Yield curve forecasting using the Nelson-Siegel model. A comparison of ARIMA, VAR and Random Forest approaches – evidence from the USA

Maciej Marcin Olszewski[a]

Abstract. The aim of this article is to compare different approaches to forecasting the US yield curve factors derived using the Nelson-Siegel (NS) model. Using daily US swap yield data from 1990 to 2026, we assess the Autoregressive Integrated Moving Average (ARIMA), Vector Autoregression (VAR) and Random Forest (RF) models in a 1-day-ahead and 1- to 20-day-ahead forecasting competition. The principal finding of this study is that ARIMA significantly outperforms the RF in forecasting the NS model factors, as does VAR, although only in terms of the Level and Curvature factors. The results of this study thus suggest that the use of machine learning methods, in the case of the US yield curve, is not always superior.

Keywords: yield curve, forecasting, Nelson-Siegel model, machine learning
JEL: C53, C58, E43

1. Introduction

The yield curve is a key element of financial markets which carries information about the state of the economy. For that reason, a good understanding of its dynamics and determinants facilitates decision-making processes in areas such as monetary policy and risk management or when developing trading strategies.

The aim of this article is to compare different approaches to forecasting the US yield curve factors based on the Nelson-Siegel (1987, NS) model, which decomposes the yield curve into three factors: Level (L), Slope (S) and Curvature (C). The Level factor describes the level of long-term interest rates, the Slope factor represents the difference between the level of short- and long-term interest rates, while the Curvature factor is responsible for yield curve convexity.

1.1. Classical approaches

[a] Student at SGH Warsaw School of Economics, Al. Niepodległości 162, 02-554 Warszawa, Poland, e-mail: maciek.m.olszewski@gmail.com, https://orcid.org/0009-0006-2351-8867.

The NS model is widely applied due to its simplicity and economic interpretability. It can be used to forecast the entire yield curve. As proposed by Diebold and Li (2006), this can be done in two steps: forecasting the NS factors and reconstructing the future shape of the yield curve. There are numerous ways of forecasting the NS factors, including the use of Autoregressive Integrated Moving Average (ARIMA) or Vector Autoregressive (VAR) models (Diebold & Li, 2006) or the state-space model and the Kalman filter (Diebold et al., 2006). Additionally, there are noteworthy arbitrage-free approaches such as the Arbitrage-Free Dynamic Nelson-Siegel model proposed by Christensen, Diebold and Rudebusch (2011).

1.2. Machine learning approaches

Recently, with the growing popularity of machine learning (ML) methods, the yield curve factors are also forecasted using models such as Support Vector Machines (SVM), Group Method of Data Handling (GMDH; Kim et al., 2020) and different variations of neural networks, e.g. Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM; Kim et al., 2020; Richman & Scognamiglio, 2024). Another approach is to use decision tree-based methods, such as Random Forests (Kostyra & Rubaszek, 2020; Puglia & Tucker, 2020; Rayeni & Naderi, 2025), LightGBM (Puglia & Tucker, 2020) or XGBoost (Puglia & Tucker, 2020; Rayeni & Naderi, 2025; Santos Soares, 2025; Zhang, 2024). Decision trees can also be used as an enhancement of the dynamic NS model to identify different interest rate regimes and predict their changes (Bie et al., 2024). Additionally, tree-based methods were successfully applied to yield-curve-derived recession prediction (see Cadahia Delgado et al., 2022).

1.3. Methodological concerns in machine learning forecasts

The advantages of ML models over traditional time-series models are not obvious. While most of the published articles indicate that ML methods deliver more accurate forecasts than traditional benchmarks, it is worth noting that this might result from publication bias, when only the results indicating that ML methods are successful in forecasting are accepted for publication or data leakage in the design of the forecasting competition (see Hewamalage et al., 2022 for a general discussion on data leakage in the context of time-series forecasting). Puglia and Tucker (2020), who worked on US Treasury data, showed that the performance of ML methods in yield curve forecasting depends heavily on the choice of the training and cross-validation samples. As stated by the authors, ‘strategies which eliminate data “peeking” produce lower, and perhaps more realistic, estimates of forecast accuracy’ (Puglia & Tucker, 2020, p.

2). Rubaszek and Sznajderska (2026), who discuss the topic of data leakage in exchange rate forecasting, show that XGBoost models produce significantly more accurate forecasts than the random walk benchmark only when data leakage is allowed.

Considering the above, the aim of this article is to explore the suitability of the random forest framework in forecasting NS factors extracted from the US swap yield curve in the years 1990–2026. For that purpose, we estimate the ARIMA, VAR and Random Forest (RF) models for each of the NS model factors and evaluate the next-day forecasts. We also check the predictive content of selected financial variables.

2. Methodology

2.1. Nelson-Siegel model

Nelson and Siegel (1987) proposed a parsimonious model that allows the reproduction of the commonly observed shapes of yield curves. They proposed the following functional form:

$$R_m = L + S\left(\frac{1-e^{-m\lambda}}{m\lambda}\right) + C\left(\frac{1-e^{-m\lambda}}{m\lambda} - e^{-m\lambda}\right), \quad (1)$$

where:

R_m is the yield for maturity m,

L is the parameter for the long-term level of the interest rates,

S is the parameter responsible for the slope of the yield curve,

C is the parameter that affects the curvature of the yield curve,

λ is the parameter responsible for the shape of latent factors (see Figure 1).

As explained by Rubaszek (2012), the formula above can be derived by replacing the equation for the instantaneous forward rate (F_m):

$$F_m = L + Se^{-m\lambda} + C(m\lambda \times e^{-m\lambda}) \quad (2)$$

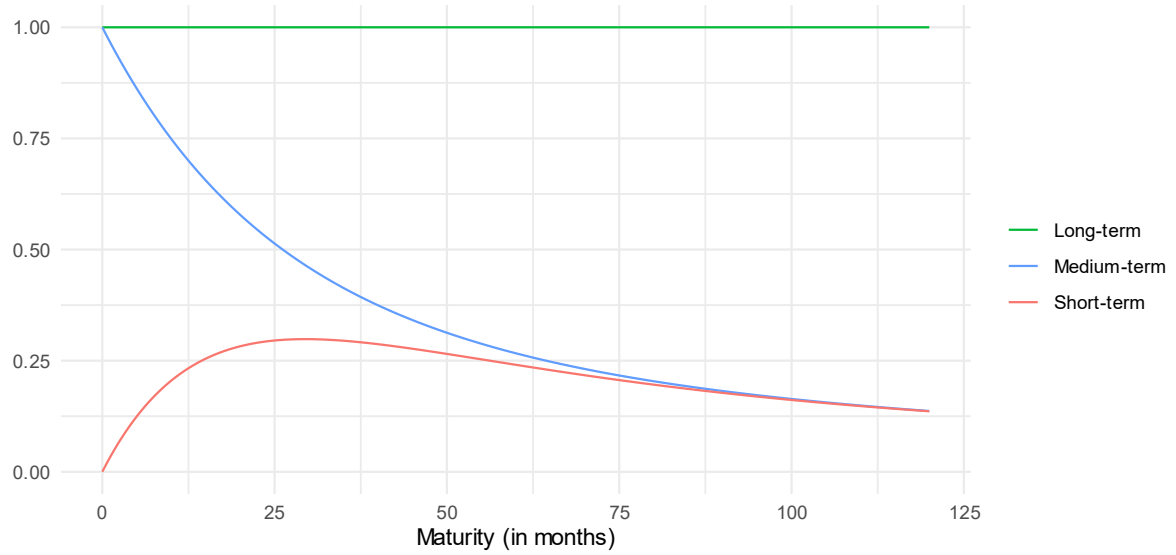
with the definition of the spot rate:

