_m} \tag{26}$$

and e_{it_m}, for $i = 1, 2$, is an idiosyncratic error term which follows an AR model

$$e_{it_m} = \sum_{j=1}^{p_i} \rho_j e_{it_m - j} + \varepsilon_{it_m}. \tag{27}$$

Let us now assume that x_{t_m} is observed monthly, but only

$$y_{t_m} = y^*_{t_m} + y^*_{t_m - 1} + y^*_{t_m - 2} \tag{28}$$

is observed for each end of the quarter. That is, y_{t_m} is observed only when $t_m = 3t$, but is a missing observation when $t_m = 3t - 1$ and $t_m = 3t - 2$. Therefore, for $t_m = 3t$, the measurement equation is given by

$$\begin{bmatrix} x_{t_m} \\ y_{t_m} \end{bmatrix} = \begin{bmatrix} x_{t_m} \\ y^*_{t_m} + y^*_{t_m - 1} + y^*_{t_m - 2} \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \lambda_2 & \lambda_2 & \lambda_2 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_{t_m} \\ f_{t_m - 1} \\ f_{t_m - 2} \\ e_{1t_m} \\ e_{2t_m} \\ e_{2t_m - 1} \\ e_{2t_m - 2} \end{bmatrix}$$

In our analysis, we use both monthly hard data and news data for x_{t_m} and forecast y_{t_m} using year-on-year growth rate transformation instead of (28).

3 Data and procedures

3.1 Predictors

3.1.1 Hard data: macroeconomic indicators

We use a subset of the data adopted in the New Indices of Business Conditions, which the Cabinet Office claims better captures gross product. In the dataset, we exclude nominal series as well as construction statistics, which have been subject to overestimation and considered unreliable by some economists.

As a set of predictors from hard data, we use 10 monthly macroeconomic indicators and lagged quarterly GDP growth rate (Table 1). These predictors are a subset of the data adopted in the New Indices of Business Conditions, which the Cabinet Office claims better captures gross product. In the dataset, we exclude nominal series as well as construction statistics, which have been subject to overestimation and considered unreliable by some economists. For each variable, the delay of release in terms of the number of days is shown in the "Delay" column of Table 1.

3.1.2 Soft data: survey-based indicators

Previous studies, such as Bragoli (2017) and Hayashi and Tachi (2023), have demonstrated that survey-based ('soft') indicators are valuable alongside standard macroeconomic ('hard') indicators in a Dynamic Factor Model (DFM) framework. Consequently, we incorporate survey-based indicators in our nowcasting exercise. Table 1 lists all survey-based indicators used in this study. Unlike Hayashi and Tachi (2023), who used the Tankan Survey compiled by the Bank of Japan, we employ Reuters' Tankan Survey. This survey is designed to capture business sentiment and provides similar information to the Bank of Japan version. We prefer the Reuters survey because it offers monthly data, whereas the Bank of Japan version is available only quarterly.

3.1.3 News data

News data from January 1992 to December 2022 are extracted from the business section in Mainichi Shimbun. Here, one document corresponds to a single article and our corpus consists of all articles. The total number of articles is approximately 300 thousands. On average, there are 1,200 articles available a month. We construct two types of text-based indicators. The release delays for both types of indicators are set to 1.

Text metrics based on term frequency

The first approach is based on term frequency. We first pick up the most frequent **1000** nouns from articles up to December 2016 as the vocabulary to calculate term frequency (tf) for each article. The tf of a term w_j in a document d_i, adjusted for document length, is defined as

$$tf(w_j, d_i) = \frac{\text{the number of } w_j \text{ appears in document } d_i}{\text{total number of words in document } d_i}$$

By calculating tf for each article (document), and then averaging the tf for all articles in each month yields a monthly observation of tf series for a particular term. Based on these tf series, we then calculate the term frequency inverse document frequency (tf-idf) defined as follows:

$$tf\text{-}idf(w_j, d_i)_t = tf(w_j, d_i) \cdot idf(w_j, D_t), \text{ where } D_t \text{ is the corpus , and}$$
$$idf(w_j, D_t) = log\frac{\text{the number of documents in} D_t}{\text{the number of documents that contain } w_j \text{ in} D_t} + 1$$

The inverse document frequency $idf(w_j, D_t)$ is a measure of how much information the term w_j provides, and it increases as w_j becomes *rare* in D_t.

Since the target of our nowcasting exercise is GDP, it is preferable that the inverse document frequency captures information relevant to current business cycles. We explicitly make D_t dependent on time t, that is, D_t contains all documents from 50 months (about 1 business cycle) before t. Also, the rolling scheme ensures us to avoid data leakage problem. In other words, since we only use information from articles up to the day we make the prediction, no future information about the target variable is utilized.

When the text metrics are fed into the models described below, we select top n principal components such that the selected components account for 50% of the total variance. This amounts to selecting $n = 15$ components in our analysis. We will refer to these principal components simply as **T0** in the following analysis.

Univariate dictionary-based text metric

The second approach to transforming preprocessed text into quantitative time series involves methods that establish a fixed relationship between input and output, without any learning component. We refer to this approach as dictionary methods. Dictionary methods assign specific scores (positive or negative) to particular terms and calculate the net score per month. The dictionaries we use are based on the works of Takamura, Inui, and Okumura (2006); Higashiyama, Inui, and Matsumoto (2008); Ito et al. (2018); and Goshima, Shintani, and Takamura (2022).

The two dictionaries by Takamura, Inui, and Okumura (2006) and Higashiyama, Inui, and Matsumoto (2008) aim to extract sentiment in a general context. On the other hand, the dictionaries provided by Goshima, Shintani, and Takamura (2022) and Ito et al. (2018) are domain-specific. The former is designed to extract sentiment about macroeconomic developments, while the latter focuses on financial documents to measure market sentiment. For the sake of brevity, we refer to the text metrics generated by these four dictionaries as **T1, T2, T3** and **T4**, respectively.

In addition, we use the method developed by Baker, Bloom, and Davis (2016) to construct the economic policy uncertainty (EPU) index. We will refer to the EPU index simply as **T5** in the following analysis.

3.2 Out-of-sample forecast evaluation

3.2.1 Models

The forecasts are constructed using one of seven machine learning methods, which are applied directly to estimate the seven U-MIDAS models. The seven machine learning methods are Elastic Net, Support Vector Regression (SVR), Random Forest, Boosted

Trees (implemented through LightGBM), Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN with LSTM). The DFM forecasts are constructed based on the specification $(p_f, p_i) = (2, 1)$, which is standard in the literature and also applied in Hayashi and Tachi (2023).

