Astin Bulletin (2025), **55**, pp. 76–96
doi:10.1017/asb.2024.27

International Actuarial Association
Association Actuarielle Internationale

RESEARCH ARTICLE

Yield curve extrapolation with machine learning

Shinobu Akiyama[1] and Naoki Matsuyama[2]

[1]Graduate School of Advanced Mathematical Sciences, Meiji University, Tokyo, Japan
[2]Graduate School of Advanced Mathematical Sciences (AMS), School of Interdisciplinary Mathematical Sciences, Meiji University, Tokyo, Japan
Corresponding author: Naoki Matsuyama; Email: ma2yama@meiji.ac.jp

Received: 1 August 2023; **Revised:** 22 May 2024; **Accepted:** 21 June 2024; **First published online:** 30 October 2024

Keywords: Long short-term memory; Nelson–Siegel model; neural network; Smith–Wilson method; Svensson model; yield curve extrapolation.

Abstract

Yield curve extrapolation to unobservable tenors is a key technique for the market-consistent valuation of actuarial liabilities required by Solvency II and forthcoming similar regulations. Since the regulatory method, the Smith–Wilson method, is inconsistent with observable yield curve dynamics, parsimonious parametric models, the Nelson–Siegel model and its extensions, are often used for yield curve extrapolation in risk management. However, it is difficult for the parsimonious parametric models to extrapolate yield curves without excessive volatility because of their limited ability to represent observed yield curves with a limited number of parameters. To extend the representational capabilities, we propose a novel yield curve extrapolation method using machine learning. Using the long short-term memory architecture, we achieve purely data-driven yield curve extrapolation with better generalization performance, stability, and consistency with observed yield curve dynamics than the previous parsimonious parametric models on US and Japanese yield curve data. In addition, our method has model interpretability using the backpropagation algorithm. The findings of this study prove that neural networks, which have recently received considerable attention in mortality forecasting, are useful for yield curve extrapolation, where they have not been used before.

1. Introduction

The extrapolation of yield curves to unobservable tenors is a key technique for the market-consistent valuation of actuarial liabilities required by the EU Solvency II regulatory framework, introduced in 2016, and similar forthcoming regulations such as the Insurance Capital Standard (ICS) for internationally active insurance groups. The regulatory-compliant method for the extrapolation is the Smith–Wilson (SW) method (Smith and Wilson, 2001) with exogenous assumptions about a fixed asymptotic forward rate and the speed of convergence. However, this method is not necessarily suitable for risk management because the assumptions are arbitrary and inconsistent with observable yield curve dynamics. Additionally, because the SW method is strictly assumption-driven, even if the extrapolated tenors include an observable tenor, the possible poor estimation accuracy of the yield for the observable tenor is not discussed at all. Jørgensen (2018) suggests that the undervaluation of actuarial liabilities resulting from the use of the regulatory-compliant SW method can be massive compared with the results obtained from some alternative methods. Leiser and Kerbeshian (2019) note that many practitioners use the SW method when it is mandated but not in other situations.

As an alternative, parsimonious parametric models, such as the Nelson–Siegel (NS) model (Nelson and Siegel, 1987), the Svensson model (Svensson, 1994), and their variations with exogenously given asymptotic yields are often used for yield curve extrapolation in risk management. However, the NS and Svensson models have a limited ability to represent yield curves because of their limited number of parameters, which can lead to excessive volatility in the extrapolation results, as highlighted by

© The Author(s), 2024. Published by Cambridge University Press on behalf of The International Actuarial Association. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

Signorelli *et al*. (2022). Using an exogenously given asymptotic yield along with the parsimonious parametric models can reduce the volatility of the extrapolation results; however, it introduces arbitrariness problems, as in the SW method.

Dynamic models related to yield curve extrapolation can be found in the literature, including the dynamic NS model (Diebold and Li, 2006) and the (real-world) affine term structure model-based approach (Balter *et al*., 2013). These dynamic models are Bayesian extensions of the original models and are designed for risk measurement. While they have the advantage of providing a confidence interval for extrapolation, they do not provide the estimation accuracy and stability that are the key concerns of the deterministic models described above.

Despite its importance, the requirements of the dynamics of the yield curve extrapolation model for risk management have not been fully discussed in the literature. Since yield curve extrapolation far beyond the last observable tenor (LOT) is inherently unverifiable, attention is often focused on the stability of the extrapolation. However, inconsistent yield curve dynamics between the observed and extrapolated yield curves impairs the continuity of risk management because the liability cash flows outside the LOT are expected to transition to those inside the LOT after a period of time; in particular, this inconsistency is unacceptable for those planning parallel-shift hedging (i.e., hedging with shorter-tenor financial instruments with a focus on the parallel-shift component of yield curve dynamics) to manage the interest rate risks of liabilities outside the LOT. Thus, ensuring reasonable consistency with the observed yield curve dynamics is a key requirement for yield curve extrapolation in risk management, and the stability of the extrapolation results needs to be discussed in this context (i.e., more stability does not necessarily mean better); this implies that extrapolation models should have a reproducibility of the observed yield curves, which is related to the observed yield curve dynamics, together with a generalization performance against overfitting, which can be indicated by an estimation error on a test data. Among the existing extrapolation models, only parametric models (e.g., NS and Svensson) have a reproducibility of observed yields but have limitations as pointed out in the literature.

To achieve the above requirements beyond the limitations of parametric models, we propose a neural network (NN)-based yield curve extrapolation method. While extrapolation and forecasting are two different tasks, NNs have been used for yield curve forecasting as NN-assisted parametric approaches, including the multivariate autoregressive approach (Numes *et al*., 2019) and the state-space approach (Kauffmann *et al*., 2022). Unlike these NN-assisted parametric approaches, our yield curve extrapolation approach is a purely data-driven generative NN approach without assuming any parametric structure and has not been attempted in the literature. Data-driven yield curve extrapolation, which requires the generation of an output data series that is much longer than the input data series, tends to lose features of the input data in the long output generation process. For this reason, among the various machine learning methods including NNs, we chose the long short-term memory (LSTM) architecture because it can generate data series that retain long-term memory of the characteristics of the input data. Using the LSTM architecture, we achieve a purely data-driven yield curve extrapolation with better estimation accuracy and stability than the previous parsimonious parametric models using US and Japanese yield curve data.

The remainder of this paper is organized as follows. Section 2 reviews previous parametric approaches to yield curve extrapolation; Section 3 reviews NNs for time series data; Section 4 proposes our NN-based extrapolation method; Section 5 explains the yield curve data used and the calibration procedure of the proposed model; Section 6 presents the results of the performance evaluation of the proposed model against the benchmark models; Section 7 shows the model interpretability using the backpropagation algorithm. Finally, Section 8 concludes the paper.

2. Parametric approaches

The NS model is the most popular parametric model for reconstructing observed yield curves and has four interpretable parameters. The NS model gives a zero-coupon yield at maturity t as:

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

$$r(t) = \beta_0 + \beta_1\left(\frac{1-\exp\left(-\frac{t}{\lambda}\right)}{\frac{t}{\lambda}}\right) + \beta_2\left(\frac{1-\exp\left(-\frac{t}{\lambda}\right)}{\frac{t}{\lambda}} - \exp\left(-\frac{t}{\lambda}\right)\right). \tag{2.1}$$

In Equation (2.1), β_0, which is independent of t and corresponds with the zero-coupon yield at infinite maturities, is interpreted as a level factor corresponding to a parallel shift, β_1, whose factor loading is a decreasing function of t, is a slope factor, and β_2, whose factor loading is a mountainous function of t, is a curvature factor. The model parameters are estimated to minimize the mean squared error (MSE) of the model in reconstructing the observed zero-coupon yield. However, the NS model is limited in that it can only represent a yield curve with a single peak or trough.

Svensson (1994) proposes an extension of the NS model to six parameters to represent a hump in the yield curve. The Svensson model gives a zero-coupon yield at maturity t as:

$$\begin{aligned} r(t) = \beta_0 &+ \beta_1\left(\frac{1-\exp\left(-\frac{t}{\lambda_1}\right)}{\frac{t}{\lambda_1}}\right) + \beta_2\left(\frac{1-\exp\left(-\frac{t}{\lambda_1}\right)}{\frac{t}{\lambda_1}} - \exp\left(-\frac{t}{\lambda_1}\right)\right) \\ &+ \beta_3\left(\frac{1-\exp\left(-\frac{t}{\lambda_2}\right)}{\frac{t}{\lambda_2}} - \exp\left(-\frac{t}{\lambda_2}\right)\right). \end{aligned} \tag{2.2}$$

These models were not originally designed for yield curve extrapolation; however, they can be used for extrapolation by substituting unobservable maturity years for t in the equations. Although the limited number of parameters in these parsimonious models makes them easier to estimate and interpret, it also introduces large variability in the model parameters used to represent the observed yield curve, which can lead to excessive volatility in the extrapolated yield.

To provide an extrapolation with an appropriate level of volatility, Signorelli *et al*. (2022) introduce a stability term into the objective function to estimate the Svensson model, penalizing the MSE corresponding to the changes from the previous estimates, with an arbitrarily defined intensity. However, the appropriate setting of the stability intensity is challenging, particularly during periods of rapid changes in observed yield curves, such as the recent one, because of the large conflict between the estimation error and the volatility of the estimate. Many practitioners use a combination of the NS model and a cubic spline with a fixed asymptotic interest rate to control the variability of the extrapolated yield curves (Leiser and Kerbeshian, 2019); however, the fixed asymptotic interest rate causes the same problems as in the SW method. Kort and Vellekoop (2016) reduce the arbitrariness and market inconsistency of the SW method by estimating the asymptotic forward rate directly from the prices and cash flows of traded instruments in the market; however, the speed of convergence parameter is still arbitrarily defined, and the estimated values are too volatile.

3. Neural networks for time series data

This section provides a brief overview of NNs for time series data. For a more theoretical background on the NNs, we refer to Wüthrich and Merz (2022). The NNs typically consist of an input layer, hidden layers, an output layer, neurons in each layer, links between the neurons in adjacent layers, and activation functions. As any compactly supported continuous function can be uniformly approximated by a two-layer NN with a continuous sigmoidal activation function (Cybenco, 1989), the NNs are considered to have a universal approximation capability for a broad class of functions. The NNs without cyclic links are called feedforward NNs (FNNs) and NNs with cyclic links are called recurrent NNs (RNNs). Convolutional NN (CNN), a sparsely connected FNN used to learn the neighborhood effect of the data, and RNN are often used to learn sequential data.

