MACROECONOMIC FORECASTING WITH LSTM AND MIXED FREQUENCY TIME SERIES DATA

Sarun Kamolthip
Faculty of Economics, Khon Kaen University
Khon Kaen 40002, Thailand
saruka@kku.ac.th

September 29, 2021

ABSTRACT

This paper demonstrates the potentials of the long short-term memory (LSTM) when applying with macroeconomic time series data sampled at different frequencies. We first present how the conventional LSTM model can be adapted to the time series observed at mixed frequencies when the same mismatch ratio is applied for all pairs of low-frequency output and higher-frequency variable. To generalize the LSTM to the case of multiple mismatch ratios, we adopt the unrestricted Mixed DAta Sampling (U-MIDAS) scheme (Foroni et al., 2015) into the LSTM architecture. We assess via both Monte Carlo simulations and empirical application the out-of-sample predictive performance. Our proposed models outperform the restricted MIDAS model even in a set up favorable to the MIDAS estimator. For real world application, we study forecasting a quarterly growth rate of Thai real GDP using a vast array of macroeconomic indicators both quarterly and monthly. Our LSTM with U-MIDAS scheme easily beats the simple benchmark AR(1) model at all horizons, but outperforms the strong benchmark univariate LSTM only at one and six months ahead. Nonetheless, we find that our proposed model could be very helpful in the period of large economic downturns for short-term forecast. Simulation and empirical results seem to support the use of our proposed LSTM with U-MIDAS scheme to nowcasting application.

Keywords: LSTM · Mixed frequency data · Nowcasting · Time series · Macroeconomic indicators

1 Introduction

Forecasting the present or very near future (also called "nowcasting") of an economic activity in order to update the state of economy is one of the crucial questions in macroeconomic area (Giannone et al., 2008). Nonetheless, the key macroeconomic data periodically released by government agencies, for example, a quarterly Gross Domestic Product (QGDP), are typically only available after several weeks or months and often revised a few months later (see Ghysels et al. (2018) for a recent discussion of macroeconomic data revision and publication delays). The low-frequency nature of QGDP data is challenging to policy makers and those who are in need of updating the state of economy for their decision making. This is so called mixed frequency data problem.

Researchers and practitioners working with the mixed frequency time series data usually pre-filter the data so that the frequency of left-hand and right-hand variables are equal. This raises the problem of how to exploit potentially useful information contained in the higher-frequency time series data. Various approaches have been proposed so far in the literature to avoid pre-filtering and to benefit from the information contained in the time series data observed at different frequencies (see Foroni & Marcellino (2013) for a comprehensive survey on the methods when dealing with mixed frequency data). Mixed DAta Sampling, denoted MIDAS, approach was first introduced by Ghysels et al. (2006, 2007) and has since been one of the main strands when modelling mixed frequency time series. MIDAS uses a flexible functional form to parameterize the weight given to each lagged regressors, such that the regressand and regressors can be sampled at different frequencies. Apart from capturing the rich information contained in high-frequency data, MIDAS also avoids the potential problems of information loss, measurement error, and timeliness caused by frequency transformation. Notable studies focusing on macroeconomic indicators include Clements & Galvão (2008); Armesto et al. (2009); Marcellino & Schumacher (2010); Kuzin et al. (2011); Breitung & Roling (2015). As long as the differences in sampling frequencies are large, implying that a curse of dimensionality might be relevant, MIDAS can be expected to perform well as the use of distributed lag polynomials can help avoiding parameter proliferation. However, when the differences in sampling frequencies are small, Foroni et al. (2015) show that the unrestricted lag polynomials in MIDAS regressions (also called U-MIDAS) generally performs better than the restricted MIDAS. Koenig et al. (2003) is one of the first papers which apply the U-MIDAS to forecast quarterly changes in real GDP. Clements & Galvão (2008) also consider U-MIDAS in their empirical forecast comparisons but call these U-MIDAS-like models as *mixed-frequency distributed lag* (MF-DL) models. Nonetheless, Foroni & Marcellino (2013) is the first work that provides the theoretical framework and parameter identification conditions for the underlying high-frequency model in the U-MIDAS regressions.

Several sources of nonstandard high-frequency data from private companies such as Google and Twitter are available for short-term economic prediction (Ettredge et al., 2005; Vosen & Schmidt, 2011; Choi & Varian, 2012; D'Amuri & Marcucci, 2017). Nonetheless, most of the nowcasting applications of MIDAS literature still rely on the standard high-frequency macroeconomic data (e.g. industrial production, employment, and capacity utilization) usually released by government agencies (Clements & Galvão, 2008; Marcellino & Schumacher, 2010; Kuzin et al., 2011). An important reason is the usually limited effective sample size of low-frequency data when involving a large number of variables sampled at high-frequency. If, for example, we consider to use 30 variables at 10 lags to forecast the quarterly GDP growth, MIDAS regression model with two parameters lag polynomial needs to estimate 60 parameters, whereas U-MIDAS regression model needs to estimate 300 parameters. However, the GDP growth series of developed countries are rarely, if any, more than 300 observations and far less than that for the developing countries.

In order to deal with such high-dimensional data environment, recent MIDAS literature start to rely on machine learning models such as tree-based algorithms (Qiu, 2020) and sparse-group LASSO (sg-LASSO) regression (Babii et al., 2021). To my knowledge, Xu et al. (2019) is the only work that applies deep learning architecture with mixed frequency data. They develop the ANN-MIDAS and ANN-U-MIDAS models which integrate the MIDAS projection approach into the artificial neural networks (ANNs). The proposed models outperform the benchmark MIDAS and U-MIDAS models in both Monte Carlo simulations and empirical applications. They also find that the ANN-MIDAS model is superior to ANN-U-MIDAS in terms of goodness-of-fit and predictive ability. So far, the potentials of machine learning and deep learning models on mixed frequency data have barely been explored. This paper fills these gaps in the literature demonstrating the use of LSTM with with time series data sampled at different frequencies.

We presents the potentials of deep learning models when applying with mixed frequency macroeconomic time series data. We apply the long short-term memory networks (LSTM) introduced by Hochreiter & Schmidhuber (1997). The LSTM is developed to overcome the vanishing error problems found in standard recurrent neural networks (RNNs) when many time lags between relevant input events and target signals are present (see Hochreiter et al. (2001) for more discussion). The LSTM has been one of the main workhorses in natural language processing, language translation, speech recognition, and sentiment analysis. LSTM and its variations has recently gained lots of attentions for financial time series forecasting application. For a comprehensive survey on financial time series forecasting with deep learning algorithms see Sezer et al. (2020). To deal with data sampled at different frequencies, we rely on the U-MIDAS scheme.

From machine learning perspective, the U-MIDAS scheme serves as an alternative feature engineering process that transforms the higher-frequency variables into a low-frequency vector, denoted frequency alignment, and allows us to exploit potentially useful information contained in the high-frequency time series data. We first assess via Monte Carlo simulation the predictive performance of our proposed LSTM based models, and find that our nowcasts and forecasts are superior to those produced by the benchmark models. For real world application, we study both nowcasting (i.e. 1-, -2, and 3-month ahead forecasts) and forecasting (i.e. 6-, 9-, and 12-month ahead forecasts) a quarterly growth rate of Thai real GDP using a vast array of macroeconomic indicators both quarterly and monthly. Although our proposed model outperform the benchmark models only at one and six month ahead, we find that our proposed model could be very helpful in the period of large economic downturns. We additionally compare our forecasts with those posted by the Bank of Thailand (hereinafter BOT). Our annualized QGDP forecasts are superior to the BOT implied annual GDP forecasts at shorter horizon (i.e. 1- and 2-month ahead).

The remainder of this paper is organized as follows: Section 2 presents alternative methods for the LSTM architecture when applying time series data sampled at different frequencies. Section 3 reports on Monte Carlo simulations the predictive performance to illustrate the potentials of the proposed model when compared to the original MIDAS regression. Section 4 provides empirical evidence, and we conclude in Section 5.

2 Alternative methods to apply mixed frequency time series data with LSTM

This section presents how the time series data sampled at mixed frequency can be accommodated into the LSTM architectures in both conventional and unconventional fashions. We first discuss the basic setup of LSTM and how we can apply it when there is only one frequency mismatch between output variable and all variables in the model specification. To deal with multiple frequency mismatches, we then briefly introduce the MIDAS approach and discuss how we can adopt the unrestricted MIDAS scheme into the LSTM architectures.

The LSTM is capable of learning the complex long time lag tasks without loss of short time lag capabilities by enforcing constant error flow through the memory cells, a unit of computation that replace traditional artificial neurons in the hidden layer of a network. However, the original version of LSTM does not work well when dealing with a continuous input stream (Gers et al., 2000). This paper apply the LSTM with a *forget gate* introduced by Gers et al. (2000). Following the description of Graves (2013) and Fischer & Krauss (2018), the LSTM is basically composed of an input layer, one or more hidden layers, and an output layer. The hidden layer(s) consists of a set of recurrently connected memory blocks. Each block contains one or more self-connected memory cells and three multiplicative units—an input gate (i_t), a forget gate (f_t), and an output gate (o_t)—that provide continuous analogues of write, read and reset operations for the cell state (c_t), respectively. For the LSTM used in this paper (Gers et al., 2000), the hidden layer function is implemented by the following group of functions:

$$i_t = \sigma(W_{xi}x_t + W_{ti}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{1a}$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{1b}$$
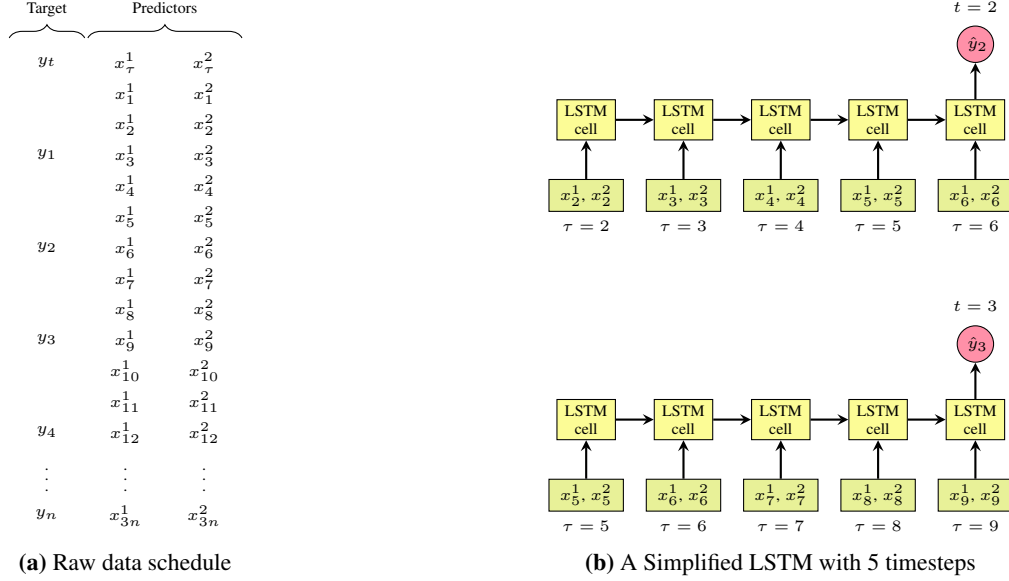$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{1c}$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{c0}c_t + b_0 \tag{1d}$$
$$h_t = o_t tanh(c_t) \tag{1e}$$

where σ is the logistic sigmoid function; and the input gate, forget gate, output gate, and cell state activation vectors are all the same size as the hidden vector (h_t). The weight matrices $Wx_i, W_{hi}, W_{ci}, W_{xf}, W_{hf}, W_{cf}, W_{xc}, W_{hc}, W_{xo}, W_{ho}$, and W_{co} have the obvious meaning, for example, W_{hi} is the hidden-input gate matrix, W_{xo} is the input-output gate matrix. The b terms denote bias vectors (e.g. b_i is input bias vector). See Fischer & Krauss (2018) for more detailed description of working sequences of LSTM.