Additionally, we conduct a forecast combination of DFM and ML, either by a simple arithmetic mean of the predictions produced by DFM and one out of 7 ML methods or by DFM and several MLs at the same time.

We evaluate the simulated out-of-sample forecast performance of 7 machine learning methods and one DFM. The hyperparameters for the machine learning models are re-tuned at each forecast point using 5-fold cross-validation. In this process, we employ an expanding window approach, progressively increasing the training data size while maintaining a validation data size of one.

For example, at period $t = R$, we construct the forecast $\widehat{y}_{R+h}$ of a target variable y_{R+h} using the information only up to $t = R$ and evaluate the forecast error $y_{R+h} - \widehat{y}_{R+h}$. For the next period $t = R + 1$, the model is re-estimated using the data up to $t = R + 1$ and forecast value $\widehat{y}_{R+h+1}$ is constructed. Therefore, the hyperparameters may be different depending on the point of forecast, even if the model specification is unchanged. For a benchmark, we also estimate an AR(1) model and update the coefficient in the same manner.

3.2.2 Evaluation scheme

We use a recursive scheme in our simulated out-of-sample nowcast exercises. Recall that we use observations available at February 1, May 1, August 1, and November 1 in each year to estimate the M1 type model. Similarly, observations at March 1, June 1, September 1, and December 1 are used for the M2 type model, while observations at January 1, April 1, July 1, and October 1 are used for the M3 type model.

In the recursive scheme, we start to produce pseudo real-time out-of-sample prediction at January 1, 2018 using information after January 1, 2003. The sample size increases as we proceed to the nowcast point. For example, to construct a nowcast ($h = 0$) for

2018Q4 based on the M3 type model, the estimation period is from January 1, 2003 to January 1, 2019 in the recursive scheme. The next nowcast for 2019Q1 based on the M3 type model is then computed by estimating the same model using data from January 1, 2003 to April 1, 2019. This procedure is repeated until the nowcast for 2022Q4 based on the observations from January 1, 2003, to October 1, 2022, is constructed.

Nowcasts based on the M1 and M2 type models can be constructed in a similar manner. For example, to construct a nowcast for 2019Q1 based on the M1 type model, the estimation period is from January 1, 2003 to November 1, 2018. The next forecast for 2019Q2 based on the M1 type model is then computed by estimating the same model using data from January 1, 2003 to February 1, 2019. This procedure is repeated until the forecast for 2022Q4 based on the observations from January 1, 2003 to November 1, 2022.

As a measure of forecast performance, we focus on the root mean square forecast errors (RMSEs) defined by the square root of $P^{-1}\sum_{t=R}^{T-h}(y_{t+h}-\widehat{y}_{t+h})^2$, where $\widehat{y}_{t+h}$ is the forecast value for horizon h by a forecast model, R is the initial sample size in estimating the model, and $P(=T-R)$ is the number of forecasts. If the RMSE for model 1 is smaller than the RMSE of model 2, we view that the former outperforms the latter in the out-of-sample forecast.

3.3 Strategy for putting text metrics into the model

One goal of our study is to measure the benefit of including text information in addition to 'hard' data or both 'hard' and 'soft' data in nowcasting models. However, we have no ex-ante information about which combination of text metrics performs best. Therefore, we explore all plausible patterns of text metric predictors.

In particular, when making a single prediction, the variables we use always include *all* of the 'hard' variables (category 'Hard' in Table 1), none or *all* of the 'soft' variables (category 'Soft' in Table 1), none or *one* of the text metrics based on a general sentiment dictionary (T0, T1, and T2 in Table 1), and none or *one* of the text metrics based on a domain-specific dictionary (T3, T4, and T5 in Table 1). Although it is common to

include 'soft' data in the existing literature using DFM, its usefulness in ML settings is less clear, and thus we examine it explicitly in this paper.

Therefore, given a model, such as Elastic Net, we consider $2 \times 4 \times 4 = 32$ different patterns of regressors and estimate 32 variants of Elastic Net. This process entails estimating seven ML models and one DFM, resulting in the estimation of $32 \times 8 = 256$ models for each prediction. In this paper, we select at most one variable from each category as described above because variables from the same category are likely to share similar information. Moreover, limiting the number of variables reduces the possible combinations of variables, thus significantly reducing computation time.

4 Results

Let us now evaluate the performance of all the models considered in terms of the Root Mean Squared Errors (RMSEs) of the nowcast ($h = 0$). Table 3 and Table 2 present the best-performing triple of input, model, and the resulting RMSE for the forecast horizon and forecast period, with the input fixed at 'hard' and 'soft' in Table 3, and without restrictions on the input in Table 2. For example, the first row in the Jan17-Dec19 column in Table 3 shows that for the M1 type nowcasting, the RMSE of mlp is 0.743, and this is the lowest among 18 combinations[3] when we consider only 'hard' and 'soft' data as input.

We divided the out-of-sample forecasting period into two periods. The first period Jan17-Dec19 corresponds to "normal" time in the sense that nowcasting exercises using DFM with 'hard' and 'soft' data work well in the literature. The second Jan20-Dec22 period corresponds to the period when the COVID-19 pandemic was at its most intense, during which the accuracy of traditional forecasting exercises deteriorates significantly compared to the "normal" period.

We compare the results of Table 2 and Table 3 in Table 5, where the relative RMSE

[3] 18 models: ar1, dfm, enet, svr, rf, lgbm, mlp, cnn, lstm, dfm+enet, dfm+svr, dfm+rf, dfm+lgbm, dfm+mlp, dfm+cnn, dfm+lstm, mean4 (unweighted average of enet, rf, mlp, and dfm), mean8 (unweighted average of 1 dfm and 7 mls), where A + B denotes forecast combination by unweighted mean of model A and model B.

columns show the ratio of the RMSEs, and the statistic columns report the test statistic developed in Pitarakis (2023) for the null hypothesis that the predictions of the best model using **only H and S** have the same accuracy as the predictions of the best model when the input is unrestricted. The test introduced in Pitarakis (2023) accommodates the comparison of nested models in a recursive scheme and is thus suitable for our situation.

