A machine learning ensemble framework to forecast the yield curve

Marcelo Martin Navarro[1] and Pablo Orazi[2]

Abstract

We examine the implementation and performance evaluation of an ensemble model to forecast the yield curve. In the strategic and tactical asset allocation process, the forecasting of yield curves plays a vital role and investment horizons matter. Adopting a single "perfect" model for forecasting the term structure of interest rates in different horizons could lead to model misspecification and model uncertainty. In addition, using only one model to address multiple forecasting targets might lead to prediction inaccuracies. Thus, we utilise machine learning tools to construct an ensemble model that aggregates forecasts from various quantitative models using a novel weighting algorithm. The ensemble aims to minimise both point and interval prediction errors while ensuring the model's robustness to unseen data points across the yield curve. Each model in the ensemble could learn a specific pattern specialised in predicting a given feature of the yield curve. The objective of this work is to build a framework for interest rate forecasting that allows for incremental improvement by incorporating different models, rather than discarding them.

JEL classifications: C52, C53, C61, E43, E47, G11.

[1] marcelo.martin@bcra.gob.ar

[2] pabloorazi@gmail.com

1. Introduction

1.1. Machine learning and finance

In recent years, machine learning (ML) has become a popular technique in various domains, finding broad acceptance in the scientific community. We adopt the terminology introduced by Gu et al (2019) and use the term "machine learning" to describe (a) a diverse collection of high-dimensional models for statistical prediction, combined with (b) so-called "regularisation" methods for model selection and mitigation of overfit, and (c) efficient algorithms for searching among a vast number of potential model specifications.

The financial industry is one of the most significant sectors with high prospects to benefit from the advantages of ML, with the availability of big data, innovative algorithms and novel methods in its various applications. For instance, ML has revolutionised retail credit risk management for banks by enabling more accurate estimation of default probability, exposure and recovery (Shi et al (2022)). By leveraging advanced algorithms and analysing comprehensive data sets, ML models can identify patterns and correlations that traditional credit scoring methods may overlook. Another prominent application of ML involves predicting cross-sectional stock returns, where ML algorithms such as boosted trees, random forests and neural networks are utilised to forecast whether individual security prices are expected to rise or fall (Jansen (2020)). This approach has also been recently highlighted in studies such as Hanauer and Kalsbach (2023) and Choi et al (2022), which investigate the impact of ML on the cross section of emerging market stock returns and international stock returns, respectively. Additionally, ML techniques have been employed to anticipate stock crashes by using algorithms like logistic regression, random forests and gradient boosted tree models to identify the worst-performing stocks (Swinkels and Hoogteijling (2022)). Moreover, the integration of natural language processing in multiple languages has facilitated the analysis of textual data, enabling better comprehension and interpretation of financial information (Shah et al (2023)). These advances in ML have extended beyond stock prediction to encompass tasks such as data cleaning (Côté et al (2023)), fraud detection (West and Bhattacharya (2016)), credit scoring (Lessmann et al (2015)) and trading optimisation (Sezer et al (2020)), demonstrating the versatility and potential of ML in revolutionising financial strategies and decision-making processes.

As for central banking, the integration of ML into financial frameworks enables policymakers to navigate increasingly complex economic environments by supporting conditional forecasts and scenario analysis, facilitating the modelling of the economy and providing valuable insights for informed decision-making. However, financial markets are characterised by constant uncertainty driven by structural changes, which reduces the relevance of historical data from distant periods for training ML algorithms. This limitation often results in inherently short data samples with low signal-to-noise ratios, posing significant challenges like overfitting. To address these issues, practitioners employ strategies such as cross-validation, feature selection, regularisation, using holdout data and ensembling (Kolanovic and Krishnamachari (2017)), which collectively enhance the robustness and reliability of ML applications in finance.

It is generally hard to choose among several forecasts due to the variability in performance across different circumstances; a model that excels for specific target horizons may perform poorly in others. Ensemble modelling, inspired by the principles of forecast combinations (Aiolfi et al (2010); Elliot and Timmermann (2016)), has emerged as a powerful forecasting technique aimed at dealing with the misspecification biases that affect individual models. Ensemble methods aim to improve prediction accuracy and robustness by integrating multiple models and diminishing model uncertainty. Moreover, ensemble models that combine predictions from a variety of statistical models can help mitigate the issue of overfitting, as emphasised by Rasekhschaffe and Jones (2019), who advocate for model averaging across different models, training sets and forecast horizons.