In a typical FNN architecture, the d_{i+1}-dimensional vector y^{i+1} representing the output value of the neurons in the $i+1$-th layer is determined by the d_i-dimensional output vector y^i in the previous

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

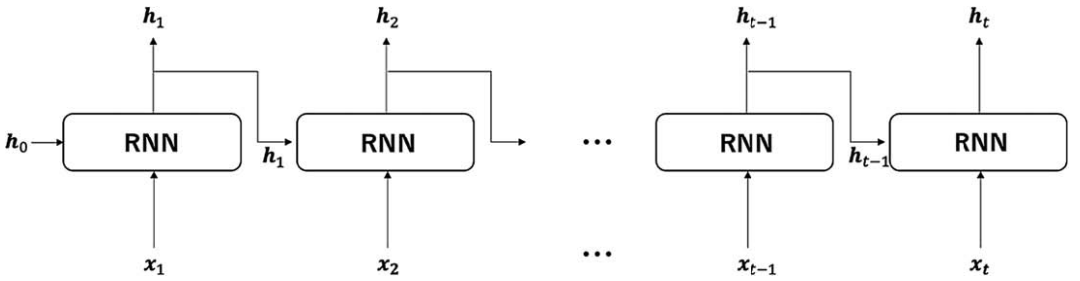


Figure 1. *Typical RNN architecture.*

layer, the activation function ϕ, the weight matrix $W^i \in \mathbb{R}^{d_{i+1} \times d_i}$, and the bias vector $b^i \in \mathbb{R}^{d_{i+1}}$ as follows:

$$y^{i+1} = \phi\left(W^i y^i + b^i\right), \tag{3.1}$$

where the weight matrices and the bias vectors can be trained using stochastic gradient descent (SGD).

Although CNNs, proposed by LeCun *et al*. (1990), are generally considered to be effective for two-dimensional images and three-dimensional spatial data, CNNs have recently been applied to one-dimensional (1D) time series data. The CNN replaces the product term $W^i y^i$ in Equation (3.1) with the convolution, as described below; it is often performed in three stages. In the first stage, the convolution with shared weights is performed; in the second stage, the value obtained by the convolution is nonlinearly transformed by an activation function; finally, a pooling function that returns a representative value of the input data is applied to output the value. The 1D CNN uses the convolution filter $W^{i,j} \in \mathbb{R}^{d \times m}$ $(j = 1, \ldots, J)$ instead of the weight matrix W^i in Equation (3.1), where $m \in \mathbb{N}$ denotes the kernel size of the filter and J denotes the number of filters. The output of layer i, $y^i \in \mathbb{R}^{d \times T}$, is transformed into

$$y^{i+1}_{j,k} = \phi\left(\sum_{s=1}^{m}\sum_{l=1}^{d} W^{i,j}_{l,s} y^i_{l,k+s-1} + b^{i,j}\right);$$
$$k = 1, \ldots, T+1-m; y^{i+1} \in \mathbb{R}^{J \times (T+1-m)}. \tag{3.2}$$

Thereafter, the pooling function is used, which generally returns the maximum, minimum, or average value within each window region of the input data.

RNNs, proposed by Elman (1990), have a recurrent architecture as shown in Figure 1 and are suitable for learning time series data. To enable the learning of the time series data, each layer receives the output of the previous layer in addition to the data at the corresponding time, which requires two weight matrices corresponding to the two inputs. For each RNN layer with k-dimensional units (neurons), the output of the RNN layer at time t is calculated from the input data $\boldsymbol{x}_t \in \mathbb{R}^d$ and the output of the previous RNN layer $\boldsymbol{h}_{t-1}$ as follows:

$$\boldsymbol{h}_t = \phi(U\boldsymbol{h}_{t-1} + W\boldsymbol{x}_t + \boldsymbol{b}), \tag{3.3}$$

using an activation function ϕ, a weight matrix $W \in \mathbb{R}^{k \times d}$ for the data at time t, a weight matrix $U \in \mathbb{R}^{k \times k}$ for the output of the previous RNN layer, and a bias vector $\boldsymbol{b} \in \mathbb{R}^k$. The weight matrices and the bias vector can be trained using the SGD with backpropagation through time (SGD-BPTT).

LSTM, a type of RNN, is proposed by Hochreiter and Schmidhuber (1997) to overcome the vanishing gradient problem of conventional RNNs, which cannot learn the long-term time dependence in time series data. Each layer of a typical RNN receives the output of the previous layer as its only past information; thus, so it can learn only short-term memory. However, each block of an LSTM receives

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

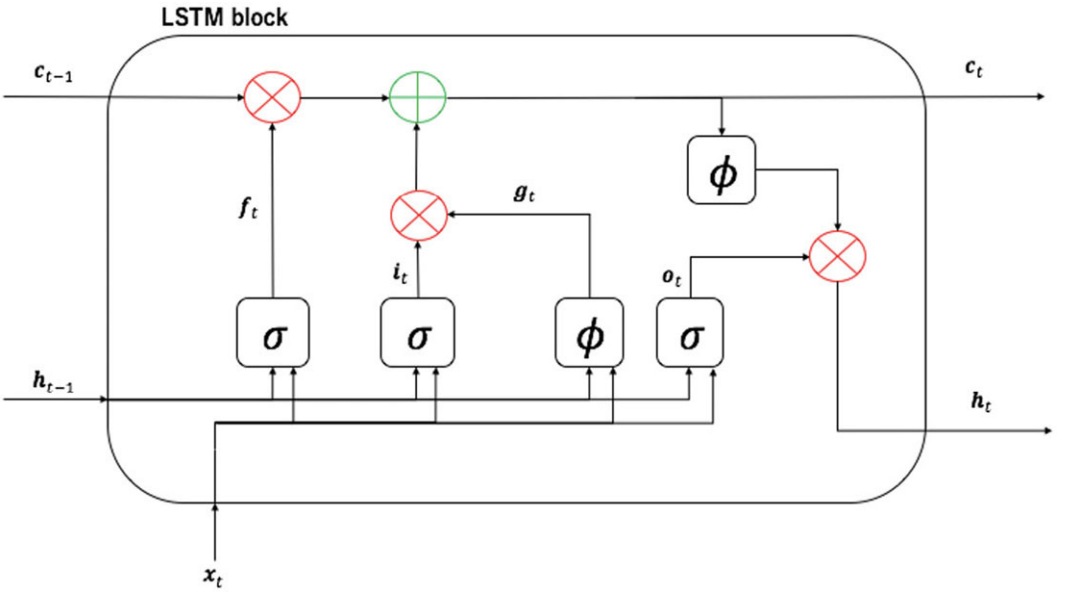


Figure 2. *LSTM architecture ($\bigoplus$: element-wise sum, $\bigotimes$ element-wise product).*

memory cells as past information in addition to the output of the previous block; thus, it can learn long-term memory in addition to short-term memory. The LSTM block, shown in Figure 2, has the input gate $\boldsymbol{i}_t$ and the output gate $\boldsymbol{o}_t$ with the sigmoidal activation function σ to deal with the weight conflict problem caused by sharing weight matrices and bias vectors over time. Further, the LSTM block also has a forget gate $\boldsymbol{f}_t$ to control the memory cell $\boldsymbol{c}_t$ for appropriate long-term memory learning. The k-dimensional (i.e., k neurons) LSTM block at time t that receives the input data $\boldsymbol{x}_t \in \mathbb{R}^d$ and $(\boldsymbol{h}_{t-1}, \boldsymbol{c}_{t-1})$ from the previous LSTM block calculates the outputs of the forget gate $\boldsymbol{f}_t$, the input gate $\boldsymbol{i}_t$, the output gate $\boldsymbol{i}_t$, and the function $\boldsymbol{g}_t$ for the memory cell as:

$$\begin{aligned}
\boldsymbol{f}_t &= \sigma\left(W_f \boldsymbol{x}_t + U_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right), \\
\boldsymbol{o}_t &= \sigma\left(W_o \boldsymbol{x}_t + U_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right), \\
\boldsymbol{i}_t &= \sigma\left(W_i \boldsymbol{x}_t + U_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right), \\
\boldsymbol{g}_t &= \phi\left(W_g \boldsymbol{x}_t + U_g \boldsymbol{h}_{t-1} + \boldsymbol{b}_g\right),
\end{aligned} \tag{3.4}$$

using the bias vectors $(b_f, b_o, b_i, b_g \in \mathbb{R}^k)$ and the weight matrices $(W_f, W_o, W_i, W_g \in \mathbb{R}^{k \times d}$; $U_f, U_o, U_i, U_g \in \mathbb{R}^{k \times k})$ for each gate. The hyperbolic tangent function (tanh) is typically used for the activation function ϕ. Thereafter, the LSTM output $\boldsymbol{h}_t$ and the memory cell $\boldsymbol{c}_t$ at time t, which are inputs to the next LSTM block, are given by:

$$\begin{aligned}
\boldsymbol{c}_t &= \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{g}_t \odot \boldsymbol{i}_t, \\
\boldsymbol{h}_t &= \phi(\boldsymbol{c}_t) \odot \boldsymbol{o}_t
\end{aligned} \tag{3.5}$$

where $\odot$ denotes the Hadamard product.