2.1 Single frequency mismatch

In practice, ones always have to pass a 3 dimensional (3D) array of shape {batch size, time steps, number of features} as an input to the LSTM network. When output y_t and all variables $x_{k,t}$ for $k = 1, 2, 3, ..., K$ in the model specification are all sampled at the same frequency, both sequence stride s (period between successive output sequences) and sampling rate r (period between successive individual timesteps within variable sequences) are usually set to 1 so as to utilize all available information. Nonetheless, when variables $x_{k,t}$ are all sampled at the same frequency mismatch ratio $m > 1$ times faster than an output y_t for all $k = 1, 2, 3, ..., K$, such preprocessing strategy of $s = r = 1$ will not work as the number of higher-frequency series $x_{k,t}$ is now m times as many as the number of low-frequency series y_t. We denote this case the *single frequency mismatch*. Subscript τ_k is used hereinafter to indicate possibly different sampling frequency of higher-frequency variables, and, consequently, the frequency mismatch ratio of individual

Figure 1a: Raw data schedule

Target	Predictors	
y_t	x^1_τ	x^2_τ
	x^1_1	x^2_1
	x^1_2	x^2_2
y_1	x^1_3	x^2_3
	x^1_4	x^2_4
	x^1_5	x^2_5
y_2	x^1_6	x^2_6
	x^1_7	x^2_7
	x^1_8	x^2_8
y_3	x^1_9	x^2_9
	x^1_{10}	x^2_{10}
	x^1_{11}	x^2_{11}
y_4	x^1_{12}	x^2_{12}
$\vdots$	$\vdots$	$\vdots$
y_n	x^1_{3n}	x^2_{3n}

(a) Raw data schedule

Figure 1b: A Simplified LSTM with 5 timesteps

$t = 2$, output $\hat{y}_2$ — LSTM cells with inputs x^1_2, x^2_2 ($\tau = 2$); x^1_3, x^2_3 ($\tau = 3$); x^1_4, x^2_4 ($\tau = 4$); x^1_5, x^2_5 ($\tau = 5$); x^1_6, x^2_6 ($\tau = 6$).

$t = 3$, output $\hat{y}_3$ — LSTM cells with inputs x^1_5, x^2_5 ($\tau = 5$); x^1_6, x^2_6 ($\tau = 6$); x^1_7, x^2_7 ($\tau = 7$); x^1_8, x^2_8 ($\tau = 8$); x^1_9, x^2_9 ($\tau = 9$).

(b) A Simplified LSTM with 5 timesteps

Figure 1: Mixed frequency time series with single frequency mismatch ratios. (a) presents a raw data schedule in which the raw time series data are aligned to match with their actual period of occurrence. (b) presents the first two feasible sequences of LSTM when timesteps is set to 5.

higher-frequency series is indicated as m_k for $k = 1, 2, 3, ..., K$. Fig.1a presents a data schedule for the case that a low-frequency output y_t is sampled, say, quarterly and the frequency mismatch ratio between low-frequency output y_t and each higher-frequency variable x_{k,τ_k} is 3 for all $k = 1, 2$; all higher-frequency variables x_{k,τ_k} are therefore sampled monthly and we require that the number of observations of higher-frequency x_k is as exactly 3 times as the number of observations of low-frequency output y_t. A simple solution to apply LSTM with a model specification having only one frequency mismatch is then to preprocess the raw time series dataset with a sequence stride $s = 1$ and a sampling rate $r = m$ in order to match the occurrence of the higher-frequency variables $x_{k,t}$ to their low-frequency counterparts. Fig.1b show a simplified LSTM architecture with 5 timesteps in which the raw time series data are preprocessed with a sequence stride $s = 1$, and a sampling rate $r = 3$.
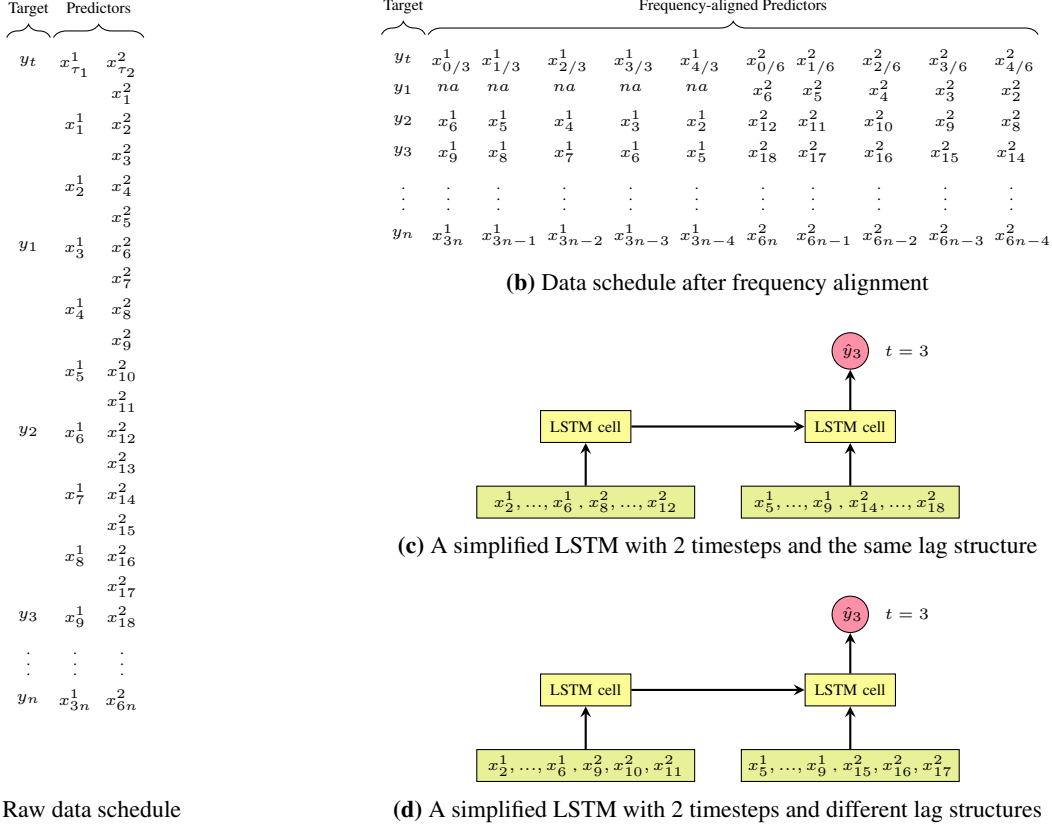
Unfortunately, the strategy of preprocessing time series data with sampling rate $r = m$ is not always workable when there are more than one frequency mismatch ratio in the model specification. Fig.2a shows an example of a quarterly series y_t, a monthly series $x^1_{\tau_1}$, and a bi-weekly series $x^2_{\tau_2}$. The frequency mismatch ratios are, therefore, 3 and 6 for the respective higher-frequency variables x_1 and x_2. As the data schedule shows, it is not possible to directly accommodate two higher-frequency variables with different frequency mismatch ratios into the same LSTM architecture. We denote this case the *multiple frequency mismatches*. To deal with multiple frequency mismatches, we adopt the technique widely used in econometrics with mixed frequency time series data called the *MIxed DAta Sampling* (MIDAS) approach (Ghysels et al., 2006, 2007).

2.2 MIDAS approach

To briefly introduce MIDAS regressions and simplify the exposition, we confine the discussion to a single variable x_t. Suppose that a higher-frequency series $x_t^{(m)}$ is sampled at m times faster than a low-frequency series y_t. Hence, say, with a quarterly series y_t and the frequency mismatch ratio $m = 3$, a higher-frequency series x_t is sampled monthly. Following the notation commonly used in Ghysels et al. (2004); Clements & Galvão (2008), a simple linear MIDAS regression for a single explanatory variable and h-step-ahead forecasting can be written as:

$$y_t = \alpha + \beta_1 B(L^{1/m}) x_{t-h}^{(m)} + \epsilon^m \tag{2}$$

where $B(L^{1/m}) = \sum_{j=0}^{J} \omega_j L^{j/m}$ is a polynomial of length J in the $L^{1/m}$ operator such that $L^{j/m} x_{t-h}^{(m)} = x_{t-h-j/m}^{(m)}$. In other words, the $L^{j/m}$ operator produces the value of x_t lagged by j/m periods (for simplicity, suppose we need to predict the end-of-quarter horizon, or $h = 0$). Given the quarterly/monthly example, the above equation represents a projection of a quarterly time series y_t onto a monthly time series x_t^m up to J monthly lags back. However, should a large number of lags of the $x_t^{(m)}$ be involved in a suitable polynomial $B(L^{1/m})$, ones would thus need to estimate

(a) Raw data schedule

Target	Predictors	
y_t	$x^1_{\tau_1}$	$x^2_{\tau_2}$
		x^2_1
	x^1_1	x^2_2
		x^2_3
	x^1_2	x^2_4
		x^2_5
y_1	x^1_3	x^2_6
		x^2_7
	x^1_4	x^2_8
		x^2_9
	x^1_5	x^2_{10}
		x^2_{11}
y_2	x^1_6	x^2_{12}
		x^2_{13}
	x^1_7	x^2_{14}
		x^2_{15}
	x^1_8	x^2_{16}
		x^2_{17}
y_3	x^1_9	x^2_{18}
$\vdots$	$\vdots$	$\vdots$
y_n	x^1_{3n}	x^2_{6n}

(b) Data schedule after frequency alignment

Target	Frequency-aligned Predictors									
y_t	$x^1_{0/3}$	$x^1_{1/3}$	$x^1_{2/3}$	$x^1_{3/3}$	$x^1_{4/3}$	$x^2_{0/6}$	$x^2_{1/6}$	$x^2_{2/6}$	$x^2_{3/6}$	$x^2_{4/6}$
y_1	na	na	na	na	na	x^2_6	x^2_5	x^2_4	x^2_3	x^2_2
y_2	x^1_6	x^1_5	x^1_4	x^1_3	x^1_2	x^2_{12}	x^2_{11}	x^2_{10}	x^2_9	x^2_8
y_3	x^1_9	x^1_8	x^1_7	x^1_6	x^1_5	x^2_{18}	x^2_{17}	x^2_{16}	x^2_{15}	x^2_{14}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
y_n	x^1_{3n}	x^1_{3n-1}	x^1_{3n-2}	x^1_{3n-3}	x^1_{3n-4}	x^2_{6n}	x^2_{6n-1}	x^2_{6n-2}	x^2_{6n-3}	x^2_{6n-4}

(c) A simplified LSTM with 2 timesteps and the same lag structure

$\hat{y}_3$ $t = 3$

LSTM cell → LSTM cell

$x^1_2, ..., x^1_6 , x^2_8, ..., x^2_{12}$ $x^1_5, ..., x^1_9 , x^2_{14}, ..., x^2_{18}$

(d) A simplified LSTM with 2 timesteps and different lag structures

$\hat{y}_3$ $t = 3$

LSTM cell → LSTM cell

$x^1_2, ..., x^1_6 , x^2_9, x^2_{10}, x^2_{11}$ $x^1_5, ..., x^1_9 , x^2_{15}, x^2_{16}, x^2_{17}$