An interesting result from the literature suggests that equal weighting, where each forecast receives the same weight, performs remarkably well compared with more complex, theoretically "optimal" combinations (Diebold (2024)). This phenomenon, known as the "equal weight puzzle", can be understood as an example of extreme shrinkage. Diebold (2024) demonstrates that the equally weighted strategy, while not generally optimal, often yields results close to the optimum and outperforms individual forecasts, with equal weights being fully optimal in cases of equi-correlation.[3] Our work will start from this baseline, attempting to make an optimal combination of models, avoiding the overfitting generally produced by optimisations, through an equally weighted prior and regularisation.

1.2. Investment process

The modelling proposed in this paper is designed to integrate into the investment process. A general framework for the investment process, as described by Joia and Coche (2010), involves three phases: strategic asset allocation (SAA), tactical asset allocation (TAA) and portfolio management and implementation. This process begins with a set of constraints, risk-return preferences and expected risk and returns. The SAA defines a strategic benchmark for the long-term portfolio strategy, encompassing multi-year expectations of returns and risk. In the TAA, an active stance is taken on the strategic benchmark. The investment views expressed in the TAA generally involve a horizon shorter than a year. For instance, if rates are expected to rise in the short term, the portfolio duration would be reduced relative to the strategic benchmark to achieve better returns if the expected scenario materialises. The deviation from the benchmark is constrained by the risk budget, usually set in terms of tracking error, which limits the tactical benchmark's potential deviation from the strategic benchmark. Finally, the third phase of the investment process is portfolio management and implementation of the tactical benchmark.

In the first two phases, forecasting yield curves plays a vital role since expected returns depend on them. The modelling of interest rates should consider the multiple investment horizons used in the process. Diverse modelling approaches often perform inconsistently across different horizons (eg three months, six months, 12 months, 36 months, etc) and terms (three-month, two-year, five-year, etc), a pattern documented by Caldeira et al (2016). With this problem in mind, we propose using an ensemble of models to target each horizon and term so that better models and forecasts can be used to improve the accuracy of forecasts through the investment process.

[3] If the models have similar RMSEs (root mean square errors) and correlations among themselves, the optimal solution tends towards 1/n.

2. Data and models

In this study we focused on end-month data of the US Treasury yield curve. Specifically, we chose to model the three-month, six-month, one-year, two-year, three-year, five-year, seven-year and 10-year market yield on US Treasury securities at given constant maturity. We also considered exogenous features such as consumer price index (CPI) inflation and the output gap,[4] collected at monthly data frequency. The data were sourced from the Federal Reserve Bank of St Louis and Bloomberg, covering the period from September 1981 to June 2024.

The primary objective of this study was to combine different time series techniques into an ensemble model for forecasting US Treasury yields. As part of constructing the general framework and technological infrastructure for this methodology, we employed different model variants, aiming to sequentially enhance the ensemble by integrating traditional econometric, factor-based and ML approaches. We expected these models not to be a priori the most precise, but to serve as a benchmark and companion to more sophisticated models in the ensemble.

2.1. Naive forecaster

We chose a naive model as the primary benchmark which provides $t + h$-step-ahead forecasts for a yield of maturity τ as follows:

$$y_{t+h}(\tau) = y_t(\tau)$$

This model served as a robust benchmark for evaluating the predictive performance of other models, given that achieving superior out-of-sample forecasting accuracy compared with the random walk is challenging in practice.

2.2. Autoregressive models

We considered traditional econometric techniques such as autoregressive (AR) models, together with their exogenous extensions. Specifically, we forecasted the maturity-τ yield using first-order univariate AR models AR(1) and ARX(1) that were estimated on the available data for that maturity:

$$y_t(\tau) = \alpha + \beta y_{t-1}(\tau) + \gamma x_t + \varepsilon_t,$$

where x_t corresponds to the exogenous features and $\varepsilon_t(\tau) \sim N(0, \sigma^2(\tau))$. In this case, we utilised CPI inflation and the output gap as exogenous variables.