$$R_m = \frac{1}{m}\int_0^m F_s ds. \quad (3)$$

To estimate the values of the NS factors, one can fix the λ parameter and use ordinary least squares (OLS) to derive the L, S and C factors. In this article, we follow Diebold and Li (2006), where $\lambda = 0.0609$ so that the peak of the Curvature factor is at a 30-month horizon (see Figure

1). This method also allows the comparability of NS factor estimates across different dates, as their interpretation depends on the value of λ.

Figure 1. Components of the spot rate



Source: author's calculations.

2.2. ARIMA

The first model used in forecasting NS latent factors is a well-known ARIMA (p, d, q) model of the following form:

$$\left(1 - \varphi_1 L - \varphi_2 L^2 - \ \ldots \ - \varphi_p L^p\right)\Delta^d y_t = \alpha_0 + \left(1 + \theta_1 L + \theta_2 L^2 + \ \ldots \ + \theta_q L^q\right)\varepsilon_t, \qquad (4)$$

where:

y_t is the dependent variable,

L is the lag operator,

Δ is the differencing operator,

p is the order of the autoregressive process,

d is the order of the differencing of the dependent variable,

q is the order of the moving average process,

$\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$ is the error term.

This model is described in detail in Hamilton (1994, pp. 43–71).

2.3. VAR

The VAR model, introduced by Christopher A. Sims (1980), represents a data-driven approach. Its main theoretical advantage over ARIMA is that it allows interactions between the variables in the equation system. The basic specification is given by:

$$\boldsymbol{y_t} = \boldsymbol{c} + \boldsymbol{\Phi_1 y_{t-1}} + \boldsymbol{\Phi_2 y_{t-2}} + \cdots + \boldsymbol{\Phi_p y_{t-p}} + \boldsymbol{\varepsilon_t}, \tag{5}$$

where:

$\boldsymbol{y_t}$ is a $(n \times 1)$ vector containing n variables in period t,
$\boldsymbol{\Phi_p}$ is the matrix of coefficients corresponding to the vector of p lagged values of the dependent variables,
$\boldsymbol{\varepsilon_t} \sim \boldsymbol{\mathcal{N}(0, \Omega)}$ is the error term.

This model is discussed in Hamilton (1994, pp. 291–350).

2.4. RF

The RF is the third model used in the forecasting competition of NS latent factors. It is an ensemble method introduced by Breiman (2001). Its main objective is to combine predictions of many de-correlated regression trees via bagging (bootstrap aggregating).

As described by James et al. (2021), the construction of a single regression tree for a response variable, in our case latent factor y_t, using a set of predictors $x_{1t}, x_{2t}, \dots, x_{pt}$ relies on partitioning the p-dimensional predictor space into J distinct and non-overlapping regions $R_1, R_2, \dots, R_J$. For each observation from the training sample, we assign it to given region R_i and calculate the predicted value of the response variable as the mean of all observations from R_i. The algorithm for creating J regions is called recursive binary splitting and consists of a loop in which one predictor variable is chosen and its value is used to divide the predictor space into two. The loop ends when a stopping criterion is reached. This criterion may rely on reaching a given (low) number of observations in each node or reaching the maximum tree depth. The algorithm for selecting the best split is based on finding the variable and its cut point value that minimises the following expression:

$$\sum_{t: x_{jt} \in R_1(j,s)} \left(y_t - \hat{y}_{R_1}\right)^2 + \sum_{t: x_{jt} \in R_2(j,s)} \left(y_t - \hat{y}_{R_2}\right)^2, \tag{6}$$

where $\hat{y}_{R_1}, \hat{y}_{R_2}$ are the means of the response variable in two regions created after the split across variable x_j and cut point s.

One of the drawbacks of a single regression tree is its relatively high variance. The results we obtain from a regression tree are very sample-sensitive. James et al. (2021) provide an explanation that e.g. splitting the dataset into two parts and fitting separate regression trees to both halves will produce two quite different trees, unlike in a low variance procedure such as linear regression, where the results would be somewhat similar (in cases where the number of observations is much higher than the number of regressors). The solution to this problem relies on growing many independent trees and then averaging their predictions. To ensure that the trees remain independent, two kinds of solutions are applied. The first one is bagging, which involves growing trees on bootstrapped (different) samples. The second one deals with strong predictors that the algorithm selects repeatedly in every tree, in which case the resulting ensemble of trees is usually highly correlated. Here, bagging decreases the variance to a limited extent. Therefore, the number of candidate predictors for each split is restricted so that the algorithm will not be allowed to continuously use the strongest predictors, thus returning less correlated trees.

The RF framework above was developed for cross-section regression rather than time-series predictions. For that reason, one of the assumptions of RFs is that the realisations of the response variable are independent and identically distributed (i.i.d.). This is clearly at odds with many time-series observations, which are characterised by serial correlation. Thus, in this article, we use an extension of the RF approach for time-series data proposed by Goehry et al. (2023). The authors suggested a modification to the bootstrapping algorithm that draws blocks of consecutive observations rather than observations from the whole dataset. Their numerical experiments proved that the moving block bootstrap is the best choice for selecting these blocks for the bootstrap, regardless of the values of other hyperparameters. Its mechanism is straightforward and involves drawing blocks of consecutive observations of a predefined length.