Figure 2: Mixed frequency time series with multiple frequency mismatch ratios. (a) presents a raw data schedule in which the raw time series data are aligned to match with their actual period of occurrence. (b) presents a new data schedule in which the raw time series data are frequency-aligned following MIDAS approach. (c) presents the first feasible sequence of LSTM when timesteps are set to 2, and both variables take the same lag structure $j = \{0, 1, 2, 3, 4\}$. (d) also presents the first feasible sequence of LSTM when timesteps are set to 2, but a higher-frequency variable x^1 takes lag structure $j = \{0, 1, 2, 3, 4\}$ whereas a higher-frequency variables x^2 takes lag structure $j = \{1, 2, 3\}$.

many parameters. To avoid such parameter proliferation, Ghysels and his coauthors impose some restrictions upon the weight ω_j in the polynomial $B(L^{1/m})$. Following the pioneering work of Ghysels et al. (2005, 2006), the most commonly used parameterization of ω_j is the second-order exponential Almon lag, or $\theta = (\theta_1, \theta_2)'$:

$$\omega_j(\theta) = \frac{exp(\theta_1 j + \theta_j j^2)}{\sum_{j=1}^{J} exp(\theta_1 j + \theta_2 j^2)} \tag{3}$$

where the vector of hyperparameters θ is unknown, $\omega_j(\theta) \in [0, 1]$, and $\sum_{j=1}^{J} \omega_j(\theta) = 1$. Practically, with this specification, a simple linear MIDAS regression model in (2) can thus be written as:

$$y_t = \alpha + \sum_{j=0}^{J} b_j x^{(m)}_{t-h} + \epsilon^m \tag{4}$$

where $b_j = \beta_1 \omega_j(\theta)$ following Andreou et al. (2013) and Ghysels et al. (2016).[1] While model (4) is a linear model in terms of variables, the nonlinear constraints imposed upon the weight ω_j result in nonlinearities with respect to the vector of hyperparameters θ, and the model (4) is estimated by nonlinear least squares (NLS) estimator.

The MIDAS specification is particularly useful for high mismatch applications in financial time series data as shown by Ghysels et al. (2005, 2006). However, for quarterly/monthly setting of macroeconomic applications in which m is small and only a few higher-frequency lags are possibly needed to capture the weights, Foroni et al. (2015) show that the

---

[1]This coefficient b_j is also termed as *MIDAS coefficient* when estimated using R package *midasr* (Ghysels et al., 2016).

U-MIDAS model might be considered to be a good approximation to the true dynamic linear model.[2] When estimating, the U-MIDAS also applies the same model (4) as its MIDAS counterpart. However, without restriction Equation (3) being imposed, the U-MIDAS do not require NLS, and can be estimated by simple ordinary least square (OLS).

2.3 Frequency alignment as a feature engineering process

Practically, when estimating either MIDAS or U-MIDAS regression model, ones need to manipulate all variables observed at various frequencies and different length of observations. The process of transforming the higher-frequency variable x_t into a low-frequency vector $(x_{3t}, ...x_{3t-2})'$ is denoted the *frequency alignment* following Ghysels et al. (2016). Suppose we assume that the monthly data in the current quarter has explanatory power, say, $J = 2$ (i.e. a contemporaneous and two lags). This means that we need to model y_t as a linear combination of variables $x_t, x_{t-1/3}, x_{t-2/3}$ observed in each quarter t. Hence, the model (2) can be written in matrix notation as:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \alpha + \begin{bmatrix} x_3 & x_2 & x_1 \\ \vdots & \ddots & \vdots \\ x_{3n} & x_{3n-1} & x_{3n-2} \end{bmatrix} \begin{bmatrix} b_0 \\ \vdots \\ b_2 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix} \tag{5}$$

Since all the variables are now observed at the same frequency, this process enables a MIDAS regression to directly accommodate variables sampled at different frequencies and turns it into a classical time series regression. Although the Equation (5) presents a simple case of a single frequency mismatch, the procedure is very generalizable. Fig.2a presents a case of quarterly output y_t and two frequency mismatches $m_1 = 3$ (monthly) and $m_2 = 6$ (bi-weekly). Suppose we assume that a contemporaneous and four lags (i.e. $j = 0, 1, 2, 3, 4$) of both variables have explanatory power. The resulting data schedule after preprocessing raw time series with frequency alignment process has now additional eight features as shown in Fig.2b. Despite the fact that x_3^1, x_2^1, x_1^1 exist, their corresponding entries of the variable x_τ^1 in the row $t = 1$ are intentionally shown as non-available because a lag order of 4 is beyond its sampling frequency $m_1 = 3$ which causes the frequency alignment at $t = 1$ to fail. The guiding model specification can be written in matrix notation as:

$$\begin{bmatrix} y_2 \\ \vdots \\ y_n \end{bmatrix} = \alpha + \begin{bmatrix} x_6^1 & x_5^1 & x_4^1 & x_3^1 & x_2^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{3n}^1 & x_{3n-1}^1 & x_{3n-2}^1 & x_{3n-3}^1 & x_{3n-4}^1 \end{bmatrix} \begin{bmatrix} b_0^1 \\ \vdots \\ b_4^1 \end{bmatrix}$$
$$+ \begin{bmatrix} x_6^2 & x_5^2 & x_4^2 & x_3^2 & x_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{3n}^2 & x_{3n-1}^2 & x_{3n-2}^2 & x_{3n-3}^2 & x_{3n-4}^2 \end{bmatrix} \begin{bmatrix} b_0^2 \\ \vdots \\ b_4^2 \end{bmatrix} + \begin{bmatrix} \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} \tag{6}$$

The frequency alignment process can also be generalized to the case of multiple frequency mismatches with different length of lag vector. Moreover, the vector of lags included for each variable can also differently start from a given order rather than from a contemporaneous lag ($j = 0$). Fig.2c and 2d present the first feasible sequence of LSTM with 2 timesteps in which the raw time series data are preprocessed with frequency alignment. The former presents the case when both higher-frequency variables take the same lag structure (contemporaneous with 4 lags) as shown in Fig.2b. The latter provides an example of different lag structures where a higher-frequency variable x^1 takes lag structure $j = \{0, 1, 2, 3, 4\}$ and a higher-frequency variables x^2 takes lag structure $j = \{1, 2, 3\}$.

Setting aside the estimation and restrictions, both MIDAS and U-MIDAS approaches rely on the frequency alignment preprocessing process and explicitly treat lagged observations (at particular orders) as their variables. Put it differently, from a machine learning perspective, the frequency alignment can simply be viewed as an alternative feature engineering process. Hence, ones can directly adopt the U-MIDAS scheme into the LSTM, and estimate all parameters as its usual using standard library such as Keras in Python.

3 Monte Carlo Simulations

In this section, we assess via Monte Carlo simulations the predictive performance to illustrate the potentials of adopting the unrestricted lag polynomials projection in LSTM architectures. Specifically, we benchmark the out-of-sample

---

[2]Nonetheless, as also suggested by Foroni et al. (2015) that both MIDAS and U-MIDAS should be regarded as approximations to dynamic linear models, we should not expect one of these approaches to dominate with empirical data.

nowcasting (i.e. 1-, 2-, and 3-month ahead) and forecasting (i.e. 6-, 9-, and 12-month ahead) performance of LSTM against the MIDAS and U-MIDAS regression models.

3.1 Experimental design

The simulation exercise is designed in a way to discuss its application in macroeconomic forecasting, in particular forecasting quarterly GDP growth. To be relevant for empirical application in the next section, we focus on the case of quarterly/monthly ($m = 3$) data with small sample size environment. We follow the simulation design in Foroni et al. (2015) that directly uses the restricted MIDAS regression as the data generating process (DGP). The guiding DGP is explicitly defined as:

$$y_{t+h} = \alpha + \sum_{k=1}^{K} \sum_{j=0}^{J} b_j^k x_{k,t-j}^{(3)} + \epsilon_{t+h} \tag{7}$$

$$\epsilon_{t+h} \sim N(0,1), \forall t = 1, ..., T$$

For simplicity, a contemporaneous and the maximum lag order $J = 11$ lags (i.e. $j = 0, 1, 2, ..., 11$) are used for all higher-frequency variables. The parameterized lag coefficients $b_j^k(j; \theta) = \beta_j^k \omega_{kj}$, where the weights, ω_{ij}, are generated by the normalized exponential Almon lag polynomial (see Ghysels et al. (2016) for more detail) and defined as:

$$\omega_{kj} = \frac{exp(\theta_1^k j + \theta_2^k j^2)}{\sum_{j=1}^{J} exp(\theta_1^k j + \theta_2^k j^2)} \tag{8}$$

The monthly variables are generated as either AR(1) process with persistence equal to 0.9 or $x_{k,\tau} \sim N(0,1)$. The sample size (expressed in the low-frequency unit) is $T = \{50, 80\}$. Four different DGPs of low-frequency output y_t are therefore simulated and evaluated. For the sake of comparison, all experiments are designed to be the single frequency mismatch $m = 3$ class. All simulations take the number of higher-frequency variables $K = 3$. The parameter pairs of $\theta = (\theta_1, \theta_2)'$ are from the sets $\theta_1 = 0.7$ and $\theta_2 = \{-0.025, -0.1, -0.5\}$ following parameters commonly used in the MIDAS literature (e.g. Ghysels et al. (2016); Foroni et al. (2015)). To some extent, the simulations are in favor of the MIDAS as a small number of the most relevant variables are practically included in the MIDAS regression model so as to preserve parsimony. On the contrary, a large vector of lags will directly affect the forecasting performance of the U-MIDAS regression model as the number of parameters to be estimated is large with respect to the sample size.

We assess the forecasting performance of the proposed data preprocessing strategies discussed in previous section. For each DGP, all three higher-frequency variables generated are taken as the input data. The input data of the first two LSTM models are preprocessed using the sampling rate $r = m$ strategy as discussed in section 2.1. The timesteps are set to 6 and 12, respectively. We denote these models *Sampling-aligned* LSTM (SA-LSTM). To apply the U-MIDAS scheme, the input data of the other three models are preprocessed using MIDAS frequency alignment strategy as discussed in section 2.2. We denote these models *Frequency-aligned* LSTM (FA-LSTM). The FA-LSTM models are set to have the lag structure starting from the lag order zero (i.e. a contemporaneous lag) and take the vector of lag orders of [0:2], [0:5], and [0:11], respectively. Since they take all higher-frequency variables ($K = 3$), the number of variables used for each FA-LSTM model are $3 \times (J + 1) = 9, 18$, and 36, respectively. The timesteps of 4, 2, and 1 are applied for three FA-LSTM models.

We conduct a grid search to select the best hyperparameter combination of epochs, dropout, batch size, and the number of LSTM memory cells for each LSTM model in each forecasting horizon. As the objective of the simulation is not to search for the highest forecasting accuracy, only the small set of epochs = $\{25, 50\}$, dropout = $\{0, 0.4\}$, and batch size = $\{1, \lceil input/10 \rceil, \lceil input/2 \rceil\}$, where $input$ is the effective size of the training data and $\lceil \cdot \rceil$ denotes the ceiling function, are examined. For the number of LSTM memory cells, a set of $\{16, 32, 64, 128\}$ is examined for FA-LSTM models, and a set of $\{8, 16, 32\}$ is examined for SA-LSTM models. We simply exploit the first twenty simulation replications (i.e. $R = \{1, 2, 3, ..., 20\}$) that already prepared for the experiment as the dataset. For each simulation replication, the ratio of 60:40 is applied to split the data into training and validation samples. To minimize computational time, we evaluate the predictive performance of each hyperparameter combination on the validation samples using a fixed forecast scheme. The average root mean squared forecast error (RMSFE) over these twenty simulation replications are then compared across all hyperparameter combinations. Table A.1 presents hyperparameter combinations that achieves the lowest average RMSFE for all forecasting horizons and DGPs.