2.3. Vector autoregressive models

The yield curve, viewed as a vector process of yields across different maturities, implies that cross-sectional information is essential for comprehending its movements. Therefore, first-order unrestricted vector autoregressive models (VAR) and vector autoregressive models with exogenous variables (VARX) for yield levels serve as natural extensions of the univariate AR model:

$$y_t = A + By_{t-1} + Cx_t + \varepsilon_t, \quad where \quad y_t = \big(y_t(\tau_1), y_t(\tau_2), \dots, y_t(\tau_N)\big)$$

[4] The output gap is estimated using an unobserved components model.

For VARX, we employed a specification with CPI inflation and the output gap as exogenous variables.

2.4. Dynamic factor model

The dynamic factor model (DFM) is utilised to capture the underlying common factors driving the movements in the yield curve. The DFM is particularly suitable for analysing and forecasting multivariate time series data, like yields of different maturities, due to its ability to reduce dimensionality and capture common dynamics. We utilised a DFM with a single factor that follows an AR(1) process which can be represented as follows:

$$y_t = \Lambda f_t + \varepsilon_t, \quad where \quad f_t = A f_{t-1} + \eta_t$$

2.5. Dynamic Nelson-Siegel model

We also considered the dynamic Nelson-Siegel (DNS) model, which is widely used for yield curve modelling and forecasting due to its ability to parsimoniously capture the level, slope and curvature of the yield curve (Diebold and Li ((2006)). We focused on three different specifications of the DNS:

a. Basic DNS model:

This specification captures the dynamics of the yield curve by modelling the level, slope and curvature as latent factors.

$$y_t(\tau) = H.\beta_t + e_t\,, \qquad where \qquad \beta_t = \mu + F\beta_{t-1} + \upsilon_t$$

$$H = \left[1 \quad \frac{1-e^{-\lambda\tau}}{\lambda\tau} \quad \frac{1-e^{-\lambda\tau}}{\lambda\tau} - e^{-\lambda\tau}\right], \qquad \beta = [Level \quad -Slope \quad Curvature]$$

b. Rotated DNS model with Taylor rule:

In this specification, a factor rotation scheme is applied to the DNS model (Nyholm (2015)), allowing for direct parametrisation of the short rate process. This approach enables the model to directly incorporate macroeconomic conditions, such as the CPI and output gap, in a Taylor-rule fashion.

$$y_t(\tau) = G.\gamma_t + e_t, \quad where\ \gamma_t = m + F\gamma_{t-1} + QM_t + z_t, \quad M = [CPI,\ Output\ Gap]'$$

$$G = H.A^{-1}, \ \gamma = [Short\ Rate \quad Slope \quad Curvature]$$

The rotation matrix A is defined explicitly as:

$$A = \begin{bmatrix} 1 & \frac{1-e^{-\lambda\tau_s}}{\lambda\tau_s} & \frac{1-e^{-\lambda\tau_s}}{\lambda\tau_s} - e^{-\lambda\tau} \\ 0 & -\frac{1-e^{-\lambda\tau_s}}{\lambda\tau_s} & -\left(\frac{1-e^{-\lambda\tau}}{\lambda\tau_s} - e^{-\lambda\tau_s}\right) \\ 0 & 1-\frac{1-e^{-\lambda\tau_s}}{\lambda\tau_s} & 1-\left(\frac{1-e^{-\lambda\tau_s}}{\lambda\tau_s} - e^{-\lambda\tau_s}\right) \end{bmatrix}$$

The short maturity τ_s is typically set at three months.

c. DNS model with long-run rate:

This specification incorporates a long-run factor LR_t into the DNS model by introducing a 10-year moving average of the 10-year interest rate as an exogenous variable. The long-run factor represents structural components or long-term trends that impact the yield curve. By including this factor, the model accounts for persistent shifts in the yield curve that are not captured by short-term dynamics alone.

$$y_t(\tau) = H.\beta_t + e_t \,, \quad where \quad \beta_t = \mu + F\beta_{t-1} + \Theta LR_t + v_t$$