3. Empirical Analysis

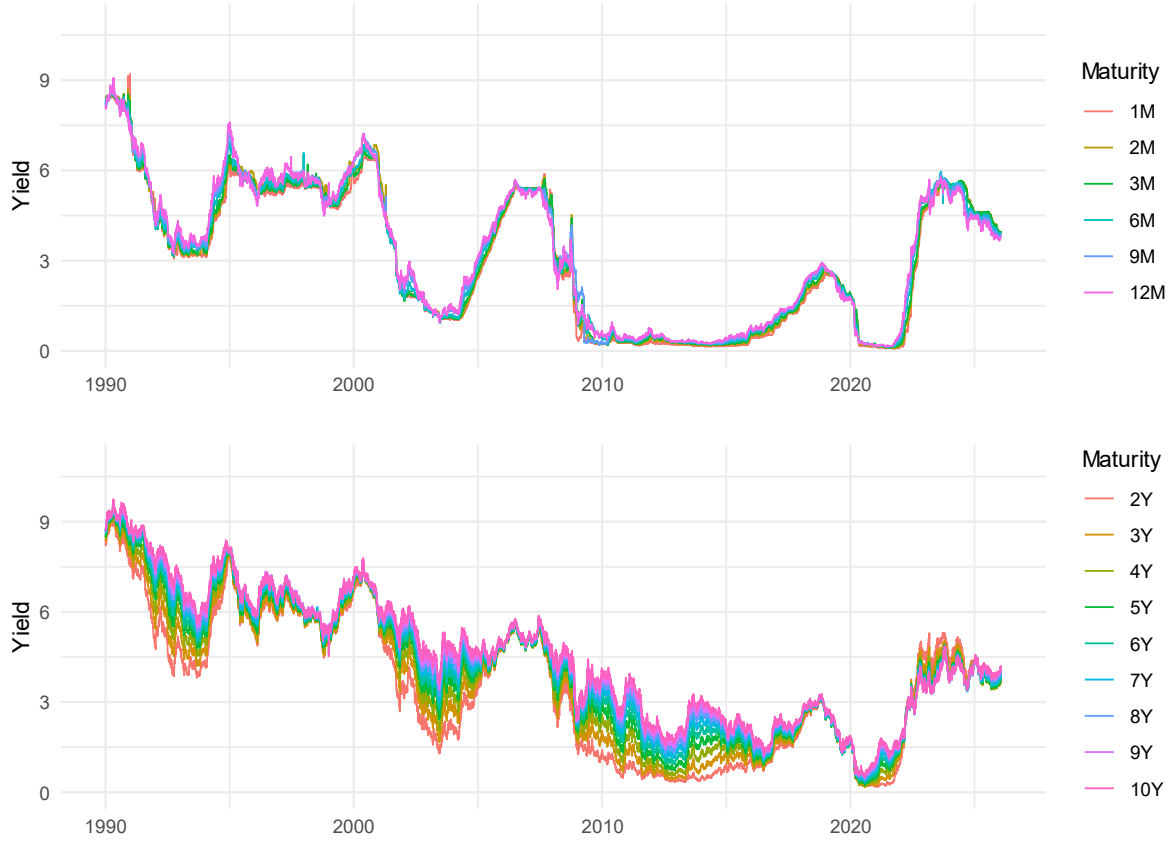
3.1. Data

The analysis covers daily United States swap yield curve data from January 1990 to January 2026, obtained from Refinitiv Workspace. The dataset contains yields for the following

maturities: 1M, 2M, 3M, 6M, 9M, 1Y, 2Y, 3Y, 4Y, 5Y, 6Y, 7Y, 8Y, 9Y, and 10Y. Swap rates are used rather than government bond yields, as they are derived from actual daily market transactions and do not require interpolation for missing maturities. Moreover, the swap market is believed to be more liquid.

As the NS model is derived for continuously compounded rates, raw data are transformed using the formula: $R_{m,t} = \ln\left(1 + \frac{r_{m,t}}{100}\right) \cdot 100$. Figure 2 shows how yields of short- and long-term maturities evolved over time.

Figure 2. US swap yields

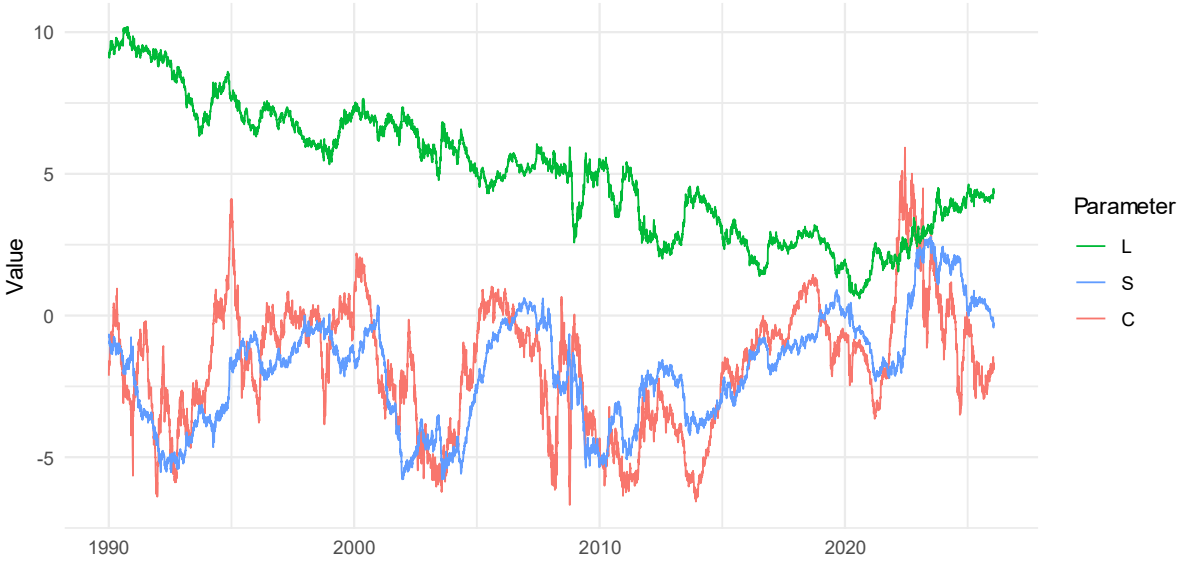


Source: author's calculations based on Refinitiv Workspace data.

In the next step, for each day t, we fitted the NS model to the yield curve. As already mentioned, following Diebold and Li (2006), the value of the λ parameter was fixed at 0.0609, whereas the values of L_t, S_t and C_t were estimated using the OLS estimator. The resulting estimates of the factor loadings are presented in Figure 3. It shows that the long-term level of interest rates decreased gradually up until 2020, followed by a sharp increase. Throughout the research period, the slope of the yield curve fluctuated from high term-premium to nearly none (flat yield

curve), only rarely inverting. In most cases, this situation occurred during financial and/or global crises, mainly in 2001 (dotcom crisis), 2007–2008 (global economic crisis), 2020 (COVID-19 pandemic) and following 2022 (Russian invasion on Ukraine). C_t exhibited primarily negative values throughout the analysed period. The yield curve was positively humped during three major events, namely the bond market crisis in 1994, and the 2001 and 2022 crises. This means that the market expected interest rate hikes in the medium term.