4.1 Benefits of news data in prediction

Table 5 shows that the incorporation of news text data consistently reduces prediction errors, and most of these improvements are statistically significant. This advantage is particularly evident during the COVID-19 period, where traditional methods experience a marked decline in accuracy. Furthermore, as shown in Table 2, the inclusion of multiple types of text data generally outperforms the inclusion of a single type of text data.

A similar comparison is conducted in Tables 6, 7, and 9, with models fixed for DFM. Table 9 demonstrates a reduction in prediction error when text data is included. However, only some instances (M2 Jan20-22; M2 Overall) are statistically significant. In particular, the benefits of incorporating text data are particularly pronounced during the COVID-19 period.

Interestingly, Tables 2 and 6 also reveal that when text metrics generated by domain-specific dictionaries (T3, T4, and T5) are included in the model, the most frequently selected metrics are either T3, which is designed to extract sentiment about macroeconomic developments, or T5, which aims to capture economic policy uncertainty. This suggests that text metrics focused on gauging macroeconomic developments contain valuable information to predict GDP.

We also report the best-performing model among the 18 models using only 'hard' data and the performance of DFM using only 'hard' data in Table 4 and 8, respectively. Comparing these with their 'hard' and 'soft' counterparts in Table 3 and 7 reveals that incorporating 'soft' data not only supports previous findings that the combination of 'hard' and 'soft' data enhances DFM performance but also improves nowcasting accuracy in the ML setting.

To evaluate the benefit of adding news text data in specific models, we report the RMSEs in Tables 10, 11 and 12, representing the "normal", "COVID-19", and the entire periods, respectively. These tables show the RMSEs of the best-performing input combinations and their relative RMSEs compared to the baseline inputs using only 'hard' and 'soft' data. For example, during the period from January 2017 to December 2019, among M1-type models, the DFM with the input combination $(H, S, T0, T4)$ exhibited the best performance, with an RMSE of 0.89 and a ratio of 0.93 compared to the (H, S) counterpart. However, this reduction in RMSE was not statistically significant according to Pitarakis (2023).

Again, we observe a consistent reduction in RMSEs across models, particularly notable in ML models. Interestingly, the magnitude and significance of the reduction in RMSE are pronounced for models that accommodate a high degree of non-linearity, such as MLP, LSTM, and CNN. Furthermore, comparing Tables 10 and 11, the reduction in RMSE for models such as Elastic Net and SVR starts to achieve statistical significance during the COVID-19 period.

The trajectories of the MLP and DFM for two target quarters, 2019Q2 and 2020Q2, representing "normal" times and the sharpest GDP decline caused by the pandemic, respectively, are shown in Figures 3 and 4. Focusing on the target during the COVID-19 period, both predictions generated using 'hard' and 'soft' data (gray dots) get closer to the actual value (red dot) as the prediction horizon moves from the M1-type forecast to the M3-type nowcast as new information becomes available. However, when news data are available (gray dots), the predictions approach the target much more quickly for both DFM and MLP.

4.2 Benefits of combining DFM with ML

As shown in Table 2, many of the best-performing models involve the forecast combination of DFM and ML. Table 13 presents a comparison between the forecast combination of DFM and ML methods versus DFM alone. Across all forecast horizons (M1, M2, M3) and time periods, the combination of DFM and ML consistently outperforms DFM alone,

as indicated by relative RMSEs below 1. This improvement is particularly significant for longer-horizon nowcasts (M1 and M2 type) across all periods. Notably, the advantage of ML+DFM remains evident even during the COVID-19 pandemic period, especially for M1 type nowcasts, where we observe a statistically significant improvement (relative RMSE of 0.86, significant at the 1% level).

These findings highlight the potential of integrating traditional econometric methods like DFM with flexible ML approaches. Their complementary strengths appear especially valuable during economic instability, such as the COVID-19 pandemic, where they may better capture complex, rapidly changing data relationships. However, the improvement varies across forecast horizons and periods, suggesting that the advantages of this combined approach may depend on specific economic conditions and the forecasting task at hand.

5 Conclusion

In this paper, we conduct a nowcasting analysis of Japan's GDP using machine learning. We use the machine learning approach because the estimation of the unrestricted mixed-data sampling (MIDAS) models in our setting involves high-dimensional data. Based on the unrestricted MIDAS model with macroeconomic indicators, survey-based indicators, and text information extracted from news articles, we find that text information helps to enhance nowcasting performance, especially during the COVID-19 period. Furthermore, the combination of machine learning and dynamic factor models has the potential to outperform using either method alone.

The use of text data offers benefits beyond improving accuracy. The spread of COVID-19 caused significant economic fluctuations over short periods due to subsequent lockdowns. This economic turmoil has underscored the importance of timely macroeconomic assessments for economists. Traditional hard and survey data, on which economists heavily rely for economic analysis, take several weeks to be released. In contrast, text-based information can be utilized in real time. Therefore, incorporating news text data is a

valuable option when the accuracy of existing models is affected by large unexpected events, such as the COVID-19 pandemic shock.

References

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

[2] Bai, J. and Ng, S., 2009. Boosting diffusion indices. Journal of Applied Econometrics 24, 607-629.

[3] Bańbura M., Giannone D. and Reichlin L. 2011. Nowcasting. In Oxford Handbook on Economic Forecasting, Clements M.P., Hendry D.F. (eds). Oxford University Press, Oxford, 193–224.

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

[5] Bragoli, D., 2017. Now-casting the Japanese economy, International Journal of Forecasting, 33, 390–402.

[6] Breiman, L., 1996a. Stacked regressions. Machine Learning, 24, 49-64.

[7] Breiman, L., 1996b. Bagging predictors. Machine Learning 24, 123-140.

[8] Breiman, L., 2001. Random forest. Machine Learning 45(1), 5-32.

[9] Breiman, L., Friedman, J., Stone, C.J. and Olshen, R.A., 1984. Classification and Regression Trees. Chapman and Hall/CRC, New York, NY.

[10] Chen, J. C., Dunn, A., Hood, K., Driessen, A. and Batch, A., 2022. Off to the races: a comparison of machine learning and alternative data for predicting economic indicators. In Abraham, K.G., Jarmin, R.S., Moyer, B., Shapiro, M.D., editors, Big Data for 21st Century Economic Statistics. University of Chicago Press, Chicago, IL., 373-402.

[11] Chikamatsu, K., Hirakata, N., Kido, Y. and Otaka, K., 2021, Mixed-frequency approaches to nowcasting GDP: An application to Japan. Japan and the World Economy, 57, Article 101056.