4. Proposed model

To cope with the limited training data on observable tenors and the need for ultra-long-term extrapolation, we address this problem by increasing the amount of training data in the direction of the observation

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

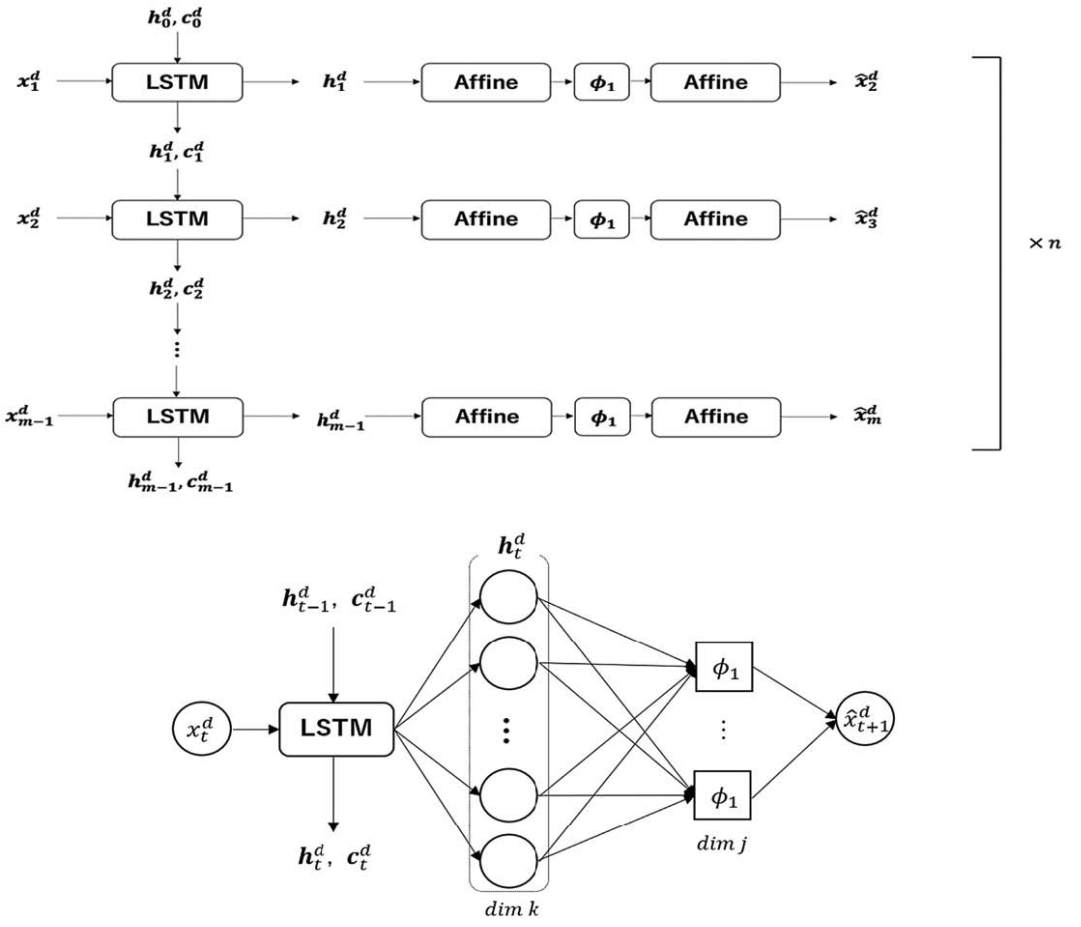


Figure 3. *Network architecture at the global and local levels.*

date and applying LSTM to the data. The LSTM is chosen because of the need to retain the memory of the observed data in ultra-long-term extrapolation. The network architecture used in the training phase is shown in Figure 3.

Let x_t^d be the interest rate data for observable tenors $t = 1, 2, \ldots, m$ and observation date $d = 1, 2, \ldots, n$ (i.e., the past n days from the valuation date $d = 1$). The model is trained with observed yield curve data for the past n days (e.g., $n = 20$ in the following sections) for each valuation date to obtain generalization performance. For each d, the k-dimensional LSTM block with the network parameter φ commonly used for $d = 1, 2, \ldots, n$ is recursively applied as follows:

$$\begin{aligned}
\left(\boldsymbol{h}_1^d, \boldsymbol{c}_1^d\right) &= LSTM_\varphi\left(\boldsymbol{h}_0^d, \boldsymbol{c}_0^d, x_1^d\right), \\
\left(\boldsymbol{h}_2^d, \boldsymbol{c}_2^d\right) &= LSTM_\varphi\left(\boldsymbol{h}_1^d, \boldsymbol{c}_1^d, x_2^d\right), \\
&\vdots \\
\left(\boldsymbol{h}_m^d, \boldsymbol{c}_m^d\right) &= LSTM_\varphi\left(\boldsymbol{h}_{m-1}^d, \boldsymbol{c}_{m-1}^d, x_m^d\right),
\end{aligned} \tag{4.1}$$

where $\boldsymbol{h}_0^d = \boldsymbol{c}_0^d = (0, , , 0) \in \mathbb{R}^k$ and $\varphi = \left(W_f, W_o, W_i, W_g,\ U_f, U_o, U_i, U_g, b_f, b_o, b_i, b_g\right)$.

The reconstructed data $\hat{x}_t^d$ is given by:

$$\hat{x}_t^d = W_2\phi_1\left(W_1\boldsymbol{h}_{t-1}^{d\prime} + \boldsymbol{b}_1{}'\right) + \boldsymbol{b}_2, \tag{4.2}$$

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

where $W_1 \in \mathbb{R}^{j\times k}$ and $\boldsymbol{b}_1 \in \mathbb{R}^j$ denote the weight matrix and the bias vector of the first affine layer to be transformed by an activation function ϕ_1; $W_2 \in \mathbb{R}^j$ and $\boldsymbol{b}_2 \in \mathbb{R}$ denote the weight vector and the bias of the second affine layer. The network parameters at the valuation date, φ and $\theta = (W_1, W_2, \boldsymbol{b}_1, \boldsymbol{b}_2)$, are optimized using the SGD-BPTT to minimize the loss function given by:

$$\sum_{d=1}^{n}\sum_{t=2}^{m}\left(x_t^d - \hat{x}_t^d\right)^2. \tag{4.3}$$

Using the optimized network parameters, the right-hand side of Equation (4.2) can be briefly rewritten as $f_\theta\left(\boldsymbol{h}_{t-1}^1\right)$. Applying $LSTM_\varphi$ and f_θ recursively, the estimated (extrapolated) yield $\hat{y}_t^1$ $(t > m)$ at the valuation date $(d = 1)$ is given by:

$$\begin{aligned}
\hat{y}_{m+1}^1 &= f_\theta\left(\boldsymbol{h}_m^1\right);\ \left(\boldsymbol{h}_m^1, \boldsymbol{c}_m^1\right) = LSTM_\varphi\left(\boldsymbol{h}_{m-1}^1, \boldsymbol{c}_{m-1}^1, x_m^1\right);\\
\hat{y}_{m+2}^1 &= f_\theta\left(\boldsymbol{h}_{m+1}^1\right);\ \left(\boldsymbol{h}_{m+1}^1, \boldsymbol{c}_{m+1}^1\right) = LSTM_\varphi\left(\boldsymbol{h}_m^1, \boldsymbol{c}_m^1, \hat{y}_{m+1}^1\right);
\end{aligned} \tag{4.4}$$

As shown in Equations (3.4) and (3.5), the latent variable vector $\boldsymbol{h}_t$ of the LSTM is the Hadamard product of the outputs of the two bounded activation functions (sigmoid and tanh); thus, $\boldsymbol{h}_t$ is element-wise bounded in $[-1, 1]$. As shown in Equation (4.4), the extrapolated interest rate is obtained by substituting this $\boldsymbol{h}_t$ into f_θ (i.e., Equation 4.2 determined in the training phase), which consists of two affine transformations and an activation function (identity or tanh); thus, unlike naive extrapolation methods such as cubic splines, the output of the model is bounded in an interval determined by the observed training data.

5. Data and calibration

5.1. Data used

We use the yield curve data from major data providers (see Data Sources) to ensure reproducibility for practitioners. Zero-coupon government bond yield curves are not directly observable in the market and are the result of a processing algorithm (i.e., bootstrapping and smoothing), which is not necessarily disclosed by the data provider, applied to observed market data. Thus, a machine learning algorithm is likely to learn the processing algorithm through the yield curve data. However, swap yield curves are directly observable in the market and are free from any processing algorithm on the data. Overnight indexed swaps (overnight interest rate swaps) are becoming more common; however, they still have limited observable tenors. In our numerical analysis, we use the 3m US Dollar LIBOR swap (USSW) and the 6m Japanese Yen LIBOR swap (JYSW), which still allow us to observe the widest range of active tickers for consecutive tenors. The USSW and JYSW are observable for 25- and 30-year tenors in addition to consecutive 1- to 20-year tenors. The swap yields can be converted into zero-coupon swap yields on the consecutive 1- to 20-year tenors. We also use the zero-coupon yield of Japanese government bonds (JGBY), which can be observed in successive maturities ranging from 1 to 30 years. Since our machine learning algorithm requires the observations of interest rates for a long consecutive series of tenors, only USSW, JYSW, and JGBY met the requirements in the vendor data sources we were able to access in this study.

5.2. Calibration procedures

The tenors of the yield curve data are divided into training, validation, and test segments. The segmentation of the tenors depends on the structure of the data. For USSW and JYSW, the validation data consists

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

Table 1. *Validated hyperparameters for JYSW, USSW, and JGBY.*

	JYSW	USSW	JGBY
Learning epochs	5000	15,000	1,000
L2 parameter	0.0001	0.001	0.0001
dim k	20	20	16
dim j	10	10	8
ϕ_1	identity	identity	tanh

of the 25-year swap rate and the test data consists of the 30-year swap rate at the estimation date, which are the only two observable tenors over 20 years. For JGBY, the validation data consists of the consecutive 21- to 25-year spot rates and the test data consists of the consecutive 26- to 30-year spot rates at the estimation date.

Our NN model is trained using yield curves of consecutive 1- to 20-year tenors for the past n days from the estimation date. The parameter n is set to 20 (days) as the number of business days per month throughout this paper.

Then, the hyperparameters, including the number of learning epochs for early stopping, the L2-normalization parameter, the output layer activation function ϕ_1, the LSTM dimension k, and the affine layer dimension j, are selected to minimize the reconstruction MSEs of the validation data at the estimation date, where the MSEs are measured as the average value for 10 random seeds. The validated hyperparameters are shown in Table 1. We employ the rectified adaptive learning rate algorithm for the optimizer. All components of the LSTM block are initialized with a uniformly distributed random variable $\mathrm{U}(-1/\sqrt{k},1/\sqrt{k})$ for the LSTM dimension k; all bias vectors are initialized to zero and the Xavier initialization (Glorot and Bengio, 2010) is used for all matrices.

Finally, the estimation MSEs on the test data at the estimation date are computed as the average value for 10 random seeds to measure the generalization performance of the model.