Figure 3. Loadings of the Nelson-Siegel model factor



Source: author's calculations.

Next, the stationarity of the resulting time series is tested using the Augmented Dickey-Fuller (ADF) test at a significance level of 0.05. The key conclusion from the results presented in Table 1 is that L_t and S_t are I(1), while C_t is stationary.

Table 1. Descriptive statistics and ADF test for NS model factors

Variable	Mean	SD	Minimum	Maximum	Skewness	Kurtosis	JB statistic	Levels	ADF test differences
L	5.00	2.24	0.60	10.19	0.21	2.19	307.73	-2.02	-68.51
S	-1.91	1.90	-5.78	2.78	0.04	2.42	125.70	-1.85	-68.73
C	-1.75	2.19	-6.69	5.93	0.04	2.77	22.22	-3.66	-

Note. The critical value for the ADF test is -3.43 at the significance level of 0.01 and -2.86 at the significance level of 0.05.
Source: author's calculations.

The NS factor predictors were downloaded from the Federal Reserve Economic Database[1] (FRED). The following variables were selected:

[1] FRED tickers for these variables are respectively: VIXCLS, KCPRU, DEXUSK, DEXJPUS, DEXSZUS, DCOILWTICO, DAAA.

- Chicago Board Options Exchange (CBOE) Volatility Index (VIX), which is a measure of the uncertainty of the US stock market; Bekaert et al. (2013) demonstrated that uncertainty and risk appetite are negatively related to interest rates;
- Kansas City Fed's Rate Uncertainty which reflects market expectations regarding short-term rate, calculated using publicly traded options contracts;
- Exchange rates of the US dollar against the British pound (USD/GBP), Japanese yen (USD/JPY) and Swiss franc (USD/CHF). For Japan, Akram and Li (2024) demonstrated that exchange rates influence interest rates, as whenever a depreciation (or appreciation) of the domestic currency occurs, the central bank, following a Taylor-type rule, reacts by adjusting the interest rates to counteract the inflationary (or deflationary) pressures;
- West Texas Intermediate (WTI) crude oil prices; according to Akram and Li (2024), commodity prices affect inflation and thus interest rates;
- Moody's Seasoned Aaa Corporate Bond Yield (Aaa yield); Gilchrist and Zakrajšek (2012) showed that positive shocks to the excess bond premium have a negative impact on economic activity, thus necessitating monetary policy easing.

The descriptive statistics of these variables are provided in Table 2. In addition to the basic statistics, we computed skewness, kurtosis and the Jarque-Bera (JB; 1980) test statistic in order to assess whether the variables are normally distributed. The results in the table indicate that none of the variables is normally distributed. Interestingly, the VIX shows the highest skewness and is leptokurtic, while the other variables demonstrate a moderate degree of asymmetry and have platykurtic distributions.

Table 2. Descriptive statistics of the financial variables

Variable	Mean	SD	Minimum	Maximum	Skewness	Kurtosis	JB statistic
VIX	19.44	7.76	9.14	82.69	2.21	11.72	35,742.63
KCPRU	0.95	0.39	0.17	2.18	0.04	2.12	292.04
USD/GBP	1.55	0.21	1.07	2.11	0.28	2.52	205.68
USD/JPY	114.31	17.96	75.72	161.73	0.31	3.02	139.58
USD/CHF	1.17	0.25	0.73	1.82	0.48	2.07	666.64
WTI	51.60	29.05	-36.98	145.31	0.44	2.14	570.80
Aaa yield	5.60	1.75	2.01	9.68	0.23	2.23	299.64

Note. The JB test statistic has a $\chi^2(2)\ distribution$ that has the critical values of 5.99 at the 0.05 significance level and 9.21 at the significance level of 0.01.
Source: author's calculations.

3.2. Forecasting competition design

We divide the whole dataset of 8,930 observations (2nd January 1990–27th January 2026) into three subsets: training (6,100 observations from the 2nd January 1990–29 August 2014 period),

validation (2,575 observations from the 30th August 2014–16th January 2025 period) and testing (255 observations from the 17th January 2025–27th January 2026 period). The first two datasets are used to determine the optimal specification of the ARIMA, VAR and RF models.

In the forecasting competition, we compare four models (ARIMA, VAR, AR RF, AR RF + exogenous variables) over a 1-day horizon and three models (ARIMA, VAR, AR RF) across horizons up to 20 days. For each factor, we fit an ARIMA, VAR, RF model with autoregressive predictors and RF with autoregressive and exogenous predictors. The exact tuning procedures for each model are described below.

For the ARIMA, we use the training set and `auto.arima()` function in the `R forecast` package, which returns the specification that optimises the Bayesian Information Criterion (BIC).

We utilise the `VARselect()` function from the `vars` package for the VAR model, through which we obtained the optimal lag order that minimises the BIC.

For the RF models, we used five lags of the dependent variable and one lag of the exogenous predictors. This approach is called predictive regression (see Stambaugh, 1999), which generates 1-day-ahead forecasts without the need to forecast exogenous predictors. As regards the hyperparameter tuning for the RF model, their values are derived from the following algorithm. For a given set of hyperparameters, we fitted the model using the `rangerts` package (based on the `ranger` package by Wright & Ziegler, 2017) to the training set. We then used its predictions in the validation set to compute the Root Mean Square Error (RMSE). We selected the hyperparameters that optimised the RMSE statistic. The exact list of the tuned hyperparameters and their feasible values are discussed in Section 3.3 of this article.

The testing set was used to compare the forecasting accuracy of the competing models. For this purpose, we employed a rolling origin setup (see Hewamalage et al., 2022). Consequently, starting from the end of the validation set (observation 8,675), we fitted the model to generate a 1-day ahead forecast and then added the actual observation to the set. The process was repeated until reaching the end of the test set. The hyperparameters were fixed at their validation-set optima, meaning that only the model coefficients were re-estimated at each forecast origin.

To obtain longer horizon forecasts, we applied the recursive forecasting approach for the three autoregressive models. Specifically, to obtain $\hat{y}_{t+h+1|t}$ (forecast for the next out-of-sample value), we used all the necessary lags as either forecasted values if they were from periods $t + h, \ \dots \ , \ t + 1$ or actual observations for periods $t, \ t - 1$ and so on.

3.3. Tuning results

The specification of the ARIMA models is presented in Table 3. For each of the three NS model factors, the optimal model specification utilised the first differences ($d = 1$) approach. The application of a second-order autoregressive component best captured the L_t and S_t variables, which exhibited high inertia. Conversely, the optimal model for C_t consisted of a first-order moving average only.