2.6. Sliding window variants

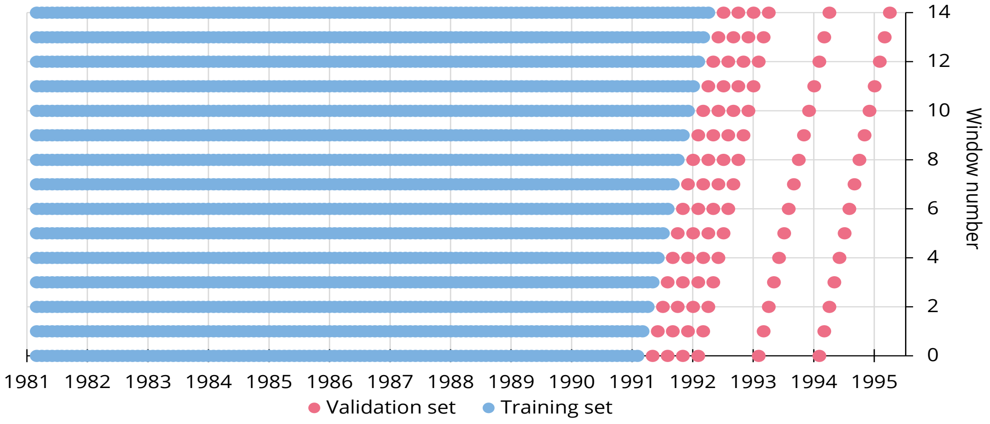
In addition to the standard models, we included sliding window variants to enhance predictive accuracy by focusing on more recent data. Specifically, we utilised a 10-year sliding window approach for AR(1), ARX(1), VAR(1), VARX(1), DFM and DNS models.

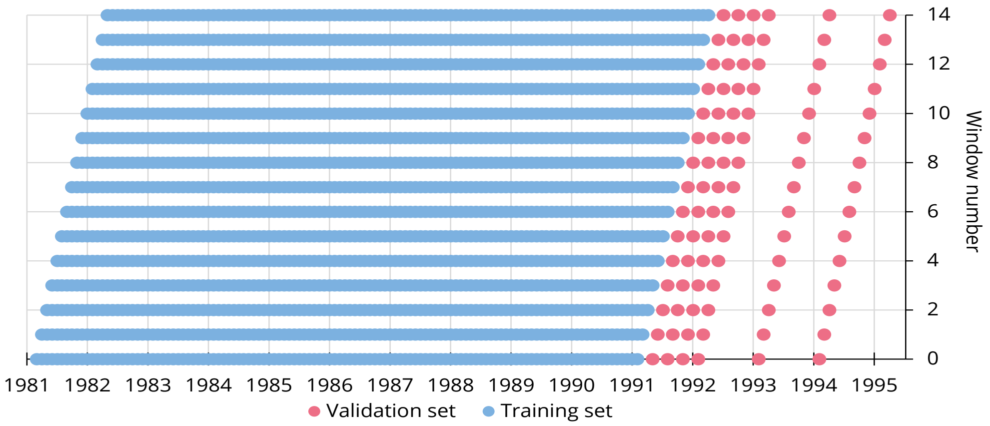
3. Methodology

3.1. Backtesting and cross validation

To create an ensemble model, we first assessed each model's forecasting accuracy in the context of a backtesting exercise. To avoid overfitting and in-sample error estimations, we utilised the technique called walk forward cross validation, which simulates how the model would perform in a real-world scenario where new data become available over time. We employed an expanding window splitter for most models, which generates folds that evolve over time, with the length of the training set for each fold increasing while the validation sets remain constant (Graph 1). The forecast horizons chosen were three, six, nine, 12, 24 and 36 months. Some models, nevertheless, were fitted with sliding windows, to reflect the possibility of changing parameters as time goes by.

Expanding window (up) and sliding window (down) splitters Graph 1





Source: Author's calculations.