We benchmark the out-of-sample forecasting performance of the LSTM models against two alternatives: MIDAS and U-MIDAS regressions. We consider the model specifications that are parsimonious but possibly misspecified functional forms. Thus, only two higher-frequency variables are used in the model specification. For a given forecasting horizon, the same model specification is applied for both MIDAS and U-MIDAS models. All (U-)MIDAS regression models are estimated by midasr package (Ghysels et al., 2016) in R. The best model specification for each forecasting horizon is

thoroughly searched over the set of potential models defined by all combinations of the number of lags (2 to 12 lags), higher-frequency variable pair (3 possible pairs), and weighting functions (normalized and non-normalized exponential Almon lag polynomials). For simplicity, the lowest lags of high-frequency variables are simply set to be equal to the respective higher-frequency forecasting horizon h_m. An automatic selection is performed using the flexible function available in the midasr package. As applied in the LSTM models, the ratio of 60:40 is applied to split the data into training and testing data. The fixed model option is also set for forecasting. We select the model that achieves the lowest mean squared forecast error (MSE).

The out-of-sample forecasting performance is assessed across a series of leading index $d = \{m - 1, m - 2, 0, 0, 0, 0\}$, where d represents the number of higher-frequency observations that are available before the release of low-frequency output. Given the fixed frequency mismatch $m = 3$ in the experiments, this corresponds to higher-frequency forecasting horizons $h_m = \{1, 2, 3, 6, 9, 12\}$ and a sequence of low-frequency forecasting horizons $h = \{1, 1, 1, 2, 3, 4\}$. For each simulation replication, the ratio of 60:40 is also applied to split the data into training observations T_1 and evaluation observations T_2. Given the sample size $T = \{50, 80\}$, the evaluation observations $T_2 = \{20, 32\}$ are held out and used as out-of-sample data, and the rest $T_1 = T - T_2$ observations as training data. To evaluate forecasting performance, we assume that the information up to period $T_1 + \Omega$, where $\Omega = 0, 1, 2, ..., T_2 - h$, is available for low-frequency output and $T_1 + \Omega + d/m$ for higher-frequency variables. We then conduct one-horizon-ahead rolling forecast without re-estimation. This yields a forecast of the low-frequency output y_t for the h_m higher-frequency periods ahead for the out-of-sample observations $\hat{y}_{T_1 + \Omega + h_m}$. The corresponding squared forecast error $SE = (y_{T_1 + \Omega + h_m} - \hat{y}_{T_1 + \Omega + h_m})^2$ is thus used to calculate the RMSFE over the evaluation sample for each simulation replication. The number of simulations are fixed at $R = 1000$. Since the random initial conditions in the LSTM architecture may result in very different results, we estimate all LSTM models twice for each replication. The out-of-sample RMSFEs reported in the next section are therefore calculated by averaging over 2×1000 replications. All LSTM models are estimated by Keras library in Python.

3.2 Simulation results

Table 1-2 report detailed results of Monte Carlo simulation experiment. The U-MIDAS in simulation scenarios performs unsurprisingly poorly as the number of parameters to be estimated is large with respect to the sample size. As such, the restricted MIDAS substantially outperforms its unrestricted counterpart across all forecasting horizons and sample sizes. We apply the Diebold & Mariano (1995) test statistics (hereinafter, DM test) to examine whether the difference of RMSFE from the two competing models are statistically significant. Since all U-MIDAS perform poorly, we only assess the statistical difference of forecast accuracy between the restricted MIDAS and the LSTM for each forecasting horizon.

When DGP of higher-frequency variables is $x_{k,\tau} \sim N(0, 1)$, the LSTM with either sampling alignment (SA-LSTM) or frequency alignment (FA-LSTM) data preprocessing strategy outperform the MIDAS model across all higher-frequency forecasting horizons $h_m = \{1, 2, 3, 6, 9, 12\}$ and sample sizes $T = \{50, 80\}$. Almost all of the DM tests are statistically significant at the level of 5%. Although the SA-LSTM models performer best in most of the simulation scenarios generated by the normal distribution process, the forecasting performance of the FA-LSTM model does not substantially differ from that of the SA-LSTM model. However, these findings are not robust to the DGP of the AR(1) process with persistence equal to 0.9. All methods perform worse with persistent higher-frequency variables across all simulation scenarios. They all obtain larger RMSFE and larger standard deviation. The SA-LSTM model seems to be affected more compared to the other methods. Most of the forecasts produced by the SA-LSTM model perform significantly worse than the FA-LSTM model and the restricted MIDAS model. Although most of the SA-LSTM models still achieve the lowest average RMSFE, especially when $h_m = \{6, 9, 12\}$, many of them perform worse than the competing models both in terms of larger RMSFE and larger standard deviation. Interestingly, the FA-LSTM performs much better than the competing models when $h_m = \{1, 2, 3\}$. This simulation evidence encourages the use of LSTM with frequency alignment data preprocessing when ones need to nowcast the low-frequency series of interest.

To sum up, even in a set up favorable to the restricted MIDAS, the LSTM with either sampling alignment or frequency alignment data preprocessing still yields a better forecasting performance than MIDAS. Simulation results indicate that the sampling alignment strategy, which we simply adjust the sampling rate in the commonly used data preprocessing procedure, is an interesting strategy when the mixed frequency data problem has only one frequency mismatch. Nonetheless, the frequency alignment strategy provides more flexibility when ones need to forecast the low-frequency data series with multiple frequency mismatches. Although we cannot expect one of the approaches to dominate with empirical data, the simulation results seem to support the use of LSTM with U-MIDAS scheme when ones need to nowcast the macroeconomic low-frequency data series, such as GDP growth, in the quarterly/monthly setting.

4 Empirical application to Thai GDP growth

This section presents an empirical application of the proposed models to real world data. Specifically, we predict quarterly growth rate of Thai real GDP series using a vast array of macroeconomic indicators both quarterly and monthly, and assess the predictive performance of the SA-LSTM and FA-LSTM models across six higher-frequency forecasting horizons (i.e. $h_m = 1, 2, 3, 6, 9, 12$). The predictive performance of the proposed LSTM modes are benchmarked against variety of competing models. Our empirical exercises focus on non-seasonally adjusted QGDP series and mimics the real-time situation in the sense that the data publication delays are replicated; hence, this is a pseudo

Table 1: Out-of-sample RMSFE results for Monte Carlo simulation (DGP: $x_{k,\tau} \sim N(0,1)$)

Raw Obs.	h_m	MIDAS	U-MIDAS	SA-LSTM		FA-LSTM		
				[6, 0:0]	[12, 0:0]	[4, 0:2]	[2, 0:5]	[1, 0:11]
50	1	1.165	1.446	**1.095***	**1.101**	**1.119**	**1.125**	**1.138**
		(0.201)	(0.299)	(0.171)	(0.176)	(0.179)	(0.178)	(0.180)
	2	1.170	1.387	**1.104***	**1.121**	**1.124**	**1.129**	**1.136**
		(0.200)	(0.267)	(0.173)	(0.195)	(0.178)	(0.182)	(0.181)
	3	1.171	1.359	**1.114***	**1.129**	**1.127**	**1.132**	**1.148**
		(0.195)	(0.260)	(0.176)	(0.187)	(0.180)	(0.180)	(0.183)
	6	1.176	1.319	**1.121***	**1.130**	**1.130**	**1.137**	**1.147**
		(0.199)	(0.250)	(0.175)	(0.186)	(0.181)	(0.183)	(0.186)
	9	1.175	1.308	**1.112***	**1.125**	**1.129**	**1.142**	**1.150**
		(0.248)	(0.243)	(0.175)	(0.190)	(0.180)	(0.185)	(0.185)
	12	1.217	2.571	**1.114***	1.207	**1.135**	**1.146**	**1.155**
		(0.220)	(0.997)	(0.175)	(0.262)	(0.181)	(0.184)	(0.183)
80	1	1.120	1.167	**1.158**	**1.146**	**1.106***	1.117	**1.130**
		(0.145)	(0.155)	(0.175)	(0.189)	(0.140)	(0.143)	(0.146)
	2	1.134	1.180	**1.116**	**1.104***	**1.117**	**1.126**	1.134
		(0.145)	(0.156)	(0.143)	(0.140)	(0.141)	(0.147)	(0.147)
	3	1.136	1.216	**1.106***	**1.111**	**1.122**	**1.128**	**1.132**
		(0.151)	(0.169)	(0.140)	(0.142)	(0.143)	(0.146)	(0.146)
	6	1.146	1.261	**1.111***	**1.168**	**1.126**	**1.130**	**1.139**
		(0.153)	(0.180)	(0.144)	(0.183)	(0.147)	(0.147)	(0.146)
	9	1.155	1.322	**1.117**	**1.114***	**1.125**	**1.136**	**1.142**
		(0.154)	(0.200)	(0.143)	(0.142)	(0.145)	(0.146)	(0.150)
	12	1.153	1.389	**1.116***	**1.122**	**1.135**	**1.138**	**1.147**
		(0.155)	(0.214)	(0.142)	(0.146)	(0.147)	(0.148)	(0.146)

Notes: This table reports out-of-sample RMSFE results for simulation exercises. Except the first two columns, the table reports average RMSFE across 1,000 replications × 2 estimations per replication. Standard deviation is reported in parentheses. * indicates the lowest average RMSFE in the given forecasting horizon. Bold text indicates that the null hypothesis of the two methods having the same forecast accuracy is rejected at the 5% significance level using Diebold & Mariano (1995) test statistics. The timesteps and lag specification of the input data, which are fixed across all forecasting horizons, are shown for each LSTM model. Since the SA-LSTM models take no lagged variables, the lag specification of [0:0] is simply shown. The optimal batch size for each LSTM model in each forecasting horizon are separately reported in Table A.1. Given the number of higher-frequency variables $K = 3$, the total number of variables used in each LSTM model are $K \times J + 1 = 3, 3, 9, 18, 36$, respectively.

real-time forecast evaluation. To be policy relevant, we additionally recalculate the GDP growth as an annual rate using the quarterly forecasts computed by the FA-LSTM model. The annualized QGDP growth forecasts are then compared with the publicly available forecasts produced by the Bank of Thailand model at various forecasting horizons.