6. Results

6.1. Benchmark models

We need to select benchmark models to compare with our NN model in terms of generalization performance (i.e., estimation accuracy on the test data) and consistency of yield curve dynamics with observations. While cubic splines are the earliest extrapolation method, as Nelson and Siegel (1987) point out, cubic polynomials are not suitable for yield curve extrapolation because a cubic polynomial in maturity will head off to either plus infinity or minus infinity as maturity increases; on the other hand, as shown in Section 4, our NN model has a bounded output. Assumption-driven approaches, such as the SW method and the constant rate method, are not designed for data reconstruction or estimation, and it is meaningless to compare their estimation accuracy. Although the dynamic extrapolation models have the advantage of providing a confidence interval for extrapolation, they cannot provide the performance metrics, estimation accuracy, and stability, comparable with the proposed model because their outputs are probability distributions. Consequently, the NS and Svensson models, which have the reproducibility of observations and the boundedness of the output value among the existing models, are used as the benchmarks for performance comparison in the following sections. All parameters of the NS and Svensson models are determined simultaneously using the Nelder–Mead optimizer on the zero-coupon training data, keeping the shape parameters $(\lambda, \lambda_1, \lambda_2)$ positive.

For the USSW and JYSW data, the benchmark models are trained with the zero-coupon yield curves converted from the swap yield curves for consecutive maturities from 1 to 20 years on the estimation date, as the swap yield data cannot be fed directly into the benchmark models, unlike the NN model.

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

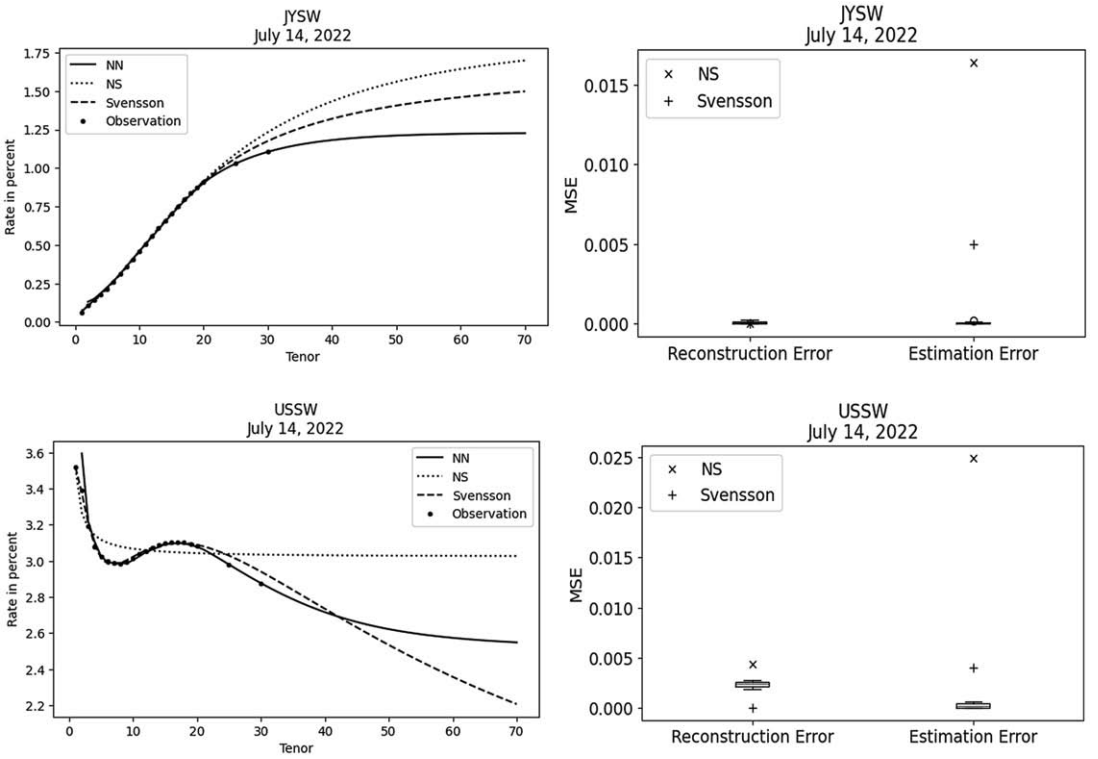


Figure 4. *Reconstruction and extrapolation results for JYSW (top row) and USSW (bottom row) on July 14, 2022.*

After extrapolating to the 30-year maturities with the benchmark models, the reconstructed and estimated zero-coupon yields from 1 to 30 years are converted to par yields and tested against the observed 30-year swap rates. Note that the NS and Svensson models cannot use the 25-year swap yield data for training because the 25-year swap yield cannot be converted to a zero-coupon yield due to the lack of data on swap yields for consecutive tenors from 21 to 24 years.

For the JGBY data, the benchmark models are trained using the zero-coupon yield curves with consecutive maturities ranging from 1 to 25 years on the estimation date. After the extrapolation to 30 years, the estimated zero-coupon yields are tested against the observed zero-coupon yields for the consecutive 26- to 30-year maturities.

6.2. Generalization performance evaluation

6.2.1. Generalization performance on the USSW and JYSW data

This section shows the performance evaluation of the models on the JYSW and USSW data, including the typical examples of the normal and inverted swap yield curves. Looking at the yield curves over the past year from May 2023, the JYSW curves had stable normal yield shapes, whereas the USSW curves had unstable inverted yield shapes.

First, as an example, we compare the performance of the models as of July 14, 2022, when the USSW showed a complicated shape that was difficult to reproduce with the models. Figure 4 (left) shows the comparison between the NN, Svensson, and NS models with the reconstructed and extrapolated swap yield curves, represented by solid, dashed, and fine dashed lines, respectively, and the observations,

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

Table 2. *Twenty-day average of MSE in estimating the 30-year yield of USSW (left) and JYSW (right).*

	USSW(7/14-8/10, 2022)	JYSW(7/14-8/12, 2022)
NS	0.0353942026	0.0227761760
Svensson	0.0012253777	0.0073734591
NN	0.0001882035	0.0000303966

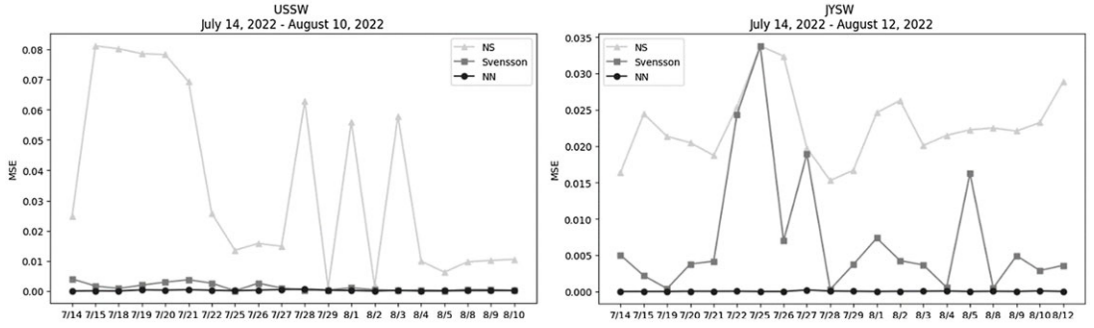


Figure 5. *Daily MSE in estimating the 30-year yield of USSW (left) and JYSW (right).*

represented by round dots, for the JYSW and the USSW on the estimation date. The each NN result is the average of the results for 10 random seeds. Figure 4 (right) shows the comparison of the three models with the MSE in reconstructing the data and estimating the 30-year swap yield on the estimation date, where the NS, Svensson, and NN models are represented by "x," "+," and the box plots, respectively. The reconstruction error is calculated using 2- to 20-year swap rate observations at the estimation date, since the output of the LSTM starts from the 2-year rate.

For both the JYSW and USSW on the estimation date, the NN model has the narrow box plots and shows the best estimation accuracy among the three models, whereas the estimation accuracy of the NS model is the worst. For the USSW on the estimation date, the NS model cannot reproduce the shape of the observed yield curve, and the Svensson model shows a straight declining extrapolation curve converging to $\beta_0 = 0.43\%$ at infinity, whereas the NN extrapolation curve is relatively natural.

Second, we compare the perfomances of the three models over 20 business days starting on July 14, 2022. Each NN result is the average of the results for 10 random seeds. The superiority of the NN model in estimating 30-year yields over 20 days can be confirmed by the 20-day average MSE shown in Table 2 and the daily MSE shown in Figure 5, where the round, square, and triangular dots correspond to the NN, Svensson, and NS results, respectively.

The extrapolations of the JYSW yields and the USSW yields over the 20 days are shown in Figures 6 and 7, respectively, in the same manner as in Figure 4 (left). In Figure 7, the poor representability of the NS model and the straight declining extrapolation curves of the Svensson model are almost identical to those in Figure 4. In Figure 8, for each of the three models, the 20 images from Figures 6 and 7 are overlaid and shown on the right, where the darkest colored dots correspond to the 60-year yields and the lightest colored dots correspond to the 5-year yields for each day; Figure 8 shows that the extrapolated yields from the NN model (bottom row) are more stable than those from the NS and Svensson models (top and middle rows) over the 20 days.