[12] Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., and Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Conference on Empirical Methods in Natural Language Processing (EMNLP 2014).

[13] Clements, M. and Galvão, A., 2008. Macroeconomic forecasting with mixed-frequency data: forecasting output growth in the United States, Journal of Business & Economic Statistics 26, 546–554.

[14] Coulombe, P. G., Leroux, M., Stevanovic, D. and Surprenant, S., 2022. How is machine learning useful for macroeconomic forecasting? Journal of Applied Econometrics, 37(5), 920-964.

[15] Diebold, F. X. and Shin, M., 2018. Machine learning for regularized survey forecast combination: partially-egalitarian lasso and its derivatives. International Journal of Forecasting 35(4), 1679-1691.

[16] Ellingsen, J., Larsen, V. H. and Thorsrud, L. A., 2022. News media versus FRED-MD for macroeconomic forecasting. Journal of Applied Econometrics, 37, 63– 81.

[17] Foroni, C., Marcellino, M. G. and Schumacher, C., 2015. Unrestricted mixed data sampling (MIDAS): MIDAS regressions with unrestricted lag polynomials. Journal of the Royal Statistical Society Series A, vol. 178(1), 57-82.

[18] Freund, Y., 1995. Boosting a weak learning algorithm by majority. Information and Computation 121(2), 256-285.

[19] Ghysels, E., Santa-Clara, P. and Valkanov, R., 2005. There is a risk-return trade-off after all. Journal of Financial Economics, 76(3), 509-548.

[20] Ghysels, E. and Valkanov, R., 2006. Linear time series processes with mixed data sampling and MIDAS regression models, University of North Carolina, mimeo.

[21] Giannone, D., Lenza, M. and Primiceri, G. E. 2021. Economic predictions with big data: the illusion of sparsity. Econometrica, 89, 2409-2437.

[22] Giannone, D., Reichlin, L. and Small, D. 2008. Nowcasting GDP and inflation: the real-time informational content of macroeconomic data releases, Journal of Monetary Economics 55, 665–676.

[23] Goshima, K., Ishijima, H., Shintani, M. and Yamamoto, H., 2021. Forecasting Japanese inflation with a news-based leading indicator of economic activities. Studies in Nonlinear Dynamics & Econometrics, 25(4), 111-133.

[24] Granger, C. W. J. and Ramanathan, R., 1984. Improved methods of combining forecasts. Journal of Forecasting 3(2), 197-204.

[25] Gu, S., Kelly, B. and Xiu, D., 2020. Empirical asset pricing via machine learning. Review of Financial Studies, 33(5), 2223-2273.

[26] Goshima, K., Shintani M., and Takamura H., 2022, Sentiment Dictionary for Business Cycle Analysis and its Applications, J-stage *Shizengengo Shori* 29(4), 1233-1253

[27] Hoerl, A. E. and Kennard, R. W., 1970. Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12(1), 55-67.

[28] Hayashi, F. and Tachi, Y., 2023. Nowcasting Japan's GDP. Empirical Economics 64, 1699–1735.

[29] Higashiyama, M., Inui K., and Matsumoto Y., 2008. Learning Sentiment of Nouns from Selectional Preferences of Verbs and Adjectives, Proceedings of the 14th Annual Meeting of the Association for Natural Language Processing, 584-587.

[30] Ito T., Sakaji H., Tsubouchi K., Izumi K., Yamashita T., 2018. Text-Visualizing Neural Network Model: Understanding Online Financial Textual Data. In: Phung

D., Tseng V., Webb G., Ho B., Ganji M., Rashidi L. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2018. Lecture Notes in Computer Science, Springer, vol 10939, 247-259.

[31] Kalamara, E., Turrell, A., Redl, C., Kapetanios, G. and Kapadia, S., 2022. Making text count: Economic forecasting using newspaper text. Journal of Applied Econometrics, 37(5), 896– 919.

[32] Kim, H. H. and Swanson, N. R., 2018. Mining big data using parsimonious factor, machine learning, variable selection and shrinkage methods. International Journal of Forecasting 34(2), 339-354.

[33] Luciani, M., Pundit, M., Ramayandi, A. and Veronese, G., 2018. Nowcasting Indonesia. Empirical Economics 55, 597-619.

[34] Maehashi, K. and Shintani, M., 2020. Macroeconomic forecasting using factor models and machine learning: an application to Japan, Journal of the Japanese and International Economies, 58, Article 101104.

[35] Mariano, R. S. and Murasawa, Y., 2003. A new coincident index of business cycles based on monthly and quarterly series. Journal of Applied Econometrics, 18(4), 427-443.

[36] Medeiros, M. C., Vasconcelos, G. F. R., Veiga, Á. and Zilberman, E., 2021. Forecasting inflation in a data-rich environment: the benefits of machine learning methods. Journal of Business and Economic Statistics, 39(1), 98-119.

[37] Pitarakis, J-Y., 2023. A novel approach to predictive accuracy testing in nested environments. Econometric Theory, First View, 1-44.

[38] Schapire, R. E., 1990. The strength of weak learnability. Machine Learning 5(2), 197-227.

[39] Tibshirani, R., 1996. Regression shrinkage and selection via lasso. Journal of the Royal Statistical Society: Series B 58(1), 267-288.

[40] Wolpert, D.H., 1992. Stacked generalization, Neural Networks, 5(2), 241-259.

[41] Zou, H. and Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B 67(2), 301-320.

Table 1. Data description and data treatment

No.	Variable	Category	Frequency	Delay	Transform
1	IIP (final demand goods)	Hard	M	29	$\Delta \log$
2	IIP (final producer goods)	Hard	M	29	$\Delta \log$
3	ITA (broad ranging personal services)	Hard	M	45	$\Delta \log$
4	ITA (broad ranging essential personal services)	Hard	M	45	$\Delta \log$
5	ITA (broad ranging non essential personal services)	Hard	M	45	$\Delta \log$
6	ITA (broad ranging business services)	Hard	M	45	$\Delta \log$
7	Real Retail Sales	Hard	M	46	$\Delta \log$
8	Real Services Exports	Hard	M	39	$\Delta \log$
9	Capital Goods Supply	Hard	M	35	$\Delta \log$
10	Real Exports	Hard	M	21	$\Delta \log$
11	**Gross Domestic Product** (target)	Hard	Q	46	$\Delta \log$
12	Economy Watchers' Survey	Soft	M	9	Δ
13	Local Business Outlook Survey	Soft	M	-1	Δ
14	Consumer Confidence Survey	Soft	M	8	Δ
15	Reuters Tankan DI (non-manufacturers)	Soft	M	-15	Δ
16	Reuters Tankan DI (manufacturers)	Soft	M	-15	Δ
17	**T0**: tf-idf extracted from the business section in Mainichi Shimbun.	Text (general)	M	1	none
18	**T1**: Text metrics based on Takamura, Inui, and Okumura (2006)	Text (general)	M	1	none
19	**T2**: Text metrics based on Higashiyama, Inui, and Matsumoto (2008)	Text (general)	M	1	none
20	**T3**: Text metrics based on Goshima, Shintani, and Takamura (2022)	Text (domain specific)	M	1	none
21	**T4**: Text metrics based on Ito, et al. (2018)	Text (domain specific)	M	1	none
22	**T5**: Economic Policy Uncertainty Index based on Baker, Bloom, and Davis (2016)	Text (domain specific)	M	1	none