4.1 Data

Regarding the macroeconomic series, we first collect all quarterly and monthly series available in the BOT database. The time of the first available observation differs from series to series, and all of them are available up to the end of

Table 2: Out-of-sample RMSFE results for Monte Carlo simulation (DGP: $x_{k,\tau} = 0.9x_{k,\tau-1} + \epsilon_\tau$)

Raw obs.	h_m	MIDAS	U-MIDAS	SA-LSTM		FA-LSTM		
				[6, 0:0]	[12, 0:0]	[4, 0:2]	[2, 0:5]	[1, 0:11]
50	1	2.066	2.311	**2.108**	**2.633**	1.922	**1.634***	**1.741**
		(0.616)	(0.696)	(1.395)	(3.581)	(0.924)	(0.559)	(0.538)
	2	2.142	3.809	**2.229**	**3.013**	**2.114**	**1.999**	**1.963***
		(0.713)	(1.523)	(1.359)	(6.200)	(0.938)	(0.646)	(0.599)
	3	2.251	3.440	**2.014**	**3.128**	**2.193**	**1.989***	**2.108**
		(0.729)	(1.190)	(0.738)	(6.857)	(0.750)	(0.625)	(0.659)
	6	2.415	3.501	**2.495**	**2.141***	2.469	**2.403**	**2.507**
		(0.815)	(1.193)	(1.226)	(0.975)	(0.816)	(0.768)	(0.782)
	9	2.787	3.630	**2.616***	**4.194**	**2.686**	**2.743**	2.798
		(1.044)	(1.250)	(0.849)	(9.771)	(0.894)	(0.905)	(0.934)
	12	2.826	3.215	**2.750***	**2.979**	**2.859**	**2.870**	**3.005**
		(0.938)	(0.995)	(0.900)	(3.350)	(0.993)	(0.978)	(1.066)
80	1	1.901	2.079	1.780	**2.663**	**1.530**	**1.524***	**1.571**
		(0.449)	(0.48)	(0.813)	(4.020)	(0.416)	(0.376)	(0.360)
	2	1.947	2.094	1.842	**2.970**	**1.750**	**1.602***	**1.653**
		(0.447)	(0.469)	(0.847)	(4.437)	(0.474)	(0.365)	(0.384)
	3	2.026	2.079	**1.842**	**2.557**	**1.824**	**1.764***	**1.830**
		(0.457)	(0.443)	(0.784)	(2.965)	(0.467)	(0.390)	(0.447)
	6	2.156*	2.187	2.183	**2.213**	**2.372**	**2.223**	**2.388**
		(0.462)	(0.470)	(0.536)	(1.466)	(0.707)	(0.536)	(0.594)
	9	2.423*	2.656	**2.547**	**3.460**	**2.570**	**2.579**	**2.631**
		(0.575)	(0.608)	(0.690)	(5.761)	(0.656)	(0.660)	(0.672)
	12	2.617	2.754	2.648	2.535*	**2.762**	**2.737**	**2.794**
		(0.644)	(0.636)	(0.649)	(1.158)	(0.774)	(0.718)	(0.743)

Notes: This table reports out-of-sample RMSFE results for simulation exercises. Except the first two columns, the table reports average RMSFE across 1,000 replications × 2 estimations per replication. Standard deviation is reported in parentheses. * indicates the lowest average RMSFE in the given forecasting horizon. Bold text indicates that the null hypothesis of the two methods having the same forecast accuracy is rejected at the 5% significance level using Diebold & Mariano (1995) test statistics. The timesteps and lag specification of the input data, which are fixed across all forecasting horizons, are shown for each LSTM model. Since the SA-LSTM models take no lagged variables, the lag specification of [0:0] is simply shown. The optimal batch size for each LSTM model in each forecasting horizon are separately reported in Table A.1. Given the number of higher-frequency variables $K = 3$, the total number of variables used in each LSTM model are $K \times J + 1 = 3, 3, 9, 18, 36$, respectively.

2020. The longest series start at 1993M1 and 1993Q1 for monthly and quarterly series, respectively. To prevent very short samples, we filter out the series of which the first period is available later than the cut-off period of 2001M1. The final dataset of variables contains five quarterly and 35 monthly series. We impute zero values for the series that start later than 1993M1 for the monthly series and 1993Q1 for the quarterly series. The full list of the series with further details appear in the Online Appendix B. The quarterly real GDP (QGDP) series in chained volumes from 1993Q1 to 2020Q4 are available from the Office of the National Economic and Social Development Council (NESDC). The growth rate is then calculated directly as $100 \times (QGDP_{t+1} - QGDP_t)/QGDP_t$.

Except QGDP series, all macroeconomic series are final revised data and supposedly published with a delay of up to one month according to the observed typical pattern between the end of the reference period and the date of the respective release. This implies that if we need to forecast the QGDP growth for 2019Q1 three months before the quarter ends (i.e. $h_m = 3$), we will need data up to the end of 2018M12. Given that all macroeconomic series are published with a delay, it effectively means that the contemporaneous lags are 2018M11 and 2018Q3 for monthly and quarterly series, respectively. The QGDP series is periodically published within eight weeks after the end of the reference period; hence, it is available for forecasting at the beginning of the third month of the next quarter. Again, for example, if we need to forecast the QGDP growth at the end of 2019Q1, the first available lag to be used as the autoregressive (AR) term of the QGDP growth series is 2018Q3 for $h_m = 2, 3$, and 2018Q4 for $h_m = 1$.

4.2   Empirical design

We evaluate the predictive performance of the LSTM using U-MIDAS scheme (i.e. FA-LSTM) and the SA-LSTM models on the out-of-sample data, containing 52 quarters, from 2008Q1 to 2020Q4. Six projections are computed for six higher-frequency forecasting horizons ($h_m = 1, 2, 3, 6, 9, 12$) for each QGDP observation of the out-of-sample period. All competing models are recursively configured and estimated at the beginning of each quarter using the information up to the end of the last available month for a given forecasting horizon h_m. For the first forecast at the end of 2008Q1, for example, the competing models are configured and estimated at the beginning of 2008M3 ($h_m = 1$), 2008M2 ($h_m = 2$), 2008M1 ($h_m = 3$), 2007M10 ($h_m = 6$), 2007M7 ($h_m = 9$), and 2007M4 ($h_m = 12$) using the data up to the end of 2008M2, 2008M1, 2007M12, 2007M9, 2007M6, and 2007M3, respectively.

As a natural comparison, we compare out-of-sample forecasts of the proposed models with the univariate LSTM, denoted as UNI-LSTM, model. The next section shows that, before the COVID-19 pandemic, this relatively much simpler univariate LSTM benchmark turns out to be a very strong competitor to the proposed models. Following the MIDAS literature for short-term GDP growth predictions (Marcellino & Schumacher, 2010; Babii et al., 2021), the simple benchmark AR(1) model is also used as an additional benchmark for assessing the short-term forecasting ($h_m = 1, 2, 3$) performance of the proposed models. Although the LSTM effectively capture the sequential information in the input data, the nature of the economy is highly dynamic and its structure may substantially change over time. If that is the case, the advantages of LSTM architecture having both a short-term and a long-term memory might be trivial. We therefore report the results for the artificial neural networks (ANN) model using the U-MIDAS scheme, denoted U-MIDAS-ANN, as another competing model.

To assess the predictive performance of the models, we compare RMSFE of the proposed models with the benchmark models in a recursive forecasting exercise. We additionally relate the mean squared forecast error (MSFE) of the proposed models to the variance of QGDP growth over the evaluation period following Marcellino & Schumacher (2010). A ratio of less than one indicates that the predictive performance of a model is to some extent informative. We also apply the DM test statistics (Diebold & Mariano, 1995) to examine whether the difference of RMSFE from the two competing models are statistically significant.

4.3   Model specification and hyperparameter tuning

The following ADL-MIDAS-like equation is applied as a *guiding* model specification for the proposed FA-LSTM and the alternative U-MIDAS-ANN models for all forecasting horizons:

$$y_{t+h} = \alpha + \sum_{j=0}^{P_Q-1} \lambda_j y_{t-j} + \sum_{k=1}^{K_Q} \sum_{j=0}^{P_Q-1} b_j^k x_{k,t-j}^Q + \sum_{k=1}^{K_M} \sum_{j=0}^{P_M-1} b_j^k x_{k,t-j}^M + \epsilon_{t+h} \tag{9}$$

where x_k^Q and x_k^M represents low-frequency quarterly and higher-frequency monthly variable, respectively. The guiding specification (9) involves a contemporaneous and $P_Q - 1$ lags of x_t^Q and a contemporaneous and $P_M - 1$ lags of x_t^M. Following Clements & Galvão (2008), we add the AR dynamics y_{t-j} to a guiding model specification so as to provide a more flexible dynamic specification. This approach is, from the econometric point of view, useful to handle eventual serial correlation in the idiosyncratic components (Marcellino & Schumacher, 2010). For simplicity, we apply the same

lag orders for all variables having the same frequency mismatch ratio; hence, quarterly variables and AR dynamics y_t included in a given model are always forced to have the same contemporaneous and $P_Q - 1$ lags. Equation (10) presents a guiding model specification for the SA-LSTM model:

$$y_{t+h} = \alpha + \sum_{k=1}^{K_M} \sum_{j=0}^{P_M-1} b_j^k x_{k,t-j}^M + \epsilon_{t+h} \tag{10}$$

It differs from that of the FA-LSTM model since the SA-LSTM model can not take more than one frequency mismatch ratio in the model specification. This guiding model specification is used for all nowcasting and forecasting horizons.

Since a choice of model specification (i.e. the number of lag orders P and a set of K variables) would consequently affect a selection of hyperparameters, both optimal number of lag orders and hyperparameters are simultaneously selected using the grid search method for all competing models, except the benchmark UNI-LSTM and AR(1) models. For each combination of lag orders and hyperparameters, we first use the Least Absolute Shrinkage and Selection Operator (LASSO) regression (Tibshirani, 1996) to select a set of variables which minimize 5-fold time series cross validation (i.e. the previous four quarters of a given out-of-sample forecast) in terms of MSFE. The resulting variables are presumably considered as the best available set of variables for forecasting the next out-of-sample period. Second, we apply the ratio of 80:20 to split the available information into training and validation samples, fit the model using training samples, and then compute the forecasts for the validation samples using a fixed forecast scheme. Lastly, we compare the predictive performance on validation samples of all combinations in terms of RMSFE. Since the random initial conditions in the algorithm may result in very different forecasts, each combination is estimated three times and calculate an average RMSFE. We select a combination of lag orders and hyperparameters which achieves the lowest RMSFE. Note that the whole model configuration process is recursively conducted in every out-of-sample forecast; hence, the combination of variables, maximal lag orders, and hyperparameters are allowed to change over time. Since the number of foreign tourists unprecedentedly drop to zero in April 2019 due to the travel restrictions imposed around the world caused by the COVID-19 pandemic, the number of foreign tourists variable is therefore directly added into the selected set of variables starting from 2019Q1 unless it is automatically selected by the LASSO.

Concerning for the candidate hyperparameters of the FA-LSTM model, we consider a following set of batch size = $\{\lceil input/5\rceil, \lceil input/3\rceil, \lceil input/2\rceil\}$; timesteps = $\{3, 6, 12, 18\}$; and the number of LSTM memory cells = $\{(128), (256), (512), (128, 128), (256, 256)\}$, where the number of elements inside the brackets equals to the number of hidden layer(s). As for the lag specification, we consider a small set of monthly lag order = $\{3, 6, 9\}$ and quarterly lag order = $\{1, 2, 3\}$. The same candidate set of batch size, LSTM memory cells size, monthly lag order, and quarterly lag order are also considered for the FA-ANN model. As for the SA-LSTM model, the different set of candidate hyperparameters are considered except the batch size. We consider the larger set of the number of LSTM memory cells = $\{32, 64, 128, [32, 32], [64, 64], [128, 128]\}$ and timesteps = $\{3, 6, 12, 18, 24\}$. Lastly, for the UNI-LSTM model, we consider a set of batch size = $\{\lceil input/5\rceil, \lceil input/3\rceil, \lceil input/2\rceil, \lceil input/1\rceil\}$; the number of LSTM memory cells = $\{(8), (16), (32), (64)\}$; and timesteps = $\{3, 6, 12, 15, 18, 24\}$. To prevent overfitting, we employ the early stopping of five epochs and set the maximum epoch to 200. The optimal set of hyperparameters and lag specification is therefore selected from a total of 540, 90, 96, and 135 combinations when recursively performing the grid search for FA-LSTM, SA-LSTM, UNI-LSTM, and FA-ANN models, respectively.