Table 3. Optimal specification of the ARIMA models

Parameter	L	S	C
p	2	2	0
d	1	1	1
q	0	0	1

Source: author's calculations.

Next, in the RF tuning procedure, the algorithm selected the optimal values from the feasible ones (indicated in square brackets) for the following hyperparameters: the number of variables to possibly split at each node (m_{try})[AR RF – 1:5; AR RF + X – 1:11], the maximum number of splits between the beginning and end of the tree ($max\ depth$) [3:30 for both models], the length of the bootstrap block ($block\ length$)[1:100 for both models] and the regularisation parameter that controls overfitting ($minimum\ node\ size$) [5:100 for both models].

As previously mentioned, the estimation was based on the moving-block bootstrap, which adapts the RF method to time series analyses. To accelerate the search for the optimal hyperparameter setting, we applied the random search approach described by Bergstra and Bengio (2012). Instead of evaluating all possible hyperparameter combinations, this approach samples with replacement from the feasible set. A total of 500 hyperparameter combinations were thus sampled.

The values of the hyperparameters that performed best in the validation set are presented in Tables 4 and 5. From among all the variables and models, the optimal $block\ length$ comprised approximately 15 observations, which is the equivalent to three trading weeks. The models for C_t had grown deeper than for the other variables; this may mean that more intricacy is necessary for the proper forecasting of this factor than in the case of the other two. Regarding the m_{try} parameter in the autoregressive models, the value of four across all factors meant that the algorithm used four out of the five available lags to construct one tree. In the models

incorporating exogenous predictors, the algorithm naturally selected a larger number of regressors per tree; for L_t, on the other hand, the requirement was lower by two. The $minimum\ node\ size$ hyperparameter is closely tied to the $max\ depth$. The deeper the tree, the smaller the final node size and vice versa. Both regularisation hyperparameters were optimised to address two different sources of overfitting.

Table 4. Optimal hyperparameters for RF with autoregressive predictors

Hyperparameter	L	S	C
$block\ length$	15	15	12
$max\ depth$	9	9	17
m_{try}	4	4	4
$minimum\ node\ size$	18	18	19

Source: author's calculations.

Table 5. Optimal hyperparameters for RF with autoregressive and exogenous predictors

Hyperparameter	L	S	C
$block\ length$	14	17	17
$max\ depth$	7	25	25
m_{try}	8	10	10
$minimum\ node\ size$	35	6	6

Source: author's calculations.

3.4. Accuracy of the forecast

The aim of the presented research is to verify whether autoregressive RFs are able to deliver additional forecasting power in comparison with ARIMA and VAR models and whether adding financial predictors improves the forecast accuracy of the latent factors. For this purpose, we analysed the 1-day-ahead forecasts generated by each of the four competing methods. Then, we discuss the forecasts of the three strictly autoregressive models over longer forecast horizons.

3.4.1. One-day-ahead forecasts

Table 6. Root mean square forecast error of NS model factors; 1-day ahead forecasts

Factor	ARIMA	Autoregressive RF	VAR	Autoregressive RF with exogenous variables
L	0.0548	0.0593	0.0542	0.0572
S	0.0563	0.0603	0.0575	0.0604
C	0.1581	0.1672	0.1609	0.1688

Source: author's calculations.

We calculated the root mean square forecast error (RMSFE) for each model and factor. The results in Table 6 show that for each factor, ARIMA outperformed both RF types. It was evident that C_t was more difficult to forecast than L_t and S_t. Interestingly, adding financial covariates

to the RF increased the forecast accuracy only for the Level factor. Additionally, despite its regularisation hyperparameters, RF proved susceptible to overfitting on the slightly noisy daily data.

In the final phase of our research, we tested whether the differences in forecast accuracy between the models is statistically significant. For this purpose, we performed the Diebold-Mariano (1995, DM) test, which uses ARIMA as a benchmark. Table 7, which contains the *p*-values of the DM tests, indicates that the use of RFs leads to statistically significant deterioration in forecasting accuracy rather than an improvement, compared to the traditional ARIMA and VAR benchmarks. These findings are at odds with the other studies discussed in the Introduction. Since our analysis is limited to the US market, the discussion that follows concerns the validation design of these studies rather than a direct comparison of forecast accuracy. The observed discrepancy may stem from data leakage, which is not always controlled by other researchers. For example, Kim et al. (2020) do not provide a clear validation strategy as they discuss the partition of their dataset into training and test subsamples only; in addition, they do not disclose which country the data come from. Furthermore, they did not ensure a level playing field for classical and ML models, e.g. they restricted the classical approach to an NS model with AR(1) factors only. This is consistent with the broader concerns raised by Hewamalage et al. (2022) and Puglia and Tucker (2020) regarding leakage (even unintentional) and validation design. Additionally, due to publication bias, many articles that do not find a significant advantage of using ML methods are simply not published.

Table 7. DM test *p*-values for 1-day-ahead forecasts

Factor	ARIMA vs VAR	ARIMA vs Autoregressive RF	VAR vs Autoregressive RF	ARIMA vs Autoregressive RF with exogenous variables	VAR vs Autoregressive RF with exogenous variables	Autoregressive RF with exogenous variables vs Autoregressive RF
L	0.7883	0.0044	0.0006	0.0317	0.0074	0.0282
S	0.0352	0.0044	0.0362	0.0141	0.0638	0.5354
C	0.0488	0.0233	0.0603	0.0248	0.0669	0.6543

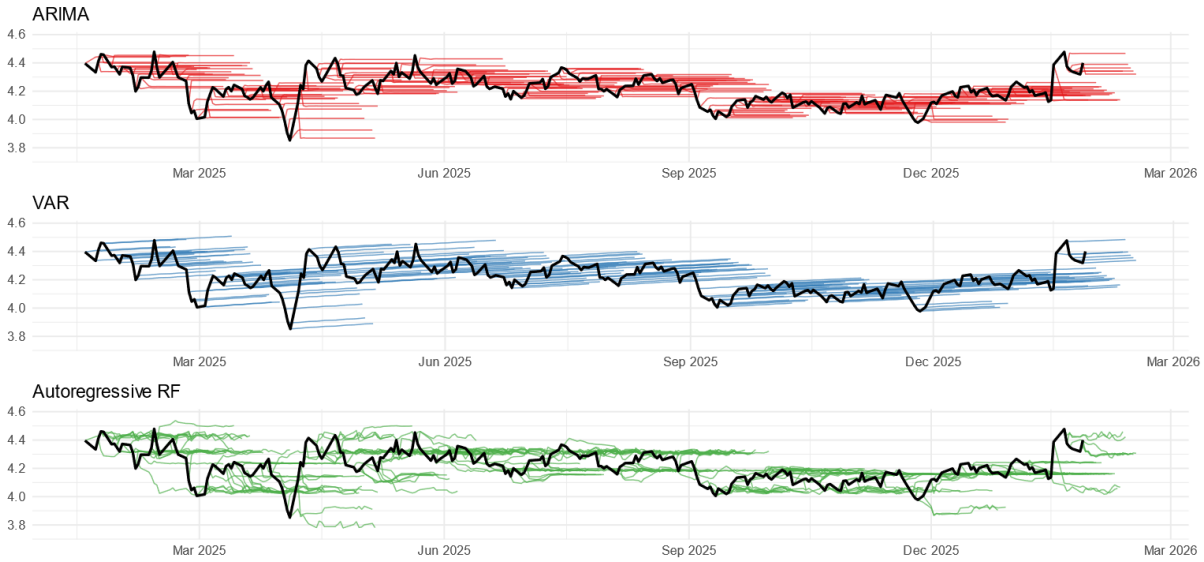
Note. H1: Model 1 is better than model 2.
Source: author's calculations.