3.2. Forecast evaluation

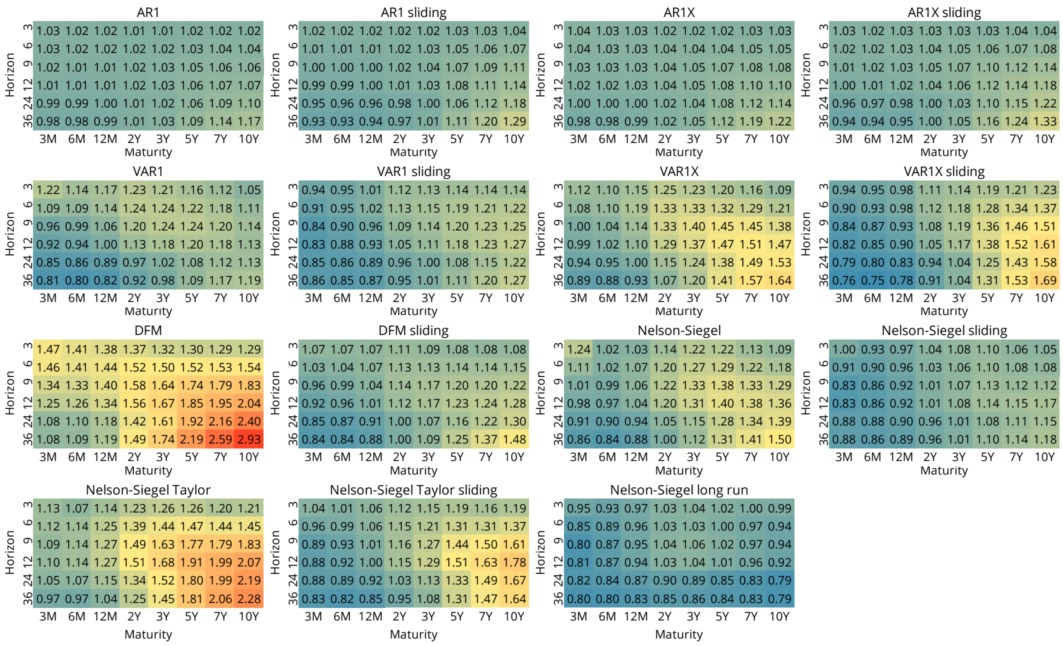
We executed the backtesting for each model and kept the historical predictions for each combination of forecast horizon and maturity. Subsequently, we merged this data set with the actual values to compute the historical errors and the root mean square errors (RMSEs). With this information as inputs, we built heatmaps for a visual comparison of each model's forecasting performance. Certainly, the rankings of forecast accuracy can vary significantly depending on the choice of loss functions and forecast horizons (Petropoulos et al (2022)).

Graph 2 exhibits relative magnitudes of RMSEs with respect to a benchmark model, in this case the naive model. Values greater (lower) than 1 indicate the model is worse (better) than the naive forecaster for that specific horizon and maturity. Graph 3 displays an alternative heatmap using a z-score standardisation to compare models, subtracting the mean and dividing by the standard deviation for each RMSE at that specific horizon and maturity. Moreover, the heatmap in Graph 4 uses a mean standardisation, in which we divide each RMSE by the mean RMSE of all models in that specific horizon and maturity.

These three types of measurements focus on different benchmarking concepts. The first uses a naive model as a benchmark/baseline and helps figure out what horizons and rates are best modelled with random walks. The second, with z-score standardisation, aids in finding the best models. This measure, as it is divided by the standard deviation of RMSE measures, gives us a sense of ordering but is not helpful in discerning whether a model is much better or slightly better than the average (all models might have similar RMSEs). That is why we also monitored the ratio of RMSE to the average RMSE of all models for the same term and horizon, as shown in Graph 5.

Overall, it can be inferred that some models perform better than others depending on the forecast horizon and maturity. For instance, models such as VAR and DNS typically outperformed the rest of the models at longer forecast horizons. Additionally, we observed that the naive model proved challenging to outperform, particularly at shorter forecast horizons.

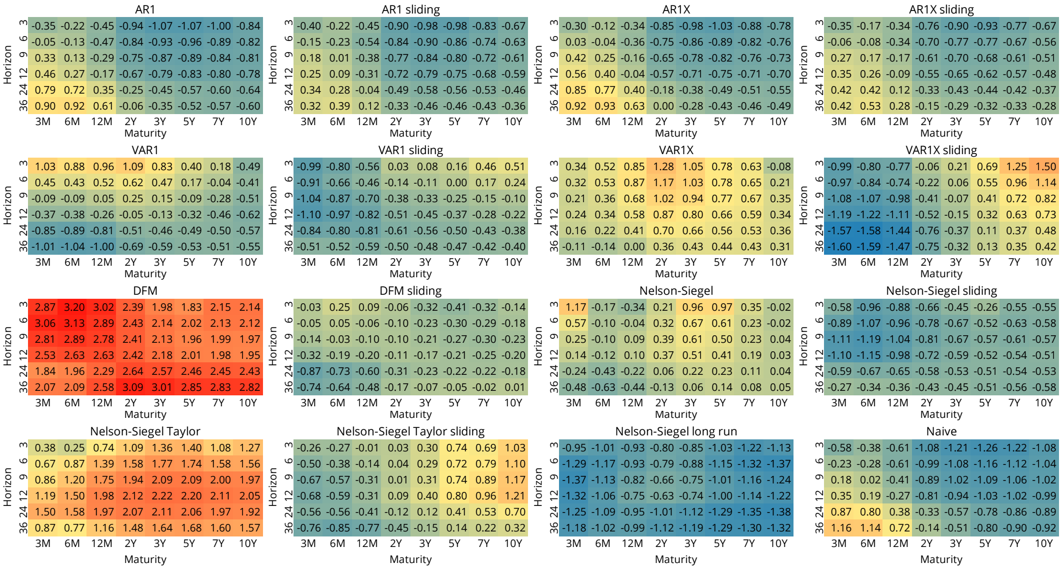
Relative magnitudes of model predictions vs naive (RMSE) Graph 2