4.4 Empirical results

Given the model configuration in each out-of-sample forecast for each forecasting horizon h_m, we conduct one-horizon-ahead forecast using the horizon-specific model. Except the simple AR(1) benchmark model, all competing models are estimated 100 times so as to account for the stochastic nature of the algorithm, and take the average value as the out-of-sample forecast of the respective model. The estimation sample depends on the model configuration and the information available at each period in time.

Relative forecast accuracy of the competing models on the non-seasonally adjusted QGDP growth series can be found in Table 3. We first focus only on the full out-of-sample 52 periods (2008Q1-2020Q4). The results show that the FA-LSTM model performs significantly better than the benchmark UNI-LSTM model only at the 6-month horizon, and is better, numerically but not statistically, at the 1-month horizon. The SA-LSTM model perform worse than those of the UNI-LSTM model at all horizons. The alternative FA-ANN model is better, numerically but not statistically, only at the 2-month horizon. Overall, the benchmark UNI-LSTM model seems to produce more accurate forecasts compared to these alternatives at all horizons. Compared with the simple benchmark AR(1) model, both FA-LSTM and the alternative FA-ANN models perform significantly better for all forecasting horizons. The SA-LSTM model also produce more accurate forecasts than the benchmark AR(1) for all horizon, the difference is only statistically significant at the 3-month horizon. Most of the relative MSFEs of all models are unsurprisingly less than one for all horizons as the variance of the non-seasonally adjusted QGDP growth series is considerably high. In terms of the

Table 3: Out-of-sample forecast comparisons for non-seasonally adjusted QGDP growth

	2008Q1-2020Q4			Exclude large downturns			Large downturns		
	Rel. RMSE		Rel. MSE	Rel. RMSE		Rel. MSE	Rel. RMSE		Rel. MSE
	Uni	AR(1)		Uni	AR(1)		Uni	AR(1)	
1-month horizon									
FA-LSTM	0.917	**0.563**	0.318	**1.046**	**0.593**	0.349	**0.728**	**0.504**	0.255
SA-LSTM	1.330	0.817	0.669	1.564	**0.887**	0.782	0.968	**0.671**	0.452
UNI-LSTM	3.593	**0.615**	0.378	3.206	**0.567**	0.320	4.340	**0.693**	0.482
FA-ANN	**1.020**	**0.627**	0.393	1.151	**0.653**	0.424	0.831	**0.576**	0.333
AR(1)	**1.627**	5.846	1.001	**1.763**	5.652	0.994	**1.442**	6.260	1.003
2-month horizon									
FA-LSTM	**1.044**	**0.608**	0.370	**1.198**	**0.676**	0.455	**0.745**	**0.457**	0.209
SA-LSTM	**1.323**	0.770	0.594	**1.381**	**0.780**	0.604	**1.229**	**0.753**	0.569
UNI-LSTM	3.403	**0.582**	0.339	3.191	**0.565**	0.317	3.837	**0.613**	0.377
FA-ANN	0.999	**0.582**	0.339	**1.126**	**0.636**	0.402	**0.761**	**0.466**	0.218
AR(1)	**1.718**	5.846	1.001	**1.771**	5.652	0.994	**1.632**	6.260	1.003
3-month horizon									
FA-LSTM	**1.237**	**0.786**	0.618	**1.262**	**0.792**	0.623	**1.193**	**0.774**	0.601
SA-LSTM	**1.431**	**0.909**	0.827	**1.317**	**0.827**	0.679	**1.608**	**1.044**	1.092
UNI-LSTM	3.713	**0.635**	0.404	3.547	**0.628**	0.391	4.062	**0.649**	0.422
FA-ANN	1.043	**0.663**	0.440	**0.902**	**0.566**	0.318	**1.249**	**0.810**	0.659
AR(1)	**1.574**	5.846	1.001	**1.594**	5.652	0.994	**1.541**	6.260	1.003
6-month horizon									
FA-LSTM	**0.938**		0.420	**0.962**		0.449	**0.890**		0.361
SA-LSTM	**1.259**		0.756	**1.296**		0.815	**1.184**		0.638
UNI-LSTM	4.034		0.477	3.950		0.485	4.217		0.455
FA-ANN	**1.101**		0.578	**1.067**		0.552	**1.167**		0.619
9-month horizon									
FA-LSTM	**1.197**		0.629	1.229		0.727	1.112		0.440
SA-LSTM	**1.295**		0.738	**1.204**		0.698	**1.500**		0.801
UNI-LSTM	3.874		0.440	3.936		0.482	3.731		0.356
FA-ANN	1.041		0.476	**1.016**		0.497	**1.100**		0.431
12-month horizon									
FA-LSTM	1.071		0.563	**0.960**		0.516	**1.345**		0.644
SA-LSTM	**1.276**		0.799	**1.242**		0.863	**1.372**		0.670
UNI-LSTM	4.092		0.490	4.243		0.560	3.728		0.356
FA-ANN	1.012		0.502	**0.878**		0.432	**1.326**		0.625

Notes: This table reports out-of-sample forecast comparisons for six forecasting horizons ($h_m = 1, 2, 3, 6, 9, 12$). Columns 2-4 report comparison results for the full out-of-sample 52 periods (2008Q1-2020Q4). Columns 8-10 report comparison results for the large economic downturn periods (see main text for more discussion), while columns 5-7 report the results for the out-of-sample data which exclude the large economic downturn periods. In each comparison period, column *Uni* and column *AR(1)* report root mean squared forecasts error relative to the univariate LSTM and AR(1) models, respectively, while column *Rel. MSE* reports mean squared forecasts error relative to the variance of QGDP growth. The variance of QGDP growth series in each evaluation period is 34.138, 32.151, and 39.066, respectively. As for the relative RMSFE measures, bold text indicates that the null hypothesis of the two methods having the same forecast accuracy is rejected at the 5% significance level using Diebold & Mariano (1995) test statistics.

relative MSFEs, the UNI-LSTM also provide the most information content at all horizons. Only the FA-LSTM model that is more informative at one and six months ahead. We also observe that, as new monthly information becomes available, the forecast accuracy of the FA-LSTM model at the nowcasting horizons ($h_m = 1, 2, 3$) improve with the horizon. However, all competing models being considered here cannot always improve with this information. This could be a result of the non-optimality in our model configuration process that simultaneously select the optimal set of variables and hyperparameters.

Following Babii et al. (2021), we additionally examine whether the gains in predictive performance are persistent over recursions or simply dominated by differences in performance in only a few periods. We compute the cumulative sum
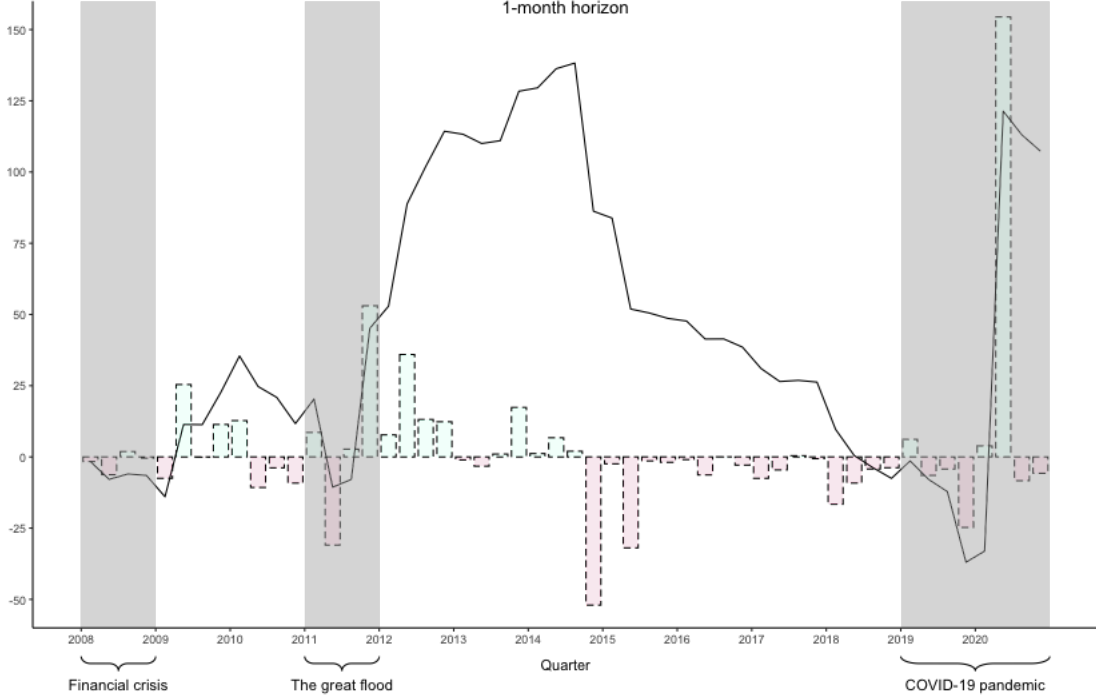
Figure 3: Cumulative sum of loss differentials (CUMSFE) of FA-LSTM model versus the benchmark univariate LSTM model at 1-month horizon.

Notes: Solid line corresponds to CUMSFE. Positive columns correspond to positive loss differential ($e^2_{q,uni} - e^2_{q,fa}$) indicating that the FA-LSTM model has smaller squared forecast errors compared to the benchmark univariate LSTM model, and negative columns indicate the opposite.

of squared forecast error (CUMSFE) loss differential of the FA-LSTM and the benchmark UNI-LSTM models for the 1-month horizon. The CUMSFE for the benchmark UNI-LSTM model versus the FA-LSTM model for quarter $q = t, t + 1, t + 2, ..., t + k$ is defined as:

$$CUMSFE_{t,t+k} = \sum_{q=t}^{t+k}(e^2_{q,uni} - e^2_{q,fa}) \qquad (11)$$

where t is the first prediction at the end of 2016Q1, $t + k$ is the last prediction at the end of 2020Q4, and $e_{q,uni}$ and $e_{q,fa}$ are the forecast errors for quarter q from the benchmark UNI-LSTM and the FA-LSTM models, respectively. For an out-of-sample forecast of quarter t, a positive value of loss differential $e^2_{q,uni} - e^2_{q,fa}$ indicates that the FA-LSTM model has smaller squared forecast errors compared to the benchmark UNI-LSTM model. Figure 3 shows the plot of CUMSFE and loss differential for the 1-month horizon. We observe that the gains in predictive performance are not persistent throughout the out-of-sample periods. Notably, the large gains (i.e. positive loss differential) of the FA-LSTM model can be observed during the unprecedented events like the 2011 great flood in Thailand and the COVID-19 pandemic. This result suggests that our predictive performance might be mostly driven by these unusual periods in our out-of-sample observations. To verify this, we thus turn our focus back to Table 3. The last three columns (8-10) report the results for the out-of-sample forecasts in the financial crisis period, the great flood in Thailand period, and the COVID-19 pandemic period. For simplicity, we define the financial crisis period as 2008Q1-2008Q4; the great flood in Thailand period as 2011Q1-2011Q4, and the COVID-19 period as 2019Q1-2020Q4. Columns 5-7 report the results for the out-of-sample periods which exclude the aforementioned large economic downturns.

The results, in terms of ranking, for the out-of-sample forecasts in the periods of large economic downturn and the periods which exclude large economic downturn largely remain the same compared to the previously discussed ones. Here, we focus only on the nowcasting horizon $h_m = 1, 2, 3$. The forecast accuracy of the FA-LSTM model improves in

14

the large economic downturn periods; however, deteriorates when we exclude the large economic downturns. Conversely, the forecast accuracy of the benchmark UNI-LSTM model notably deteriorate in the large economic downturn, but improve when we remove the large economic downturn periods. Compared with the benchmark UNI-LSTM model, the results indicate that the FA-LSTM model perform significantly better during economic downturn periods at 1- and 2-month horizons; however, the benchmark UNI-LSTM model fits better when the economic downturn is absent. The results imply that our FA-LSTM model can help to nowcast QGDP during unprecedented events.