3.4.2. Forecasts for 1- to 20-day horizons

In this section, we compare forecasts from the three autoregressive models across horizons from 1 to 20 days. Figures 4–6 demonstrate that all the models tend to provide mean-reverting forecasts. Furthermore, they fail to predict sudden, sharp changes of the latent factors. It is

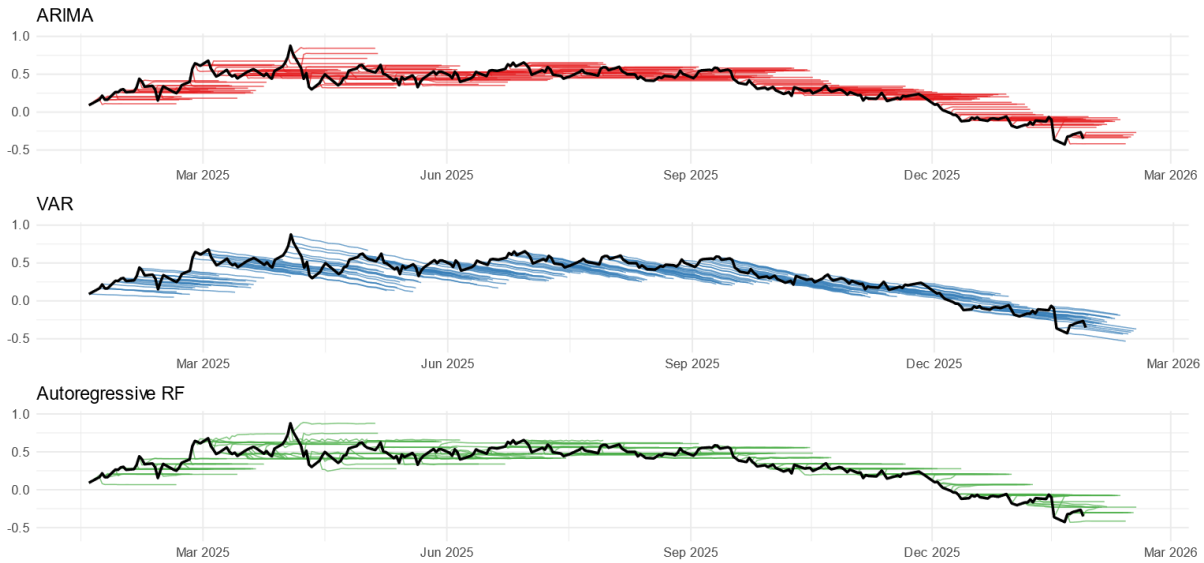
worth noting that the autoregressive RFs return volatile forecasts, whereas ARIMA and VAR models provide smooth trajectories of future values.

Figure 4. Sequential forecasts for the Level factor



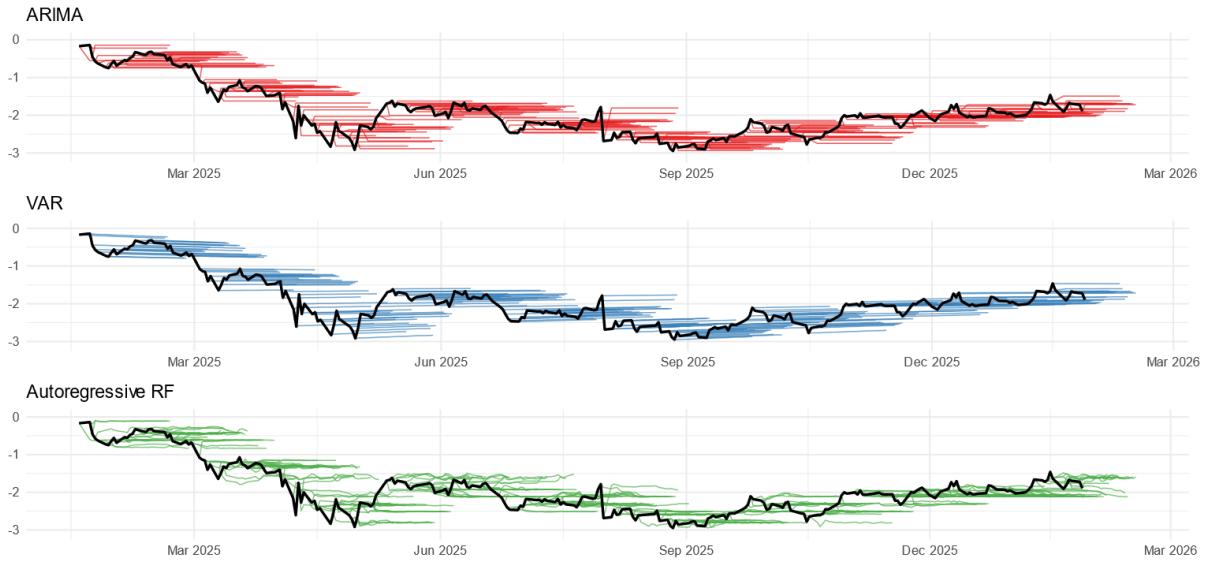
Source: author's calculations.

Figure 5. Sequential forecasts for the Slope factor



Source: author's calculations.

Figure 6. Sequential forecasts for the Curvature factor



Source: author's calculations.

Table 8 shows the RMSFEs of the three autoregressive models across longer horizons, while Table 9 presents the *p*-values of the DM test that allows the verification of the significance of pairwise forecast accuracy differences. To account for serial correlation in the loss differential at longer horizons, the long-run variance is estimated using autocovariances up to lag h-1, where h denotes the forecast horizon (see more in Diebold & Mariano, 1995).