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

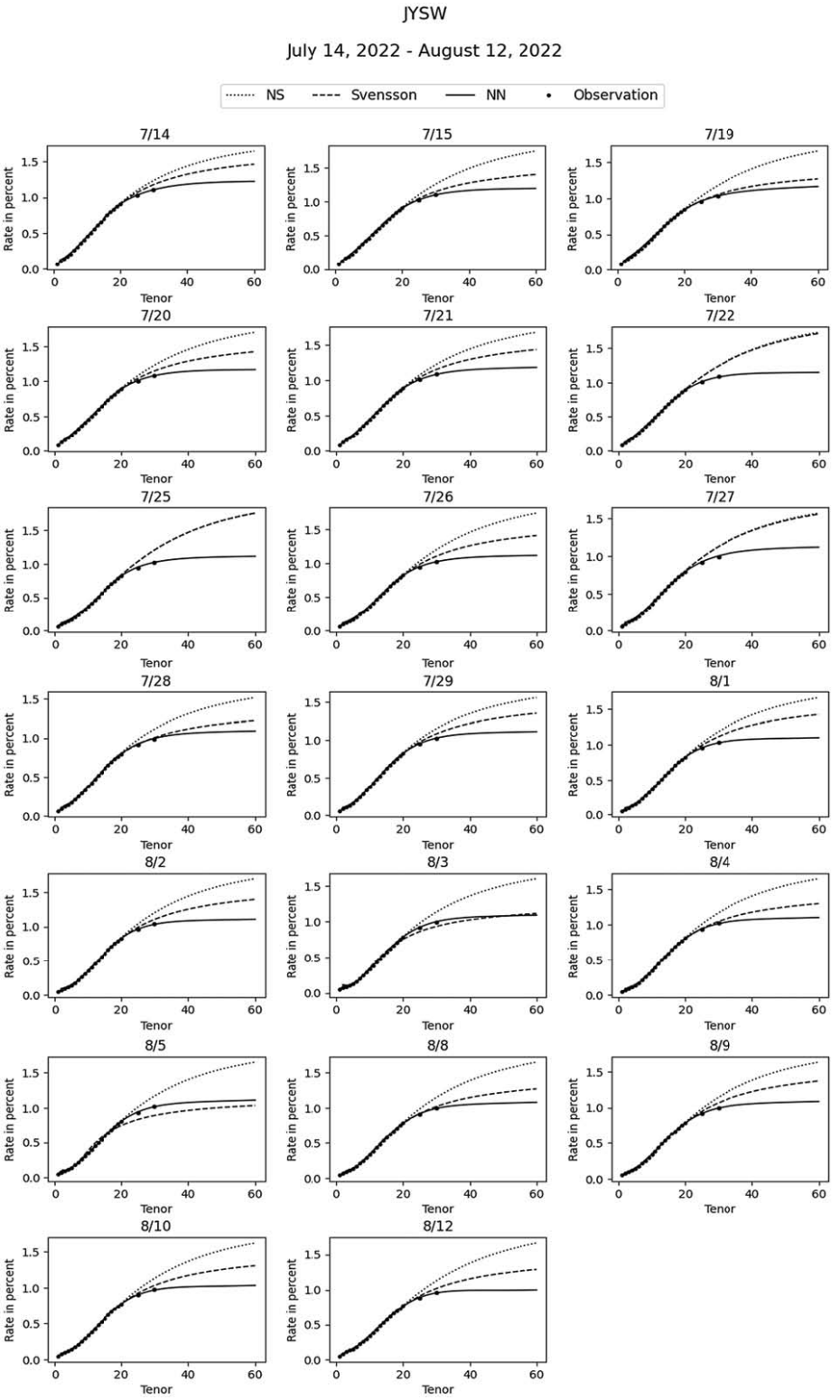


Figure 6. *Extrapolations of JYSW (20 business days from July 14, 2022).*

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

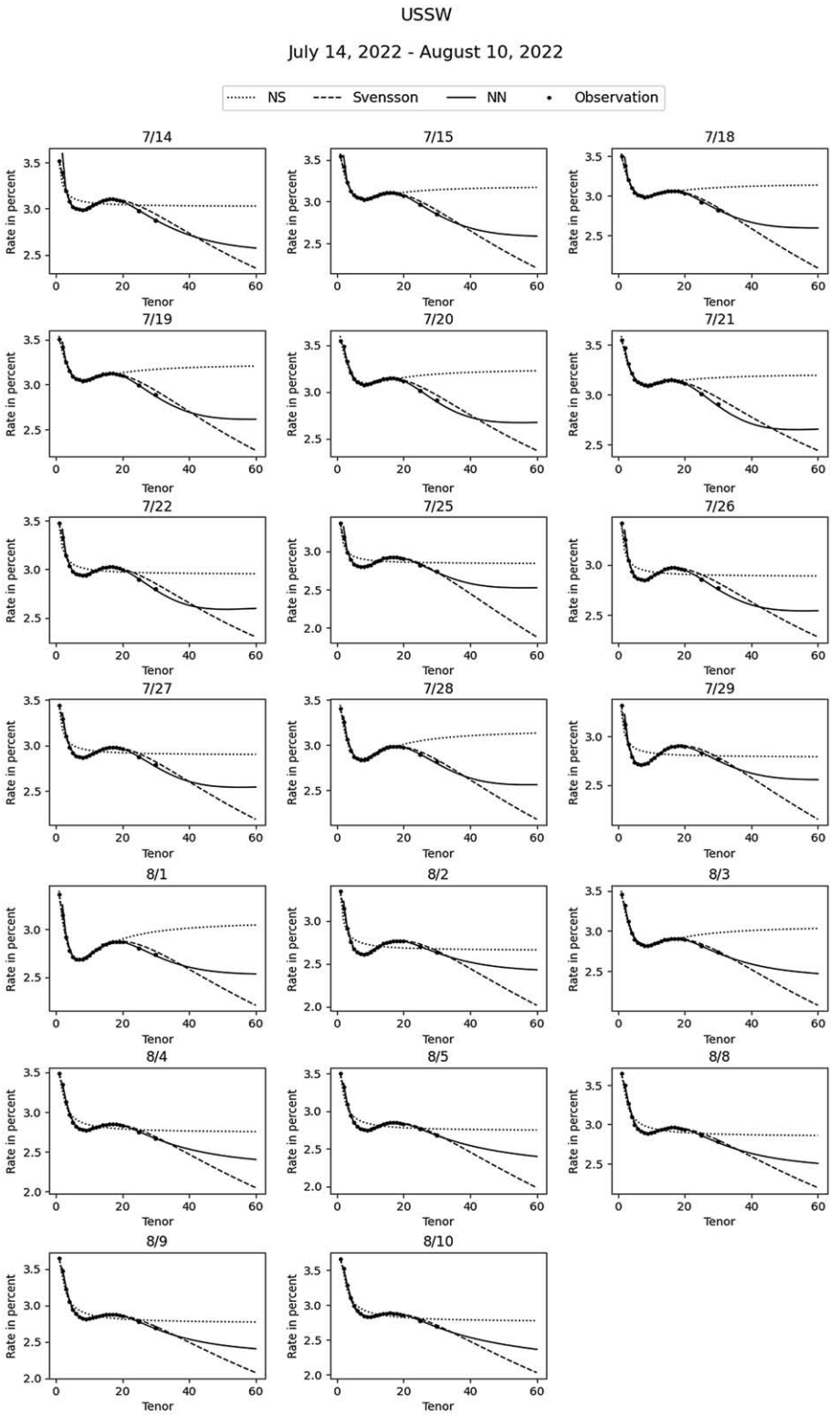


Figure 7. *Extrapolations of USSW (20 business days from July 14, 2022).*

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

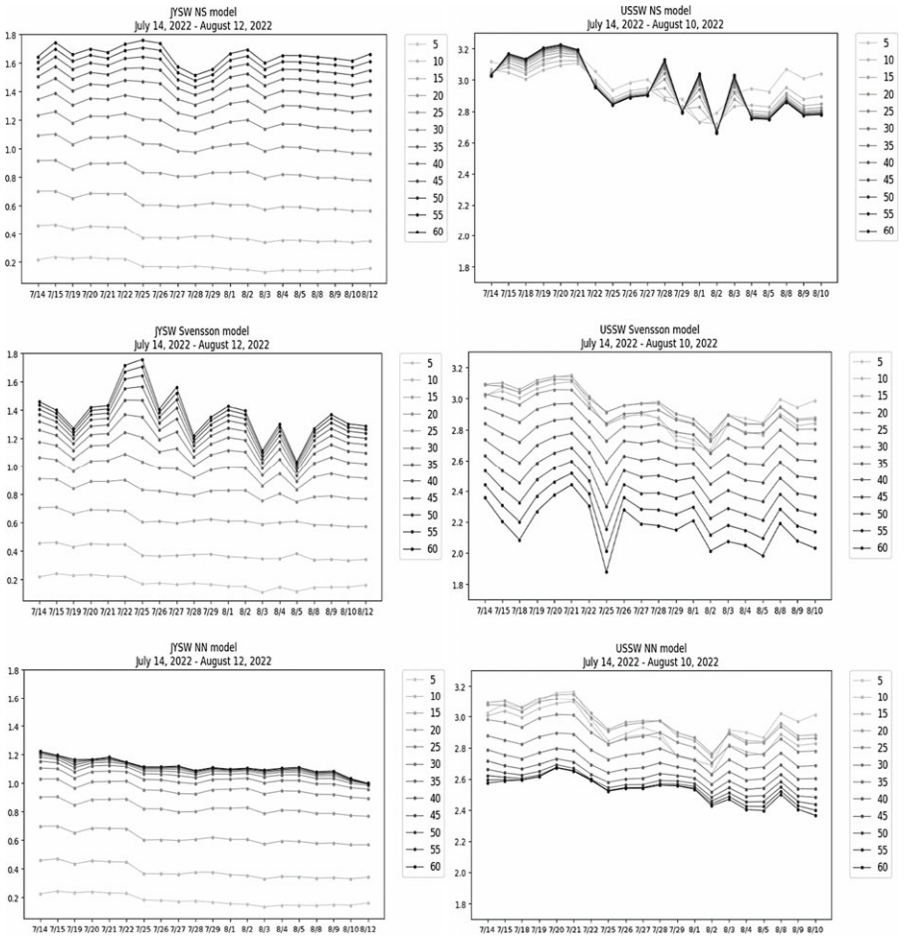


Figure 8. *Daily reconstructed and estimated yields for JYSW (left) and USSW (right) using NS (top row), Svensson (middle row), and NN (bottom row) models.*