To be policy relevant, we additionally compare the annualized QGDP forecasts of the FA-LSTM model with the publicly available BOT model implied annual GDP forecasts at $h_m = 1, 3, 6, 9, 12$. To this end, we exploit both the forecasts produced by FA-LSTM model and the actual QGDP of the prior quarter(s) in the same year. For the 12-month ahead ($h_m = 12$) annual forecast at the beginning of 2008, for example, we use the forecasts of 2008Q1 at 3-month horizon, 2008Q2 at 6-month horizon, 2008Q3 at 9-month horizon, and 2008Q4 at 12-month horizon. Then, for the 9-month ahead ($h_m = 9$) annual forecast at the beginning of April 2008, we use the actual QGDP growth of 2008Q1, the forecasts of 2008Q2 at 3-month horizon, 2008Q3 at 6-month horizon, and 2008Q4 at 9-month horizon. Note that we abstract from additional complications such as those resulting from publication delay of the QGDP series.
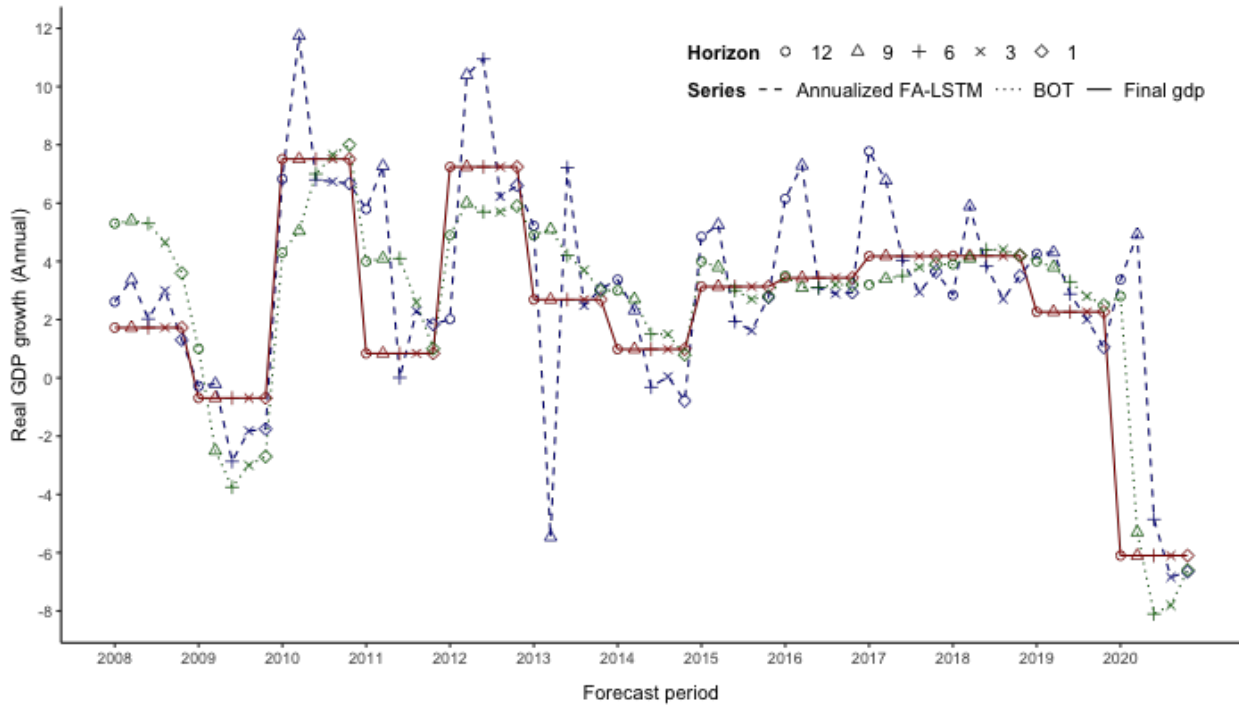
Figure 4: Annualized QGDP growth of the FA-LSTM model versus the BOT model implied annual GDP growth. *Notes*: For each year, the projection of annual GDP growth are made at the beginning of January ($h_m = 12$), April ($h_m = 9$), July ($h_m = 6$), October ($h_m = 3$), and December ($h_m = 1$). Dashed line corresponds to annualized QGDP growth produced by the FA-LSTM model, dotted line corresponds to the BOT model implied annual GDP growth, and solid line corresponds to the final annual GDP growth.

Concerning for the BOT forecast, the figures are compiled from various documents officially published by the BOT. Specifically, we take the annual GDP growth forecasts that appear in the minutes of the monetary policy committee meetings, the monetary policy report, inflation report, and the press release, which are all publicly available on the BOT websites. We use the first available forecast that published later than the beginning of the annualized QGDP forecast period for each forecasting horizon. Again, for example, for the 12-month ahead ($h_m = 12$) annual forecast at the beginning of 2008, the 2008 annual GDP growth forecast published in January 2008 inflation report is used.

In Figure 4, we plot the annualized QGDP growth forecast of the FA-LSTM model versus BOT model implied annual forecast for $h_m = 1, 3, 6, 9, 12$. In each year, we intentionally plot all five forecasts to present the development of the forecasts in each forecasting horizon h_m. The BOT model unsurprisingly produces smoother forecasts as the FA-LSTM model performs poorly at longer horizons, i.e. $h_m = 9, 12$. However, we can observe substantially improvement at the shorter horizon $h_m = 1$ and 3. In Table 4, we reports the predictive performance of both models at each forecasting horizon. The results confirm that the forecast accuracy of the annualized QGDP growth forecast of the FA-LSTM

Table 4: Forecast comparisons for annual GDP growth

	RMSE	
	Annualized FA-LSTM	BOT
1-month horizon	0.861	0.890
3-month horizon	1.054	1.367
6-month horizon	1.873	1.833
9-month horizon	4.764	1.920
12-month horizon	3.764	3.215

Notes: This table reports out-of-sample forecast comparisons for five forecasting horizons ($h_m = 1, 3, 6, 9, 12$). The last two columns report the RMSFE of the forecasts between 2008-2020 based on the annualized QGDP growth of the FA-LSTM model and BOT model implied annual GDP growth forecasts, respectively.

model is better than the BOT model implied forecast at shorter horizons $h_m = 1$ and 3. While the BOT model implied forecasts have better forecast accuracy than the annualized QGDP growth forecasts at longer horizons $h_m = 9$ and 12; the forecast accuracy of both models at 6-month horizon are not substantially different.

5 Conclusions

This paper demonstrates the potentials of applying the LSTM with time series data sampled at different frequencies. Our contribution is to combine methods from the literature on machine learning and on mixed frequency time series data. We adopt the U-MIDAS scheme into the LSTM architecture as a forecasting tool. From machine learning perspective, the U-MIDAS scheme serves as an alternative feature engineering process that transforms the higher-frequency variables into a low-frequency vector, denoted frequency alignment, and allows us to exploit potentially useful information contained in the high-frequency time series data.

We have shown via Monte Carlo simulation the predictive performance of the LSTM with sampling alignment strategy (i.e. the SA-LSTM), and the LSTM using U-MIDAS scheme (i.e. the FA-LSTM). Both models largely outperform the MIDAS regression even in a set up favorable to the restricted MIDAS. The simulation results suggest that the FA-LSTM is particularly suited to provide nowcasts of output variables in the quarterly/monthly data with small sample size. We then continue our assessment in the empirical application for quarterly growth rate of Thai real GDP. The results show that the univariate LSTM is generally a strong benchmark compared to the commonly used AR(1) model. The SA-LSTM performs poorly compared to the benchmark univariate LSTM model at all horizons. We find that the FA-LSTM performs significantly better than the benchmark univariate LSTM model only at the 6-month horizon, and is better, numerically but not statistically, at the 1-month horizon. To be policy relevant, we additionally compare the annualized QGDP forecasts of the FA-LSTM with the publicly available BOT model implied annual GDP forecasts for 1-, 3-, 6-, 9-, and 12-month horizons. The results show that the forecast accuracy of the annualized QGDP growth forecast of the FA-LSTM model is better than the BOT model implied forecast at shorter horizons ($h_m = 1, 3$).

To sum up, both the Monte Carlo simulation and the empirical application show that the LSTM using U-MIDAS scheme provides more flexibility when ones need to nowcast the macroeconomic low-frequency data series, such as GDP growth, in the quarterly/monthly data. This does not mean that our proposed LSTM using U-MIDAS scheme can not be applied to other frequency mismatch ratios. The frequency alignment procedure can be generalized to other data sampled at different frequencies, such as daily or unconventional search data. Nonetheless, there are some limitations that should be taken into account for future research. Similar to conventional LSTM model, the proposed LSTM using U-MIDAS scheme requires an optimal set of hyperparameters. And even worse, the proposed model additionally requires an optimal set of lag specification; consequently, the computational burden is even larger as a large number of hyperparameters and lag specification combinations are required to be considered. The model configuration strategy used in this paper is very time-consuming and obviously not optimal. This is an interesting area for future research.

References

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

Armesto, M. T., Hernández-Murillo, R., Owyang, M. T., & Piger, J. (2009). Measuring the information content of the beige book: A mixed data sampling approach. *Journal of Money, Credit and Banking*, *41*(1), 35–55.

Babii, A., Ghysels, E., & Striaukas, J. (2021). Machine learning time series regressions with an application to nowcasting.

Breitung, J. & Roling, C. (2015). Forecasting inflation rates using daily data: A nonparametric midas approach. *Journal of Forecasting*, *34*(7), 588–603.

Choi, H. & Varian, H. (2012). Predicting the present with google trends. *Economic record*, *88*, 2–9.

Clements, M. P. & Galvão, A. B. (2008). Macroeconomic forecasting with mixed-frequency data: Forecasting output growth in the united states. *Journal of Business & Economic Statistics*, *26*(4), 546–554.

Diebold, F. X. & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *13*(3), 253–263.

D'Amuri, F. & Marcucci, J. (2017). The predictive power of google searches in forecasting us unemployment. *International Journal of Forecasting*, *33*(4), 801–816.

Ettredge, M., Gerdes, J., & Karuga, G. (2005). Using web-based search data to predict macroeconomic statistics. *Communications of the ACM*, *48*(11), 87–92.

Fischer, T. & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669.

Foroni, C. & Marcellino, M. (2013). A survey of econometric methods for mixed-frequency data. Technical report, European University Institute.

Foroni, C., Marcellino, M., & Schumacher, C. (2015). Unrestricted mixed data sampling (midas): Midas regressions with unrestricted lag polynomials. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *178*(1), 57–82.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with lstm. *Neural computation*, *12*(10), 2451–2471.

Ghysels, E., Horan, C., & Moench, E. (2018). Forecasting through the rearview mirror: Data revisions and bond return predictability. *The Review of Financial Studies*, *31*(2), 678–714.

Ghysels, E., Kvedaras, V., & Zemlys, V. (2016). Mixed frequency data sampling regression models: the r package midasr. *Journal of statistical software*, *72*(1), 1–35.

Ghysels, E., Santa-Clara, P., & Valkanov, R. (2004). The midas touch: Mixed data sampling regression models. Technical report, Anderson Graduate School of Management, UCLA.