Note: IIP: Index Industrial Production; ITA: Index Tertiary Activity.
Frequency shows whether a variable is monthly (M) or quarterly (Q). Delay shows the release delay expressed in number of days. This can take negative values if the release date is prior to the end of the reference period. All indicators are seasonally adjusted either by the source or by the authors.
Following the suggestion of Vermeulen (2012), we apply logistic transformation $100 \log \frac{U-x}{x-L}$ to diffusion indexes (No. 12 - 16), where $(U, L) = (100, 0)$ for No. 12 and No. 14; $(U, L) = (100, -100)$ for No. 13, No. 15, and No. 16.

Table 2. Best models

	Jan17-Dec19			Jan20-Dec22			Overall		
	Input	Model	RMSE	Input	Model	RMSE	Input	Model	RMSE
M1	H, S, T2	lstm	0.556	H, S, T1, T5	cnn+dfm	1.687	H, S, T1, T5	cnn+dfm	1.300
M2	H, S, T0, T5	cnn+dfm	0.486	H, S, T2, T3	svr+dfm	0.724	H, S, T0, T5	cnn+dfm	0.705
M3	H, S	svr+dfm	0.349	H, S, T1, T5	cnn+dfm	0.685	H, S, T1, T5	cnn+dfm	0.594

Note: Hereafter, plus appears in the column Model denotes forecast combination by unweighted mean of the two models.

Table 3. Best models when inputs fixed at Hard and Soft indicators

	Jan17-Dec19			Jan20-Dec22			Overall		
	Input	Model	RMSE	Input	Model	RMSE	Input	Model	RMSE
M1	H, S	mlp	0.743	H, S	enet+dfm	1.780	H, S	enet+dfm	1.417
M2	H, S	dfm	0.771	H, S	cnn+dfm	0.952	H, S	cnn+dfm	0.902
M3	H, S	svr+dfm	0.349	H, S	dfm	0.797	H, S	dfm	0.620

Table 4. Best models when inputs fixed at Hard indicators

	Jan17-Dec19			Jan20-Dec22			Overall		
	Input	Model	RMSE	Input	Model	RMSE	Input	Model	RMSE
M1	H	cnn	0.765	H	lstm	2.697	H	lstm	1.983
M2	H	cnn	0.724	H	dfm	1.305	H	dfm	1.115
M3	H	svr+dfm	0.349	H	mean_4	0.959	H	mean_4	0.734

Note: mean_4 denotes forecast combination by unweighted mean of enet, rf, mlp, and dfm.

Table 5. Relative accuracy of hard/soft/news inputs to hard/soft inputs

	Jan17-Dec19		Jan20-Dec22		Overall	
	relative RMSE	statistic	relative RMSE	statistic	relative RMSE	statistic
M1	0.75	3.72***	0.95	1.06	0.92	1.68**
M2	0.63	14.27***	0.76	8.38***	0.78	8.90***
M3	1.00	0.30	0.86	1.47*	0.96	-1.92

Note: This table compares the results from Table 2 and Table 3. The relative RMSE columns show the ratio of RMSEs, and the statistic columns report the test statistic $\overline{\mathcal{S}}_T(\tau_0, \lambda_2^0)$ in Pitarakis (2023) of the null that the predictions of the best model using **only H and S** have better accuracy compared to the predictions of the best model when the input is unrestricted. We set $(\tau_0 = 0.8, \lambda_2^0 = 1)$ in line with the guideline provided in Pitarakis (2023).
*10% significance level. **5% significance level. ***1% significance level.

Table 6. Best models for Dynamic Factor Model

	Jan17-Dec19			Jan20-Dec22			Overall		
	Input	Model	RMSE	Input	Model	RMSE	Input	Model	RMSE
M1	H, S, T0, T4	dfm	0.890	H, S, T0, T5	dfm	1.967	H, S, T0, T5	dfm	1.534
M2	H, S, T3	dfm	0.762	H, S, T0, T5	dfm	0.755	H, S, T0, T5	dfm	0.766
M3	H, S, T0	dfm	0.364	H, S, T1	dfm	0.791	H, S, T1	dfm	0.617

Table 7. Best models for Dynamic Factor Model when inputs fixed at Hard and Soft indicators

	Jan17-Dec19			Jan20-Dec22			Overall		
	Input	Model	RMSE	Input	Model	RMSE	Input	Model	RMSE
M1	H, S	dfm	0.958	H, S	dfm	1.986	H, S	dfm	1.559
M2	H, S	dfm	0.771	H, S	dfm	1.018	H, S	dfm	0.903
M3	H, S	dfm	0.368	H, S	dfm	0.797	H, S	dfm	0.620

Table 8. Best models for Dynamic Factor Model when inputs fixed at Hard indicators

	Jan17-Dec19			Jan20-Dec22			Overall		
	Input	Model	RMSE	Input	Model	RMSE	Input	Model	RMSE
M1	H	dfm	0.965	H	dfm	2.794	H	dfm	2.090
M2	H	dfm	0.886	H	dfm	1.305	H	dfm	1.115
M3	H	dfm	0.363	H	dfm	1.041	H	dfm	0.779

Table 9. Relative accuracy of hard/soft/news inputs to hard/soft inputs for DFM

	Jan17-Dec19		Jan20-Dec22		Overall	
	relative RMSE	statistic	relative RMSE	statistic	relative RMSE	statistic
M1	0.93	-1.58	0.99	0.48	0.98	0.78
M2	0.99	-1.63	0.74	8.91***	0.85	5.64***
M3	0.99	-0.01	0.99	-1.32	0.99	-1.94

Note: This table compares the results from Table 6 and Table 7. The relative RMSE columns show the ratio of RMSEs, and the statistic columns report the test statistic $\overline{\mathcal{S}}_T(\tau_0, \lambda_2^0)$ in Pitarakis (2023) of the null that the predictions of the best model using **only H and S** have the same accuracy as the predictions of the best model when the input is unrestricted. We set $(\tau_0 = 0.8, \lambda_2^0 = 1)$ in line with the guideline provided in Pitarakis (2023).