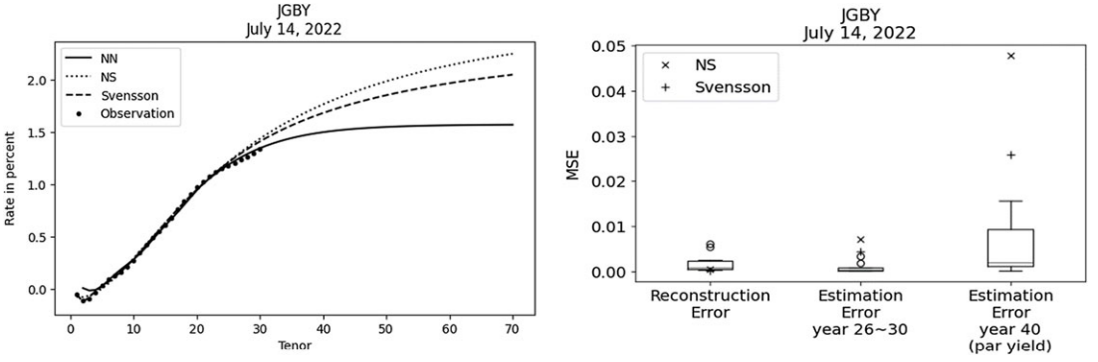


Figure 9. *Reconstruction and extrapolation results for JGBY on July 14, 2022.*

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

Table 3. *Twenty-day average of MSE in estimating 26- to 30-year JGBY and 40-year JGB par yield.*

	JGBY (7/14-8/12, 2022)	
	26–30	40 (par yield)
NS	0.0050600174	0.0350947188
Svensson	0.0021703990	0.0134894456
NN	0.0004831965	0.0010006696

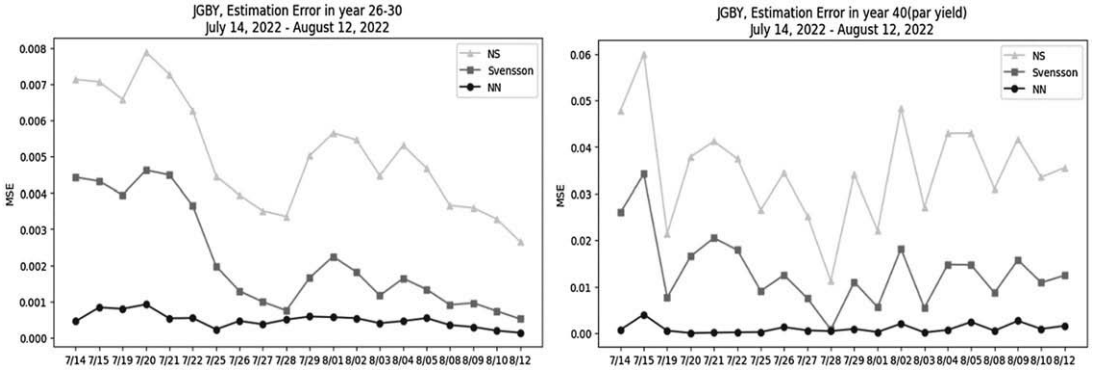


Figure 10. *Daily MSE in estimating 26- to 30-year JGBY (left) and 40-year JGB par yield (right).*

6.2.2. Generalization performance on the JGBY data

This section presents the performance evaluation using the JGBY curves, which have normal yield shapes but are more complex than the JYSW curves.

First, we compare the performance of the models on a single day. Figure 9 shows the comparison between the NS, Svensson, and NN models for the JGBY data as of July 14, 2022, in the same manner as in Figure 4. In contrast to Figure 4, the reconstruction errors are calculated using the JGBY data for maturities of 2–25 years, and the estimation errors are calculated using the JGBY data for maturities of 26–30 years along with the 40-year JGB par yield provided by the Ministry of Finance of Japan. The wide box plot of the NN results of the 40-year par yield is because the par yield is not directly estimable and is calculated from zero-coupon estimates for each year up to 40 years. Resultantly, the NN model has the best estimation accuracy for the JGBY data as of July 14, 2022.

Second, we compare the performance of the three models over 20 consecutive business days starting from July 14, 2022, in the same manner as described in Section 6.2.1. For the 26- to 30-year JGBY and the 40-year JGB par yield, the overall superiority of the estimation accuracy of the NN model within the 20 days can be confirmed by the results oresented in Table 3, and its daily results are shown in Figure 10, in the same manner as in Figure 5.

The extrapolations of the JGBY curves over the 20 days are shown in Figure 11, in the same manner as in Figure 9. For the JGBY curves, the tendency to obtain low extrapolated results in the order of the NS, Svensson, and NN models is stable. In Figure 12, the 20 images from Figure 11 are overlaid and shown on the right for each of the three models, in the same manner as in Figure 8; Figure 12 shows that the extrapolated yields from the NN model (bottom) are more stable than those from the NS and Svensson models (top and middle) over the 20 days, similar to Figure 8.