Table 9 suggests that at a 0.05 significance level, ARIMA outperforms considerably VAR just for 7 out of 20 horizons for L_t, and for S_t and C_t in the case of 1-day-ahead forecasts only. ARIMA yields better results than the autoregressive RF across all 20 horizons for L_t and S_t; similarly, it dominates in forecasting C_t up to h = 14 (with the exception of h = 2 and h = 10). VAR generates more accurate forecasts than the autoregressive RF for L_t up to h = 13; however, it fails to outperform the RF significantly at any of the horizons for S_t (except h = 1). For C_t, VAR performs more efficiently for horizons from h = 3 to h = 13.

Table 8. RMSFEs of NS model factors for h = 1, …, 20 forecast horizons

Variable	h = 1	h = 2	h = 3	h = 4	h = 5	h = 6	h = 7	h = 8	h = 9	h = 10	h = 11	h = 12	h = 13	h = 14	h = 15	h = 16	h = 17	h = 18	h = 19	h = 20
ARIMA																				
L	0.0548	0.0784	0.0915	0.1034	0.1120	0.1171	0.1217	0.1254	0.1277	0.1293	0.1291	0.1302	0.1298	0.1284	0.1282	0.1271	0.1263	0.1264	0.1264	0.1268
S	0.0563	0.0799	0.0942	0.1066	0.1169	0.1226	0.1284	0.1335	0.1372	0.1409	0.1434	0.1465	0.1470	0.1474	0.1492	0.1501	0.1518	0.1548	0.1579	0.1610
C	0.1581	0.2006	0.2313	0.2637	0.2868	0.3067	0.3232	0.3447	0.3668	0.3911	0.4083	0.4261	0.4421	0.4538	0.4657	0.4776	0.4913	0.5014	0.5153	0.5230
VAR																				
L	0.0542	0.0786	0.0923	0.1046	0.1136	0.1186	0.1230	0.1270	0.1300	0.1323	0.1325	0.1341	0.1338	0.1327	0.1332	0.1323	0.1321	0.1326	0.1334	0.1344
S	0.0575	0.0812	0.0958	0.1090	0.1204	0.1269	0.1339	0.1400	0.1451	0.1501	0.1544	0.1594	0.1621	0.1645	0.1687	0.1721	0.1761	0.1818	0.1872	0.1930
C	0.1609	0.2014	0.2314	0.2636	0.2869	0.3058	0.3210	0.3414	0.3622	0.3862	0.4023	0.4196	0.4350	0.4460	0.4566	0.4672	0.4800	0.4882	0.5018	0.5080
Autoregressive RF																				
L	0.0593	0.0862	0.1027	0.1156	0.1282	0.1349	0.1385	0.1428	0.1477	0.1499	0.1452	0.1441	0.1430	0.1409	0.1400	0.1385	0.1403	0.1433	0.1435	0.1466
S	0.0603	0.0828	0.0980	0.1103	0.1218	0.1286	0.1328	0.1371	0.1412	0.1455	0.1485	0.1525	0.1541	0.1553	0.1570	0.1590	0.1607	0.1623	0.1641	0.1676
C	0.1672	0.2059	0.2433	0.2820	0.3031	0.3254	0.3371	0.3598	0.3800	0.4010	0.4166	0.4326	0.4513	0.4623	0.4720	0.4829	0.4971	0.5032	0.5157	0.5271

Note. H1: Model 1 is better than model 2.
Source: authors' calculations.

Table 9. DM test *p*-values for h = 1, …, 20 forecast horizons

Variable	h = 1	h = 2	h = 3	h = 4	h = 5	h = 6	h = 7	h = 8	h = 9	h = 10	h = 11	h = 12	h = 13	h = 14	h = 15	h = 16	h = 17	h = 18	h = 19	h = 20
ARIMA vs. VAR																				
L	0.7883	0.4396	0.1611	0.0120	0.0101	0.0967	0.1615	0.1514	0.1025	0.0585	0.0595	0.0479	0.0472	0.0391	0.0303	0.0463	0.0563	0.0756	0.0866	0.0969
S	0.0352	0.1874	0.2653	0.2598	0.2313	0.2456	0.2385	0.2413	0.2273	0.2141	0.1949	0.1727	0.1585	0.1541	0.1493	0.1486	0.1557	0.1576	0.1631	0.1654
C	0.0488	0.3146	0.4970	0.5357	0.4823	0.6122	0.7188	0.7952	0.8597	0.8494	0.8751	0.8785	0.8760	0.8810	0.9072	0.9227	0.9335	0.9472	0.9397	0.9383
ARIMA vs. Autoregressive RF																				
L	0.0044	0.0051	0.0021	0.0025	0.0004	0.0000	0.0001	0.0000	0.0000	0.0000	0.0002	0.0013	0.0037	0.0033	0.0028	0.0037	0.0003	0.0004	0.0028	0.0069
S	0.0044	0.0246	0.0098	0.0416	0.0140	0.0117	0.0207	0.0536	0.0446	0.0403	0.0493	0.0383	0.0343	0.0375	0.0288	0.0185	0.0069	0.0057	0.0001	0.0010
C	0.0233	0.1768	0.0022	0.0001	0.0001	0.0031	0.0301	0.0145	0.0284	0.0553	0.0403	0.0146	0.0311	0.1062	0.1136	0.1273	0.1707	0.3279	0.4703	0.2344
VAR vs. Autoregressive RF																				
L	0.0006	0.0032	0.0042	0.0071	0.0014	0.0001	0.0002	0.0001	0.0001	0.0000	0.0003	0.0058	0.0177	0.0325	0.0814	0.1465	0.0871	0.0661	0.1052	0.1175
S	0.0362	0.2234	0.2329	0.3716	0.3803	0.3985	0.5580	0.6241	0.6478	0.6620	0.6909	0.7069	0.7103	0.7149	0.7391	0.7375	0.7452	0.7699	0.7821	0.7823
C	0.0603	0.1969	0.0016	0.0009	0.0025	0.0040	0.0137	0.0064	0.0141	0.0351	0.0272	0.0310	0.0238	0.0682	0.0715	0.0559	0.0533	0.0552	0.1013	0.0575

Note. H1: Model 1 is better than model 2.
Source: author's calculations.

4. Conclusions

The principal finding of this study is that the persistent use of ML methods for modelling the US yield curve is not always superior to classical approaches. We have found that using ARIMA and VAR for modelling the NS model parameters outperforms the RF (with time-series specific bootstrap) approach in 1-day-ahead forecasts. Adding financial variables to the autoregressive RFs significantly improved the forecasts for the Level factor only. These conclusions suggest that the RFs tended to overfit to the noise in the daily data and that the daily NS factors are just heavily autocorrelated. Therefore, more parsimonious models such as ARIMA or VAR remain effective in capturing the whole signal and providing accurate forecasts.