Table 10. Best input combination (versus hard/soft) during Jan17-Dec19

	Input	Model	RMSE	relative RMSE	statistic
M1	H, S, T0, T4	dfm	0.89	0.93	-1.58
M2	H, S, T3	dfm	0.76	0.99	-1.63
M3	H, S, T0	dfm	0.36	0.99	-0.01
M1	H, S, T0, T3	enet	0.89	0.97	-1.68
M2	H, S, T0	enet	0.91	1.00	-1.89
M3	H, S, T0, T5	enet	0.43	0.97	0.86
M1	H, S, T2	svr	0.94	0.99	-1.91
M2	H, S, T3	svr	0.94	1.00	-1.95
M3	H, S	svr	0.37	1.00	0.56
M1	H, S, T0, T3	rf	0.85	0.89	-1.27
M2	H, S, T2	rf	0.86	0.95	-1.74
M3	H, S, T1, T4	rf	0.42	0.84	5.84***
M1	H, S, T0	lgbm	0.95	0.87	-0.42
M2	H, S, T0	lgbm	0.84	0.67	7.51***
M3	H, S, T1, T5	lgbm	0.59	0.83	9.35***
M1	H, S, T0	mlp	0.58	0.77	3.06***
M2	H, S, T2, T4	mlp	0.64	0.64	7.17***
M3	H, S, T5	mlp	0.53	0.80	9.04***
M1	H, S, T2	lstm	0.56	0.72	6.07***
M2	H, S, T0, T3	lstm	0.72	0.83	0.11
M3	H, S, T2	lstm	0.44	0.72	6.36***
M1	H, S, T3	cnn	0.59	0.68	18.59***
M2	H, S, T0, T3	cnn	0.66	0.67	12.84***
M3	H, S, T5	cnn	0.57	0.79	8.73***

Note: The relative RMSE columns show the ratio of RMSE to the H, S counterpart

Table 11. Best input combination (versus hard/soft) during Jan20-Dec22

	Input	Model	RMSE	relative RMSE	statistic
M1	H, S, T0, T5	dfm	1.97	0.99	0.48
M2	H, S, T0, T5	dfm	0.75	0.74	8.91***
M3	H, S, T1	dfm	0.79	0.99	-1.32
M1	H, S, T0, T3	enet	2.01	0.83	3.08***
M2	H, S, T0	enet	1.25	0.94	2.27**
M3	H, S, T1, T4	enet	1.17	0.93	1.82**
M1	H, S, T0, T4	svr	2.29	0.72	6.33***
M2	H, S, T4	svr	0.95	0.37	74.93***
M3	H, S, T5	svr	1.18	0.76	6.26***
M1	H, S, T2, T3	rf	2.97	0.96	0.73
M2	H, S, T2	rf	2.44	0.99	0.35
M3	H, S	rf	2.34	1.00	0.26
M1	H, S, T0, T5	lgbm	2.99	0.94	1.42*
M2	H, S, T1, T4	lgbm	2.46	0.96	1.17
M3	H, S, T3	lgbm	2.55	0.93	2.17**
M1	H, S, T0, T3	mlp	2.51	0.92	2.72***
M2	H, S, T1, T3	mlp	0.99	0.50	64.34***
M3	H, S, T0, T3	mlp	1.06	0.69	17.81***
M1	H, S, T5	lstm	2.42	0.86	2.25**
M2	H, S, T0, T3	lstm	2.21	0.87	1.91**
M3	H, S, T1, T4	lstm	1.88	0.91	1.55*
M1	H, S, T1, T5	cnn	1.74	0.90	2.24**
M2	H, S, T1, T3	cnn	1.16	0.98	1.38*
M3	H, S, T1, T3	cnn	1.07	0.79	12.91***

Note: The relative RMSE columns show the ratio of RMSE to the H, S counterpart

Table 12. Best input combination (versus hard/soft) for the entire sample (Jan17-Dec22)

	Input	Model	RMSE	relative RMSE	statistic
M1	H, S, T0, T5	dfm	1.53	0.98	0.78
M2	H, S, T0, T5	dfm	0.77	0.85	5.64***
M3	H, S, T1	dfm	0.62	0.99	-1.94
M1	H, S, T0, T3	enet	1.56	0.85	2.68***
M2	H, S, T0	enet	1.10	0.96	1.37*
M3	H, S, T1, T4	enet	0.88	0.94	0.91
M1	H, S, T0, T4	svr	1.75	0.75	5.44***
M2	H, S, T4	svr	0.95	0.49	33.92***
M3	H, S, T5	svr	0.88	0.78	6.17***
M1	H, S, T2, T3	rf	2.21	0.97	0.9
M2	H, S, T2	rf	1.83	0.99	0.61
M3	H, S	rf	1.69	1.00	0.36
M1	H, S, T0	lgbm	2.23	0.94	1.55*
M2	H, S, T1, T4	lgbm	1.92	0.95	1.56*
M3	H, S, T3	lgbm	1.86	0.93	0.31
M1	H, S, T0, T3	mlp	1.86	0.93	2.25**
M2	H, S, T1, T3	mlp	1.04	0.66	18.68***
M3	H, S, T0, T3	mlp	0.89	0.76	11.09***
M1	H, S, T5	lstm	1.78	0.87	2.44***
M2	H, S, T0, T3	lstm	1.64	0.87	2.28**
M3	H, S, T1, T4	lstm	1.38	0.90	1.83**
M1	H, S, T1, T5	cnn	1.32	0.89	3.08***
M2	H, S, T1, T3	cnn	1.01	0.93	3.56***
M3	H, S, T1, T3	cnn	0.89	0.82	8.38***