Values greater (lower) than 1 indicate the model is worse (better) than the Naive forecaster for that specific horizon and maturity.

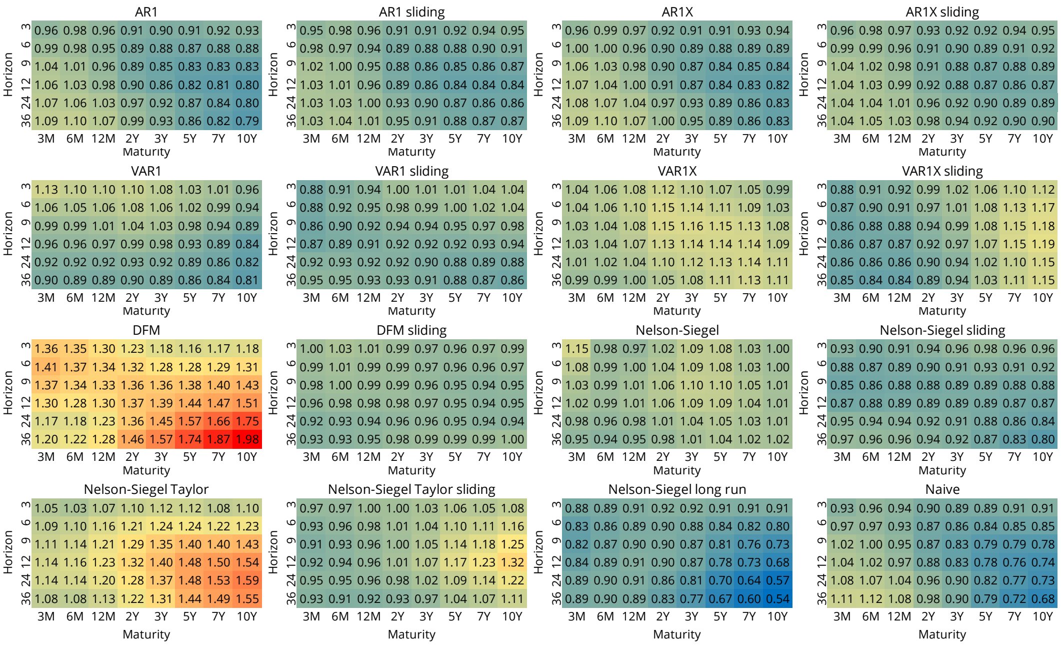
Source: Author's calculations.

Comparison of model predictions, z-score standardisation (RMSE) — Graph 3



Source: Author's calculations.

Comparison of model predictions, mean standardisation (RMSE) — Graph 4



Source: Author's calculations.

3.3. Ensemble optimisation

The next step involved the optimisation process to construct the ensemble of models. We proposed an objective function that simultaneously includes an exponentially decaying discount factor, which adjusts the influence of past errors, and a regularisation parameter, which controls weight diversification. Specifically, our goal was to minimise the following equation:

$$[w^T X^T \, diag(d_i) \, X \, w] + \frac{\lambda}{m}\left(w - \frac{1}{m}\right)^T \left(w - \frac{1}{m}\right) \; subject \; to \; 1e'w = 1 \; and \; w \geq 0 \qquad (1)$$

$$d_i = \frac{e^{-\alpha(N-1-i)}}{\sum_{j=0}^{N-1} e^{-\alpha(N-1-j)}} \quad \textit{for each i from 0 to N-1,}$$

where X denotes the historical forecast errors, w the weights of each model in the ensemble, N the number of observations, m the number of models, d_i the discount factor, α the decay rate and λ the regularisation parameter.