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

Ghysels, E., Santa-Clara, P., & Valkanov, R. (2006). Predicting volatility: getting the most out of return data sampled at different frequencies. *Journal of Econometrics*, *131*(1-2), 59–95.

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

Giannone, D., Reichlin, L., & Small, D. (2008). Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics*, *55*(4), 665–676.

Graves, A. (2013). Generating sequences with recurrent neural networks.

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Koenig, E. F., Dolmas, S., & Piger, J. (2003). The use and abuse of real-time data in economic forecasting. *Review of Economics and Statistics*, *85*(3), 618–628.

Kuzin, V., Marcellino, M., & Schumacher, C. (2011). Midas vs. mixed-frequency var: Nowcasting gdp in the euro area. *International Journal of Forecasting*, *27*(2), 529–542.

Marcellino, M. & Schumacher, C. (2010). Factor midas for nowcasting and forecasting with ragged-edge data: A model comparison for german gdp. *Oxford Bulletin of Economics and Statistics*, *72*(4), 518–550.

Qiu, Y. (2020). Forecasting the consumer confidence index with tree-based midas regressions. *Economic Modelling*, *91*, 247–256.

Sax, C. & Eddelbuettel, D. (2018). Seasonal adjustment by x-13arima-seats in r. *Journal of Statistical Software*, *87*(1), 1–17.

Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, *90*, 106181.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

Vosen, S. & Schmidt, T. (2011). Forecasting private consumption: survey-based indicators vs. google trends. *Journal of forecasting*, *30*(6), 565–578.

Xu, Q., Zhuo, X., Jiang, C., & Liu, Y. (2019). An artificial neural network for mixed frequency data. *Expert Systems with Applications*, *118*, 127–139.

APPENDIX

Appendix A Optimal Hyperparameters selected for Monte Carlo Simulation

Table A.1: Optimal hyperparameter combinations for Monte Carlo simulation

h_m	Raw Obs	SA-LSTM [6, 0:0]				SA-LSTM [12, 0:0]				FA-LSTM [4, 0:2]				FA-LSTM [2, 0:5]				FA-LSTM [1, 0:11]			
		p1	p2	p3	p4	p1	p2	p3	p4	p1	p2	p3	p4	p1	p2	p3	p4	p1	p2	p3	p4
Panel A. $x_{k,\tau} \sim N(0,1)$																					
1	50	25	0.4	15	32	25	0.4	15	32	25	0.4	14	16	25	0.4	15	32	25	0.0	15	16
	80	50	0.4	3	8	25	0.4	5	16	25	0.4	23	32	25	0.0	24	32	25	0.4	24	16
2	50	25	0.4	15	16	50	0.0	15	8	25	0.0	14	16	25	0.0	15	32	25	0.0	15	32
	80	50	0.4	24	16	25	0.4	24	32	25	0.4	23	32	25	0.0	24	16	25	0.0	24	16
3	50	50	0.4	15	8	25	0.4	3	8	25	0.4	14	32	25	0.4	15	32	25	0.4	15	64
	80	25	0.4	24	32	25	0.0	24	32	25	0.0	23	32	25	0.4	24	16	25	0.4	24	32
6	50	50	0.0	15	8	25	0.4	3	8	25	0.0	14	16	25	0.4	15	32	25	0.0	14	32
	80	25	0.4	24	32	25	0.4	5	16	25	0.4	23	32	25	0.0	24	32	25	0.4	23	32
9	50	25	0.0	14	8	50	0.4	14	8	25	0.0	14	16	25	0.0	14	16	25	0.4	14	16
	80	25	0.4	23	32	25	0.4	23	32	25	0.0	23	16	25	0.4	23	16	25	0.0	23	16
12	50	25	0.4	13	16	50	0.4	13	16	25	0.4	13	16	25	0.0	14	16	25	0.0	13	16
	80	25	0.0	22	16	25	0.4	22	32	25	0.0	22	32	25	0.0	23	16	25	0.0	22	16
Panel B. $x_{k,\tau} = 0.9x_{k,\tau-1} + \epsilon_\tau$																					
1	50	50	0.4	3	16	50	0.4	15	32	25	0.4	1	128	25	0.4	15	128	25	0.0	1	128
	80	50	0.0	3	16	25	0.4	1	32	25	0.0	23	64	50	0.4	24	64	50	0.4	24	32
2	50	50	0.4	1	8	50	0.4	3	8	50	0.4	1	128	25	0.4	15	16	50	0.0	1	64
	80	50	0.0	5	16	50	0.4	5	16	50	0.4	23	16	25	0.4	24	128	25	0.4	24	128
3	50	50	0.4	15	16	25	0.4	1	8	25	0.0	14	16	25	0.0	15	32	25	0.4	15	32
	80	25	0.4	5	32	50	0.4	24	32	25	0.0	23	64	25	0.4	24	64	25	0.0	24	128
6	50	50	0.4	15	32	25	0.4	14	32	25	0.4	14	16	25	0.0	15	32	25	0.4	14	32
	80	25	0.0	24	32	50	0.4	23	16	25	0.4	5	16	25	0.4	24	32	25	0.0	23	16
9	50	50	0.4	14	8	25	0.4	3	32	25	0.4	14	16	25	0.0	14	32	25	0.4	14	32
	80	25	0.4	5	16	25	0.4	1	32	25	0.4	23	16	25	0.4	23	32	25	0.4	23	16
12	50	50	0.4	13	8	25	0.0	3	8	25	0.0	13	16	25	0.4	14	16	25	0.0	13	32
	80	25	0.0	22	32	50	0.4	22	8	25	0.4	22	16	25	0.4	23	16	25	0.0	22	16

This table reports the optimal set of hyperparameters selected by the grid search method discussed in the main text. Column 1 and 2 indicate the higher-frequency forecasting horizons and the sample size, respectively. The remaining columns respectively report the optimal values of epoch (p_1), dropout (p_2), batch size (p_3), and the number of LSTM memory cells (p_4) for the LSTM models in each forecasting horizon. Panel A and B report the optimal hyperparameters for the experiments in which the higher-frequency variables are generated as $x_{k,\tau} \sim N(0,1)$ and AR(1) process with persistence equal to 0.9, respectively. The timesteps and lag specification of the input data, which are fixed across all forecasting horizons, are shown for each LSTM model. Since the SA-LSTM models take no lagged variables, the lag specification of [0:0] is simply shown. Given that three higher-frequency variables ($K = 3$) are used, the total number of variables used for each LSTM model are thus $K \times (J + 1) = 3, 3, 9, 18, 36$.

Appendix B   Data description

All quarterly and monthly macroeconomic series are from the Bank of Thailand (BOT) database. All series are final revised data and supposedly published with a delay of up to one month according to the observed typical pattern between the end of the reference period and the date of the respective release.

Table B.1: Data description table

Description	Identifier	Since	SA	Transformation
1 year treasury Bill	FM_RT_001	2001M1	No	MoM % change
10 years treasury bill	FM_RT_001	2001M1	No	MoM % change
USD Mid FX RATE	FM_FX_001	1981M1	No	MoM % change
GBP Mid FX RATE	FM_FX_001	1981M1	No	MoM % change
JPY Mid FX RATE	FM_FX_001	1981M1	No	MoM % change
Manufacturing production index	EC_EI_001	1993M1	Yes	MoM % change
Gross value added tax (at 2000 prices)	EC_EI_001	1993M1	Yes	MoM % change
Domestic automobiles sales	EC_EI_001	1993M1	Yes	MoM % change
Authorized capital of newly registered companies	EC_EI_002	1993M1	Yes	MoM % change
Construction areas permitted	EC_EI_002	1993M1	Yes	MoM % change
Number of foreign tourists	EC_EI_028	1993M1	Yes	Level change
Stock exchange of Thailand (SET) index	EC_EI_002	1993M1	Yes	MoM % change
Broad money (at 2000 prices)	EC_EI_002	1993M1	Yes	MoM % change
Oil price inverse index	EC_EI_002	1993M1	Yes	MoM % change
Business sentiment index - performance	EC_EI_005	1999M1	No	Index
Business sentiment index - total order book	EC_EI_005	1999M1	No	Index
Business sentiment index - investment	EC_EI_005	1999M1	No	Index
Business sentiment index - employment	EC_EI_005	1999M1	No	Index
Land and building transactions nationwide	EC_EI_009	1995M1	No	MoM % change
Registered applicants	EC_EI_018	1995M1	No	MoM % change
Vacancies	EC_EI_018	1995M1	No	MoM % change
Placements	EC_EI_018	1995M1	No	MoM % change
Retail sales index - non-durable goods	EC_EI_032	2000M1	Yes	MoM % change
Retail sales index - durable goods	EC_EI_032	2000M1	Yes	MoM % change
Retail sales index - department stores, etc.	EC_EI_032	2000M1	Yes	MoM % change
Retail sales index - motor vehicles and fuel	EC_EI_032	2000M1	Yes	MoM % change
Wholesales index - non-durable goods	EC_EI_033	2000M1	Yes	MoM % change
Wholesales index - durable goods	EC_EI_033	2000M1	Yes	MoM % change
Wholesales index - intermediate goods	EC_EI_033	2000M1	Yes	MoM % change
Agricultural products export value index	EC_EI_026	2000M1	Yes	MoM % change
Fishery products export value index	EC_EI_026	2000M1	Yes	MoM % change
Manufactured products export value index	EC_EI_026	2000M1	Yes	MoM % change
Consumer goods import value index	EC_EI_025	2000M1	Yes	MoM % change
Raw materials import value index	EC_EI_025	2000M1	Yes	MoM % change
Capital goods import value index	EC_EI_025	2000M1	Yes	MoM % change
Land index	EC_EI_008	1991Q1	No	MoM % change
Single-detached house (including land) index	EC_EI_008	1991Q1	No	MoM % change
Town house (including land) index	EC_EI_008	1991Q1	No	MoM % change
Total gross NPLs outstanding	FI_NP_003	1991Q1	No	MoM % change
Total credit outstanding of credit card	FI_CB_080	1991Q1	Yes	MoM % change

Notes: The *Description* column presents a macroeconomic time series name. The second column *Identifier* denotes an assigned Report ID of a time series in the BOT database. The third column gives the first available period of information for a time series. The column *SA* reports whether a seasonal patterns of a time series is removed. The last column *Transformation* denotes the data transformation applied to a time series.

Table B.1 reports the transformation applied to each macroeconomic time series. Most of the macroeconomic series are transformed using their original information. The following macroeconomic series are needed to be processed prior

to subsequent transformation: the monthly number of foreign tourists, the monthly business sentiment indexes, the monthly retail sales indexes, and the monthly wholesales indexes.

To this end, we re-reference all four series of the monthly business sentiment index from 50 to zero so as to clearly represent the direction of business sentiment. The seasonal patterns of all series of the monthly retail sales indexes and the monthly wholesales indexes are removed using the X-13ARIMA-SEATS algorithm available in the seasonal package (Sax & Eddelbuettel, 2018) in R prior to calculating the percentage change. The monthly number of foreign tourists data used in this paper are from two series. The old series of the seasonally adjusted number of foreign tourists had been published since 1993M1 and was obsolete after 1996M12. The BOT has since published the new series of non-seasonally adjusted number of foreign tourists until present. To this end, we first deseasonalized the number of foreign tourists from 1997M1 to 20203 using the X-13ARIMA-SEATS algorithm. The observations from 2020M4 to 2020M12 are intentionally excluded to avoid inappropriate deseasonalization because the number of foreign tourists are either zero or drastically low since 2020M4 due to the COVID-19 pandemic. The original data are simply used if the observations from 2020M4 to 2020M12 are required when estimating the model.