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

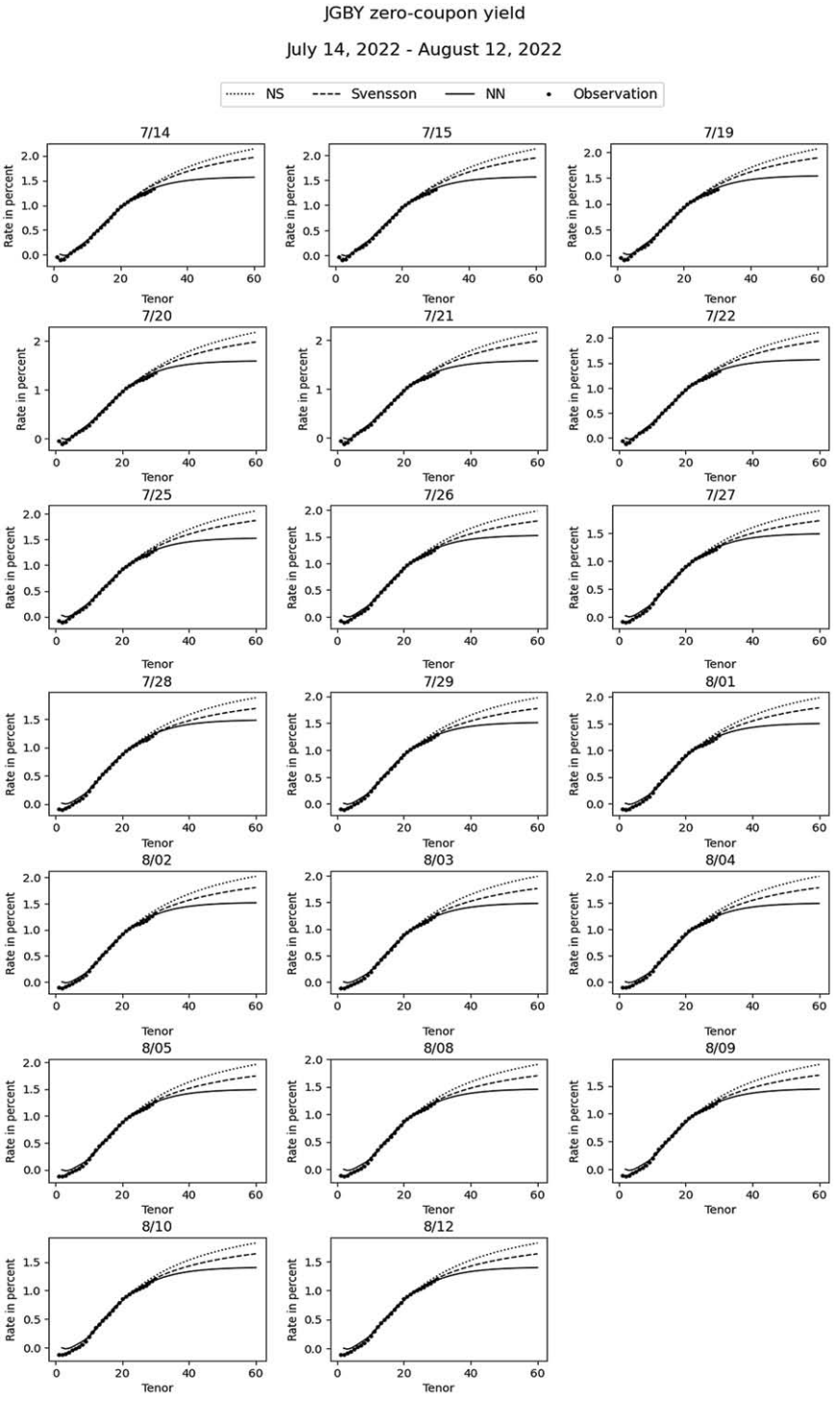


Figure 11. *Extrapolations of JGBY (20 business days from July 14, 2022).*

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

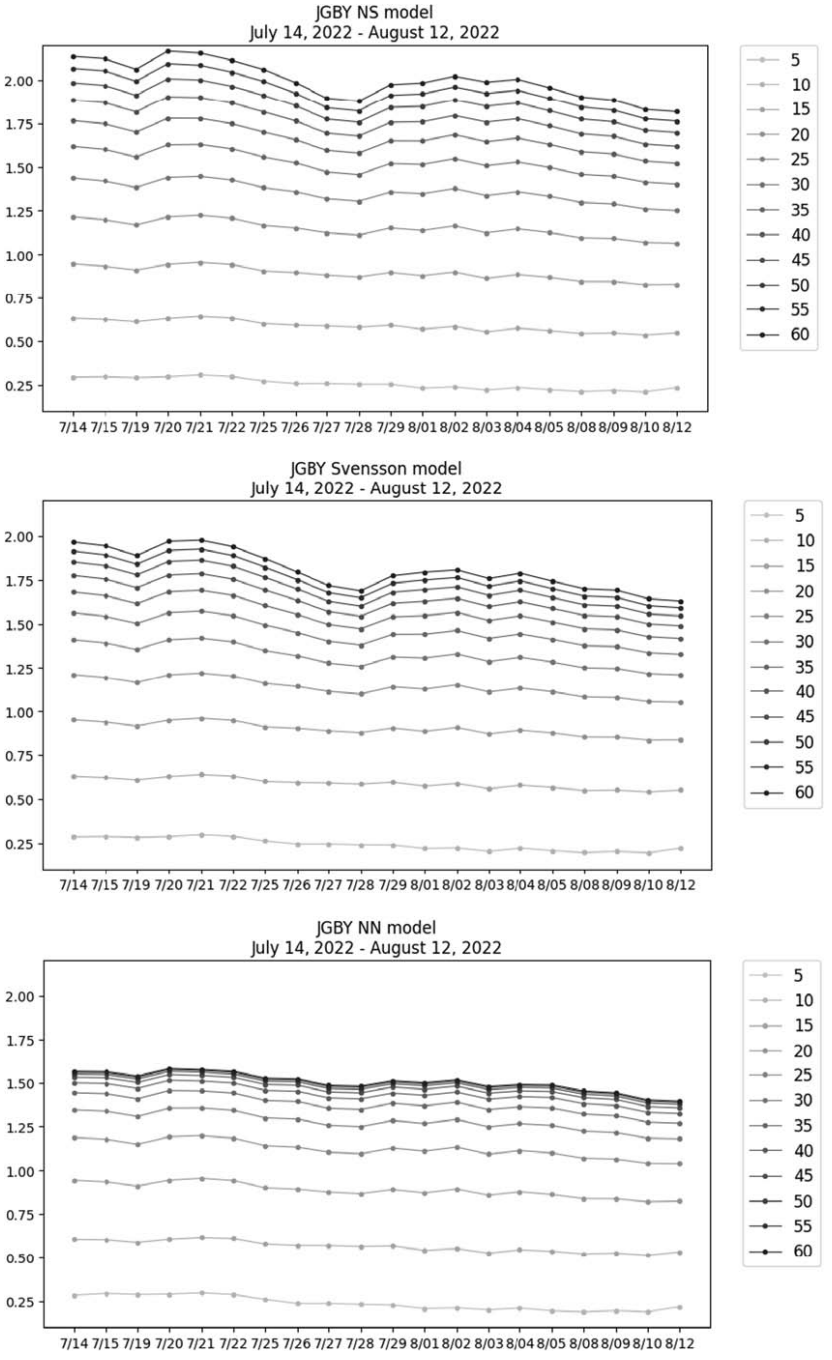


Figure 12. *Daily reconstructed and estimated yields for JGBY using NS (top), Svensson (middle), and NN (bottom) models.*

6.3. Consistency of yield curve dynamics

While Figures 8 and 12 indicate that the extrapolated yields from the NN model are more stable than those from the NS and Svensson models, the stability should be analyzed in terms of consistency with observations. In this section, we examine the consistency of the yield curve dynamics between

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

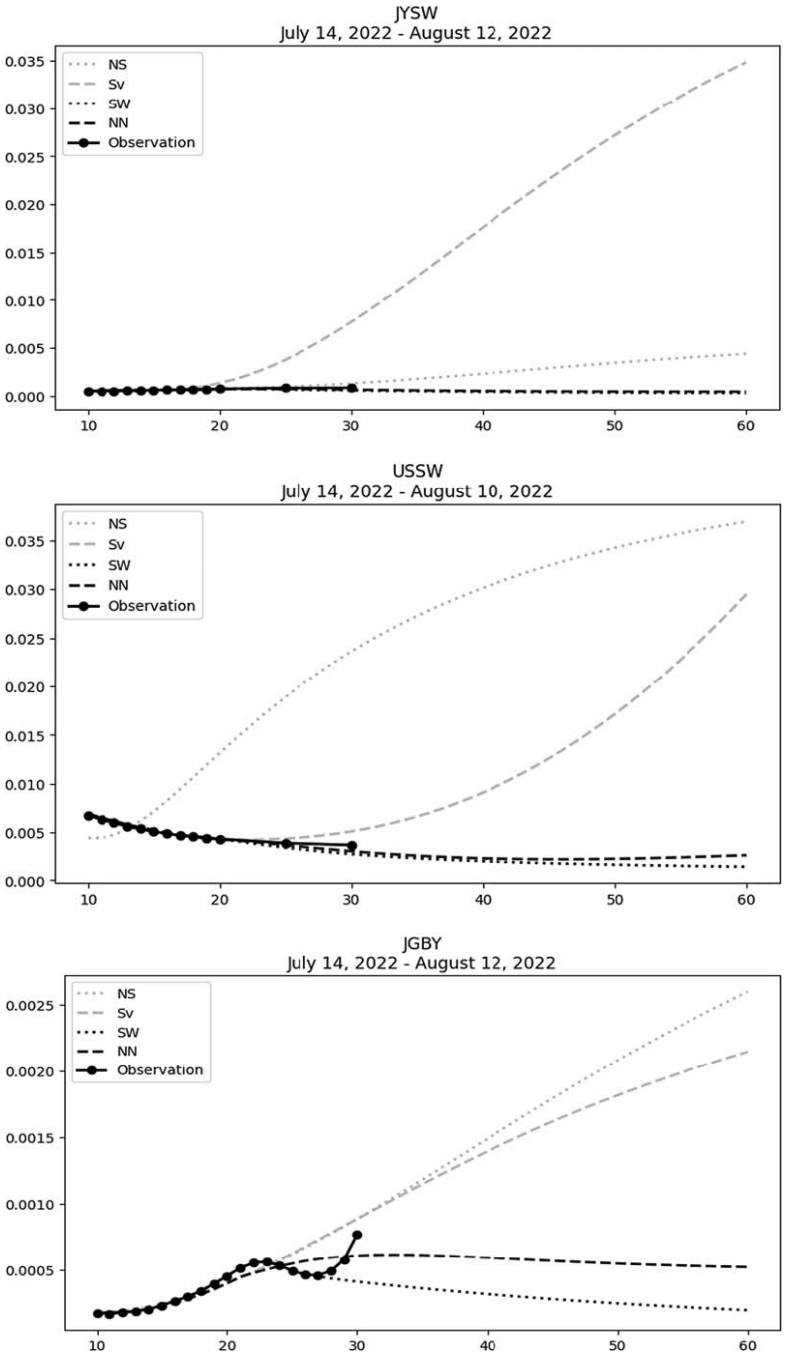


Figure 13. *Variance term structures of daily changes in yields for JYSW, USSW, and JGBY (from top to bottom) over the 20 business days starting July 14, 2022.*

observation and extrapolation, including the SW method in the benchmarks. The setup of the SW method here is that the convergence parameter is estimated by finding the smallest value such that the difference between the forward rate at the convergence maturity (60 years) and the UFR (3.5%) is smaller than 1bps, using the same extrapolation starting point as the other parametric benchmark models. The yield

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

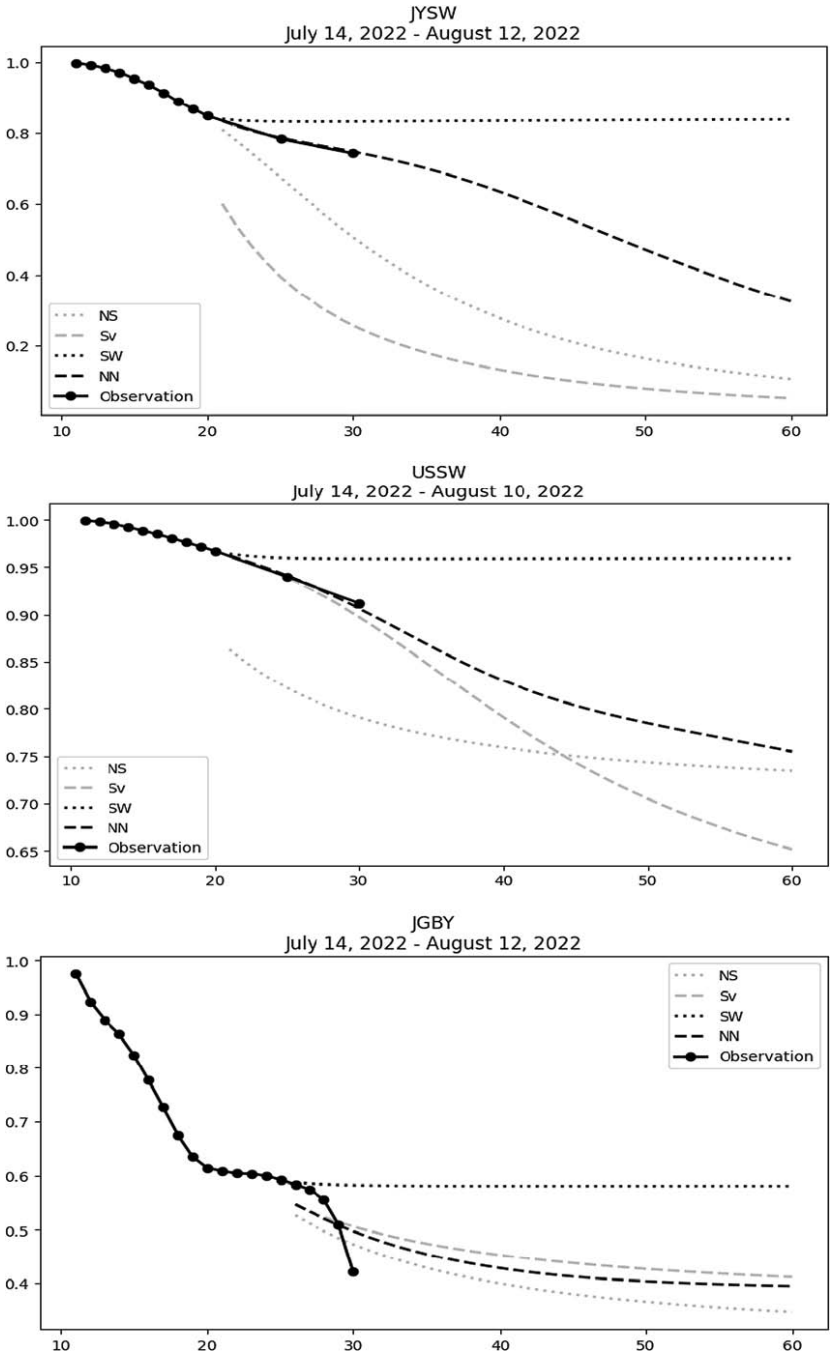


Figure 14. *Correlation term structures with daily changes in 10-year yields for JYSW, USSW, and JGBY (from top to bottom) over the 20 business days starting July 14, 2022.*

curve dynamics can be expressed as the term structure of a volatility metric by tenor, and the consistency between the observed dynamics and the extrapolated dynamics can be observed by the similarity of the metric values between them on the test data. In order to focus on the dynamics of the term structure of yields, this section analyzes the daily changes in yields by tenor rather than the yields themselves,