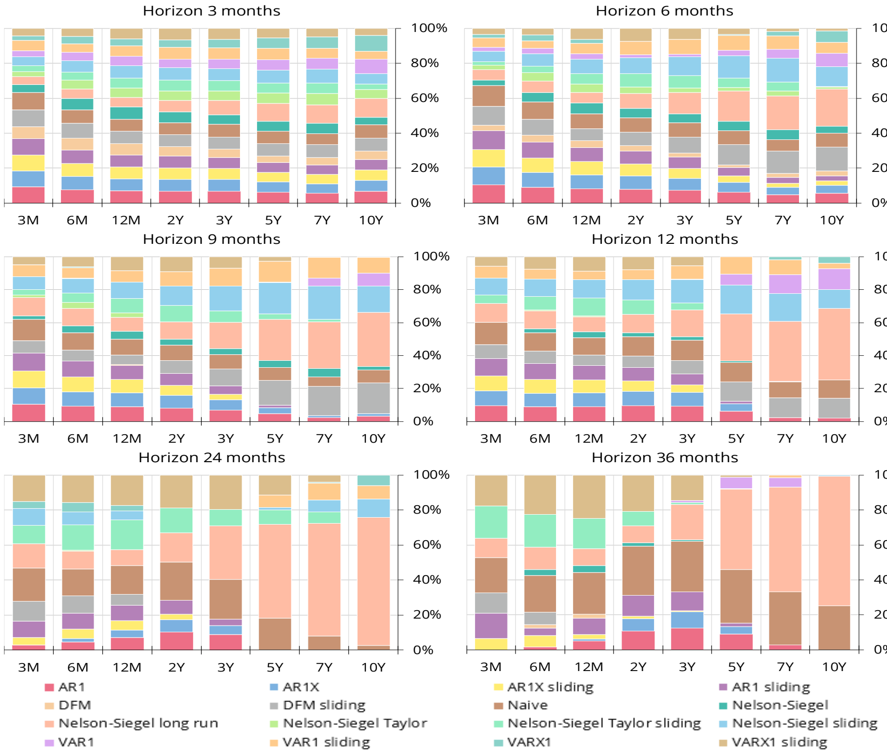
The first term in equation (1) calculates the mean square error with an exponentially decayed factor governed by α, ensuring that more recent errors have higher weights and thus have a greater impact on the combined forecast. The second term corresponds to a regularisation penalty, controlled by λ, which discourages extreme weights and encourages diversification, helping to prevent overreliance on a single model and potentially improving robustness. This regularisation technique is akin to Bayesian regression, where it is assumed a priori that all weights are equal, and the evidence modifies these values. Such an approach is crucial in avoiding overfitting, as it ensures that no individual model disproportionately influences the ensemble prediction unless supported by substantial evidence in the data. The constraints determine that all weights must sum up to 1 and that they must be greater than or equal to 0.

4. Empirical results

Before conducting the optimisation, we performed a train-test split to ensure that the evaluation of the ensemble would be conducted on an out-of-sample data set. Specifically, we selected the most recent seven years of data as the final test set, providing an unbiased assessment of the ensemble's generalisation ability. After conducting a grid search for various values of λ and α, we determined the values of w that minimise equation (1) for each combination of *horizon – maturity* in the training set. We created a walk forward cross validation with an initial window of 70 months and forecasted the targeted horizon. Using the optimised weights, we computed the forecast errors ahead and calculated different metrics to measure accuracy: root mean square error (RMSE), mean square error (MSE), mean absolute error (MAE) and median absolute error (MEDAE).

All metrics yielded an $\alpha = 0.05$, while optimal values for λ varied across metrics: both RMSE and MEDAE resulted in $\lambda = 3.8$, while MAE and MSE achieved their optimal performance with $\lambda = 4.9$ and $\lambda = 2.6$, respectively. Higher values for λ imply that the model gets closer to using equal weights. Therefore, we proposed two alternative strategies: an equally weighted and an optimally weighted ensemble. Finally, we established $\lambda = 3.8$ and $\alpha = 0.05$ and ran the optimisation again to get the optimal weights (Graph 5).