Likewise, for longer forecast horizons (up to 20 trading days), ARIMA and VAR models outperformed the RF. Compared to the RF, ARIMA generated more accurate forecasts for all the factors, whereas VAR provided better forecasts for the Level and Curvature factors. Interestingly, ARIMA generated more accurate forecasts for the Level and Slope factors (although not statistically significant as far as the second variable is concerned) than VAR. Curvature forecasts, however, were more accurate when made from the VAR model, probably because lagged values of the two other factors allow a better understanding of the yield curve's curvature.

The yield forecasts from all three models were sometimes systematically higher or lower than the actual values. This indicates the presence of a cross-section error stemming from a poor fit of the NS model to the actual yield curve. It is worth noting that this is an inherent fitting error as the NS decomposition is only an approximation of the yield curve. This may also result from using a fixed λ that can vary from the optimal λ for each day. The main advantage of using a fixed λ is that it allows the comparability with other studies.

This study additionally emphasises the need for rigorous data-leakage protection and addressing the problem of publication bias in the ML field.

References

Akram, T., & Li, H. (2024). Empirical Models of JGB Yields Using Daily Data. *Journal of Economic Issues*, *58*(3), 1011–1034. https://doi.org/10.1080/00213624.2024.2382051.

Bekaert, G., Hoerova, M., & Lo Duca, M. (2013). Risk, uncertainty and monetary policy. *Journal of Monetary Economics*, *60*(7), 771–788. https://doi.org/10.1016/j.jmoneco.2013.06.003.

Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, *13*(10), 281–305. https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf.

.

Bie, S., Diebold, F. X., He, J., & Li, J. (2024). *Machine Learning and the Yield Curve: Tree-Based Macroeconomic Regime Switching*. https://doi.org/10.2139/ssrn.4934442.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1023/A:1010933404324.

Cadahia Delgado, P., Congregado, E., Golpe, A. A., & Vides, J. C. (2022). The Yield Curve as a Recession Leading Indicator. An Application for Gradient Boosting and Random Forest. *International Journal of Interactive Multimedia and Artificial Intelligence*, *7*(3), 7–19. https://doi.org/10.9781/ijimai.2022.02.006.

Christensen, J. H. E., Diebold, F. X., & Rudebusch, G. D. (2011). The affine arbitrage-free class of Nelson–Siegel term structure models. *Journal of Econometrics*, *164*(1), 4–20. https://doi.org/10.1016/j.jeconom.2011.02.011.

Diebold, F. X., & Li, C. (2006). Forecasting the term structure of government bond yields. *Journal of Econometrics*, *130*(2), 337–364. https://doi.org/10.1016/j.jeconom.2005.03.005.

Diebold, F. X., & Mariano, R. S. (1995). Comparing Predictive Accuracy. *Journal of Business & Economic Statistics*, *13*(3), 253–263. https://doi.org/10.1080/07350015.1995.10524599.

Diebold, F. X., Rudebusch, G. D., & Aruoba, S. B. (2006). The macroeconomy and the yield curve: a dynamic latent factor approach. *Journal of Econometrics*, *131*(1–2), 309–338. https://doi.org/10.1016/j.jeconom.2005.01.011.

Gilchrist, S., & Zakrajšek, E. (2012). Credit Spreads and Business Cycle Fluctuations. *American Economic Review*, *102*(4), 1692–1720. https://doi.org/10.1257/aer.102.4.1692.

Goehry, B., Yan, H., Goude, Y., Massart, P., & Poggi, J.-M. (2023). Random Forests for Time Series. *REVSTAT – Statistical Journal*, *21*(2), 283–302. https://doi.org/10.57805/revstat.v21i2.400.

Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press.

Hewamalage, H., Ackermann, K., & Bergmeir, C. (2022). Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, *37*, 788–832. https://doi.org/10.1007/s10618-022-00894-5.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning* (2nd ed.). Springer.

Jarque, C. M., & Bera, A. K. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, *6*(3), 255–259. https://doi.org/10.1016/0165-1765(80)90024-5.

Kim, W. J., Jung, G., & Choi, S.-Y. (2020). Forecasting CDS Term Structure Based on Nelson–Siegel Model and Machine Learning. *Complexity*, *1*, 1–23. https://doi.org/10.1155/2020/2518283.

Kostyra, T. P., & Rubaszek, M. (2020). Forecasting the Yield Curve for Poland. *Econometric Research in Finance*, *5*(2), 103–117. https://doi.org/10.2478/erfin-2020-0006.

Nelson, C. R., & Siegel, A. F. (1987). Parsimonious Modeling of Yield Curves. *The Journal of Business*, *60*(4), 473–489.

Puglia, M., & Tucker, A. (2020). *Machine Learning, the Treasury Yield Curve and Recession Forecasting* (Finance and Economics Discussion Series 2020-038). https://doi.org/10.17016/feds.2020.038.

Rayeni, A., & Naderi, H. (2025). Predicting the Canadian Yield Curve Using Machine Learning Techniques. *International Journal of Financial Studies*, *13*(3), 1–30. https://doi.org/10.3390/ijfs13030170.

Richman, R., & Scognamiglio, S. (2024). Multiple yield curve modeling and forecasting using deep learning. *ASTIN Bulletin*, *54*(3), 463–494. https://doi.org/10.1017/asb.2024.26.

Rubaszek, M. (2012). *Modelowanie polskiej gospodarki z pakietem R*. Oficyna Wydawnicza SGH.

Rubaszek, M., & Sznajderska, A. (2026). *Data leakage in time-series forecasting: Lessons from exchange rate prediction*.

Santos Soares, S. A. (2025). *Eu-Bonds Yield Curve Forecast: Comparing ARIMA and XGBoost Models* [master's thesis, Lisbon School of Economics & Management].

Sims, C. A. (1980). Macroeconomics and Reality. *Econometrica*, *48*(1), 1–48. https://doi.org/10.2307/1912017.

Stambaugh, R. F. (1999). Predictive regressions. *Journal of Financial Economics*, *54*(3), 375–421. https://doi.org/10.1016/s0304-405x(99)00041-0.

Wright, M. N., & Ziegler, A. (2017). ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. *Journal of Statistical Software*, *77*(1), 1–17. https://doi.org/10.18637/jss.v077.i01.

Zhang, J. (2024). *Forecasting Chinese Government Bond Yield Curves: An Empirical Comparison of DNS (Dynamic-Nelson-Siegel) Model and Machine Learning Approaches* [master's thesis, University of Chicago].