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

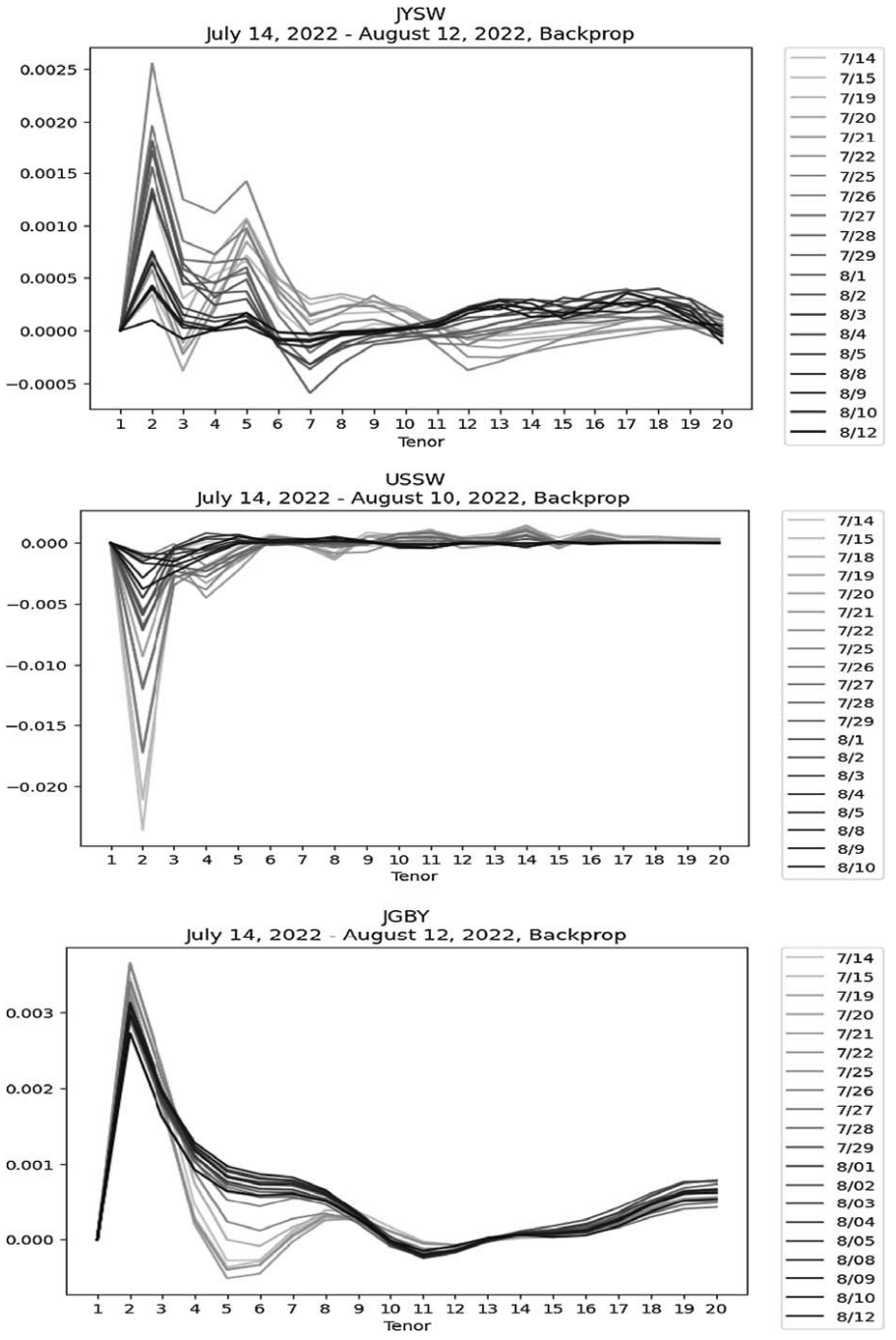


Figure 15. *Input relevance by tenor for JYSW, USSW, and JGBY (from top to bottom) on each day over the training period.*

since, for example, the variance of yields does not reflect any time order in the data (i.e., it is invariant to changing time order in the data).

Figure 13 compares the term structure of the 20-day variances of daily changes in yields from the NN model (dashed line), SW model (dotted line), Svensson model (light dashed line), NS model (light dotted

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

line), and observations (solid line) for the JYSW, USSW, and JGBY data; while the term structure of the variances from the NN and SW models are relatively flat and relatively consistent with the observations, those from the NS and Svensson models rise unnaturally with tenor.

However, the yield curve dynamics should be assessed not only by tenor-by-tenor variances but also by inter-tenor correlations. The inter-tenor correlations between the extrapolated yield and the observed yield may be of interest to those seeking to somehow hedge the unobservable tenor interest rate risk embedded in actuarial liabilities. The correlation between the distribution of daily changes in each tenor yield, including extrapolation, over the training period and that of the 10-year yield is shown in Figure 14 in the same manner as in Figure 13. In contrast to Figure 13, the term structure of the inter-tenor correlations from the SW method is too flat and clearly loses consistency with the observations.

Although the NN model, like the parametric models, has limited consistency with the complicated dynamics of the JGBY under the Bank of Japan's yield curve control policy, overall the NN model produces the most natural yield curve dynamics among the four models in terms of the consistency with observations.

7. Model interpretability

NN models in general have the limitation that the relationship between inputs and outputs is not transparent, but our NN model can provide model interpretability as the gradients of outputs with respect to inputs using the backpropagation algorithm given by the backward function in the Autograd package of PyTorch. For the input training data consisting of successive 1- to 20-year tenor yields at each day (d) over the training period, we define an input relevance metric as the sum of the gradients of outputs with respect to inputs given by:

$$\sum_{t=21}^{60} \frac{\partial \hat{y}_t^d}{\partial x_1^d}, \sum_{t=21}^{60} \frac{\partial \hat{y}_t^d}{\partial x_2^d}, \ldots, \sum_{t=21}^{60} \frac{\partial \hat{y}_t^d}{\partial x_{20}^d}. \tag{7.1}$$

For the JYSW, USSW, and JGBY data, the input relevance metric, as the averaged value for 10 random seeds, at each day over the training period is shown in Figure 15, with the color of the lines are darkening as the end of the training period approaches. It is likely that the 2-year yields have a significant impact on the distinction between normal and inverted yields for extrapolated tenors.

8. Conclusions

This study argues that NNs, which have recently received considerable attention in mortality forecasting, are useful for yield curve extrapolation, where they have not been used before. By extending the dimensionality and nonlinearity of the model without any exogenous assumptions, our extrapolation method achieves better generalization performance, stability, and consistency with observed yield curve dynamics than previous parametric models for the USSW, JYSW, and JGBY data. A higher generalization performance (i.e., a higher estimation accuracy on the test data) of the model does not necessarily imply a higher certainty of extrapolated yields far beyond the LOT, but a lower risk of overfitting to the observed data, which is a key requirement for models with data reproducibility to achieve consistency with observed yield curve dynamics. Hhowever, this study is limited by the narrow range of data used. Expanding the data availability of the model is a future challenge, as our model requires yield curve data with 20 consecutive tenors and more.

Since the limited training data in the direction of the tenor axis is an obstacle for NN-based yield curve extrapolation, we addressed this issue by increasing the amount of training data in the direction of the observation date and using the LSTM architecture to retain the memory of observed data in the long-term extrapolation. While the simple LSTM architecture contributes to output stability and backpropagation-based interpretability, achieving NN-based yield curve extrapolation with architectural interpretability

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press

by extracting interpretable feature values without losing generalization performance and stability is also a topic for future work.

Acknowledgments. This research is supported by Numerical Technologies Inc. and the Institute of Science and Technology, Meiji University.

Data sources. The data source for the JYSW and USSW was Bloomberg and that for JGBY was FactSet; permission to use the data in this paper was granted by the data providrers. The 40-year JGB par yield was obtained from the open source provided by the Ministry of Finance of Japan.

References

Balter, A., Shotman, P. and Pelsser, A. (2013) Extrapolating the term structure of interest rate with parameter uncertainty. arXiv: 1312.5073v1

Cybenko, G. (1989) Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, **2**, 303–314.

Diebold, F.X. and Li, C. (2006) Forecasting the term structure of government bond yields. *Journal of Economics*, **130**(2006), 337–364.

Elman, J.L. (1990) Finding structure in time. *Cognitive Science*, **14**(2), 179–211.

Glorot, X. and Bengio, Y. (2010) Understanding the difficulty of training deep feedforward neural network. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. Society for Artificial Intelligence and Statistics.

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

Jørgensen, P.L. (2018) An analysis of the Solvency II regulatory framework's Smith-Wilson model for the term structure of risk-free interest rates. *Journal of Banking and Finance*, **97**, 219–237

Kauffmann, P.C., Takada, H.H., Terada, A.T. and Stern, J.M. (2022) Learning forecast-efficient yield curve factor decompositions with neural networks. *Econometrics*, **10**(2), 15.

Kort, J. and Vellekoop, M. (2016) Term structure extrapolation and asymptotic forward rates. *Insurance: Mathematics and Economics*, **67**, 107–119.

Lecun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E. and Jackel, L.D. (1990) Handwritten digit recognition with a back-propagation network. In *Neural Information Processing Systems* 2 (NIPS 1989), pp. 396–404.

Leiser, B. and Kerbeshian, J. (2019) Methodologies for Valuing Cash Flows That Extend Beyond the Maximum Yield Curve. SOCIETY OF ACTUARIES, Article from Risk Management, September 2019, Issue 45.

Nelson, C. and Siegel, A. (1987) Parsimonious modeling of yield curves. *Journal of Business*, **60**(4), 473–489.

Numes, M., Gerding, E. and Mcgroarty, F. (2019) A comparison of multitask and single task learning with artificial neural networks for yield curve forecasting. *Expert Systems with Applications*, **119**, 362–375

Signorelli, T.P., Campani, C.H. and Neves, C. (2022) Extrapolating long-run yield curves: an innovative and consistent approach. *North American Actuarial Journal*, published online: 13 Sept, 2022. doi: 10.1080/10920277.2022.2102040

Smith, A. and Wilson, T. (2001) Fitting yield curves with long term constraints. Research Notes. Bacon and Woodrow.

Svensson, L. (1994) Estimating and interpreting forward interest rates: Sweden 1992–1994. Working Paper No. 4871. National Bureau of Economic Research.

Wüthrich, M.V. and Merz, M. (2022) Statistical foundations of actuarial learning and its applications. Available at SSRN id=3822407.

https://doi.org/10.1017/asb.2024.27 Published online by Cambridge University Press