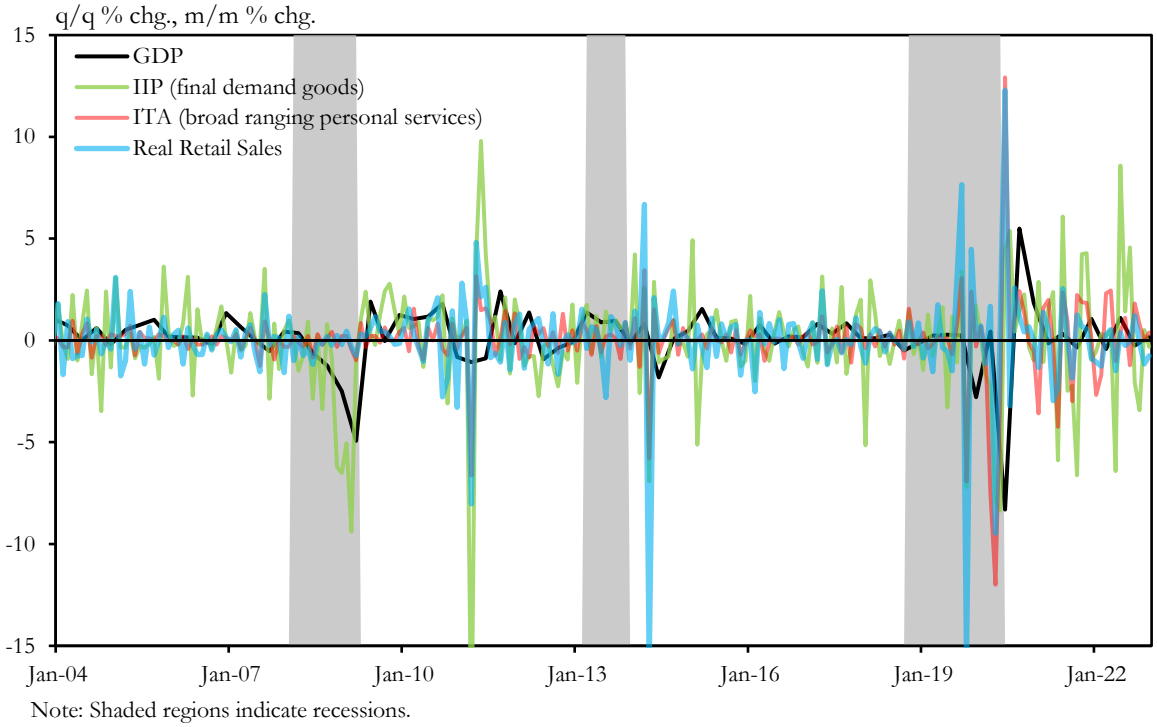
Note: The relative RMSE columns show the ratio of RMSE to the H, S counterpart

Table 13. Relative accuracy of forecast combination of DFM and ML to DFM

	Jan17-Dec19		Jan20-Dec22		Overall	
	relative RMSE	statistic	relative RMSE	statistic	relative RMSE	statistic
M1	0.72	0.92	0.86	2.72***	0.85	3.36***
M2	0.64	12.01***	0.96	1.46*	0.92	2.87***
M3	0.98	0.52	0.86	1.10	0.92	-0.66

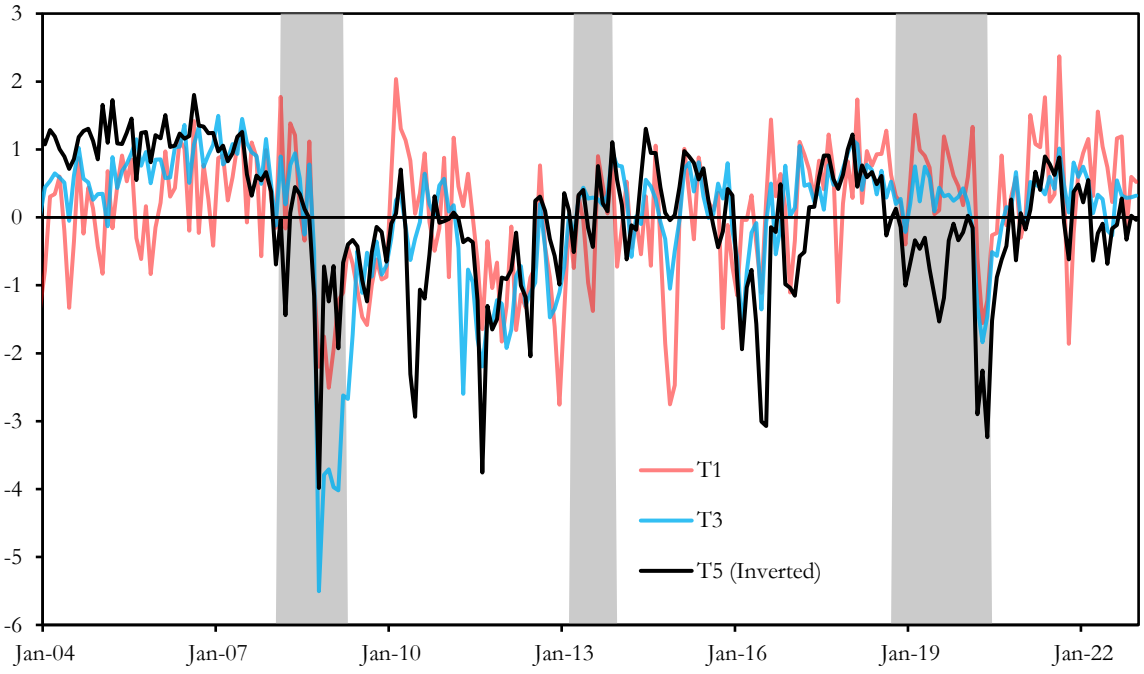
Note: This table compares the results of two model forms: ml+dfm and dfm. For each type (M1, M2, M3) and period combination, we compare the best model among all input patterns that takes the form of forecast combination with dfm, and the best model among all input patterns using dfm alone.
The relative RMSE columns show the ratio of RMSEs, and the statistic columns report the test statistic $\overline{\mathcal{S}}_T(\tau_0, \lambda_2^0)$ in Pitarakis (2023) of the null that the predictions of dfm have better accuracy compared to the ml+dfm. We set $(\tau_0 = 0.8, \lambda_2^0 = 1)$ in line with the guideline provided in Pitarakis (2023).
*10% significance level. **5% significance level. ***1% significance level.

Figure 1. GDP and selected hard data



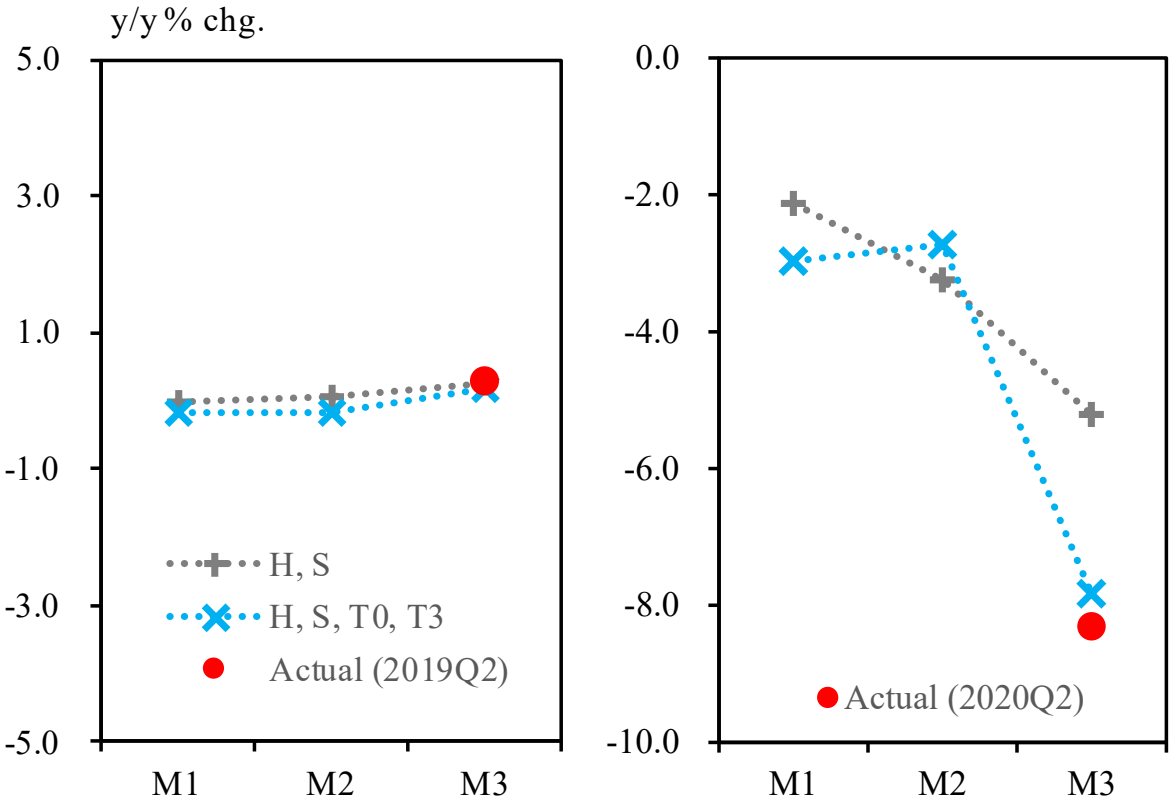
Note: Shaded regions indicate recessions.

Figure 2. Text-based indicators



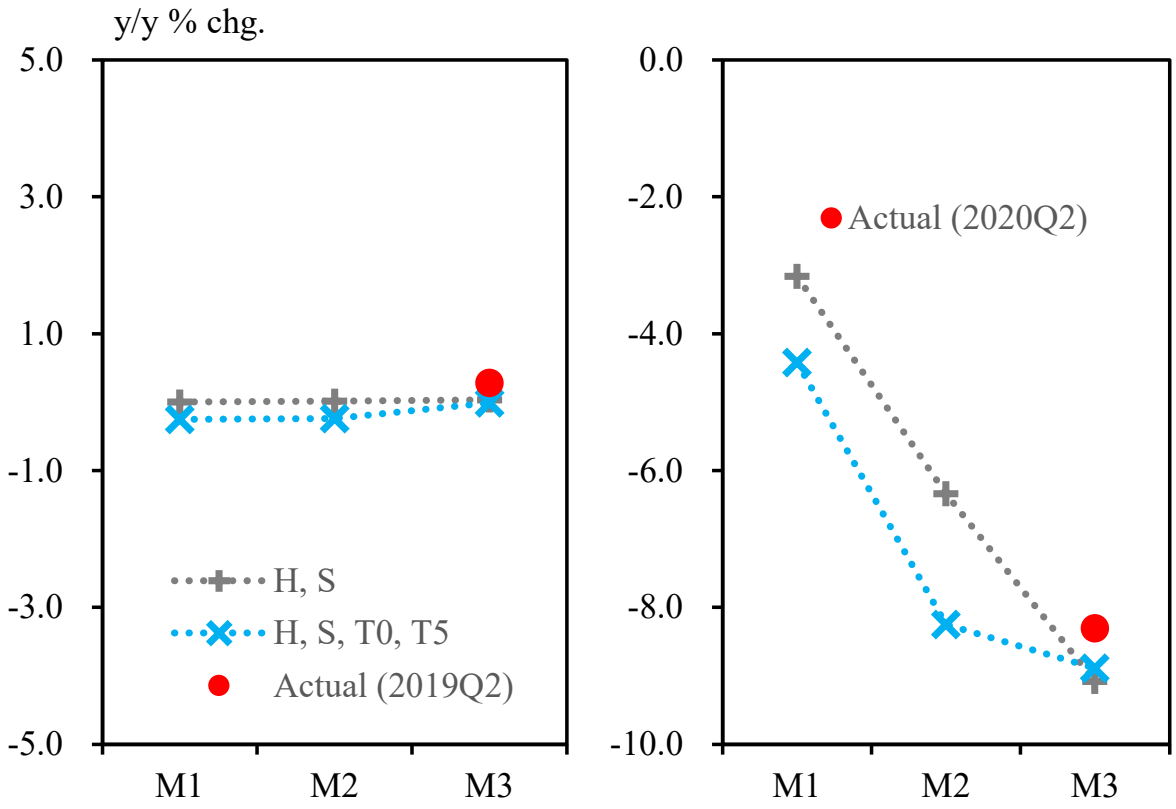
Note: All series are standardized for display purpose. Shaded regions indicate recessions.

Figure 3. Predictions of MLP by input



Note: This figure tracks the predictions produced by MLP with different inputs. Predictions are made at the first day of each month.

Figure 4. Predictions of DFM by input



Note: This figure tracks the predictions produced by DFM with different inputs. Predictions are made at the first day of each month.