Optimal weights Graph 5



Source: Author's calculations.

Next, we performed an out-of-sample evaluation of the ensemble using the reserved test subset. We computed the forecast errors of the ensemble model using both optimal and equal weights to assess their forecasting performance relative to the other models in the test subset. We constructed heatmaps in a similar fashion to our previous method, employing the same three baselines. Graphs 6 and 7 illustrate that both ensemble methods performed favourably across all benchmarks. Notably, the optimally weighted ensemble outperformed the naive benchmark in 32 out of 40 targets, compared with 29 targets for the equally weighted ensemble. Moreover, when averaging the mean standardised errors across all targets, the optimally weighted approach achieved a value of 0.94, slightly lower than the 0.95 observed for the equally weighted method. Similarly, the average z-score standardised errors were -0.43 for the optimally weighted ensemble and -0.41 for the equally weighted ensemble, indicating a slight advantage for the optimal weighting approach. This comprehensive evaluation solidifies the ensemble model's position as a robust and effective tool for forecasting the yield curve, offering valuable insights for decision-making and the investment process.

Optimally weighted ensemble — Graph 6

Ensemble vs naive

Horizon	3M	6M	12M	2Y	3Y	5Y	7Y	10Y
3	0.88	0.91	0.98	1.06	1.10	1.15	1.12	1.09
6	0.87	0.91	0.98	1.06	1.09	1.14	1.15	1.11
9	0.86	0.90	0.97	1.03	1.05	1.08	1.10	1.07
12	0.87	0.91	0.96	0.99	0.99	1.00	1.02	0.99
24	0.90	0.91	0.91	0.91	0.89	0.84	0.82	0.81
36	0.86	0.85	0.84	0.85	0.87	0.87	0.87	0.86

Maturity

Ensemble, z-score standardisation

Horizon	3M	6M	12M	2Y	3Y	5Y	7Y	10Y
3	-0.60	-0.49	-0.43	-0.41	-0.39	-0.34	-0.37	-0.53
6	-0.49	-0.56	-0.57	-0.51	-0.48	-0.47	-0.51	-0.68
9	-0.37	-0.48	-0.47	-0.41	-0.47	-0.54	-0.62	-0.84
12	-0.37	-0.38	-0.30	-0.39	-0.57	-0.75	-0.79	-1.01
24	-0.04	0.02	0.07	0.05	-0.20	-0.96	-1.29	-1.43
36	0.03	-0.16	-0.12	0.20	0.44	0.20	-0.27	-0.68

Maturity

Ensemble, mean standarisation

Horizon	3M	6M	12M	2Y	3Y	5Y	7Y	10Y
3	0.91	0.94	0.95	0.96	0.96	0.95	0.96	0.94
6	0.92	0.94	0.95	0.95	0.94	0.92	0.91	0.88
9	0.95	0.95	0.96	0.96	0.94	0.91	0.88	0.84
12	0.96	0.96	0.97	0.96	0.94	0.88	0.85	0.80
24	1.00	1.00	1.01	1.00	0.98	0.90	0.84	0.78
36	1.00	0.99	0.99	1.02	1.05	1.02	0.97	0.90

Maturity

Source: Author’s calculations.

Equally weighted ensemble — Graph 7

Ensemble vs naive

Horizon	3M	6M	12M	2Y	3Y	5Y	7Y	10Y
3	0.88	0.91	0.98	1.06	1.11	1.16	1.13	1.12
6	0.86	0.92	0.99	1.08	1.11	1.18	1.20	1.20
9	0.84	0.90	0.97	1.04	1.07	1.14	1.18	1.21
12	0.86	0.90	0.95	0.99	1.01	1.08	1.13	1.17
24	0.88	0.89	0.88	0.88	0.87	0.89	0.92	0.97
36	0.83	0.83	0.81	0.79	0.78	0.78	0.82	0.87

Maturity

Ensemble, z-score standardisation

Horizon	3M	6M	12M	2Y	3Y	5Y	7Y	10Y
3	-0.59	-0.48	-0.37	-0.34	-0.32	-0.28	-0.30	-0.34
6	-0.54	-0.52	-0.44	-0.35	-0.31	-0.26	-0.28	-0.30
9	-0.50	-0.51	-0.44	-0.33	-0.29	-0.25	-0.27	-0.28
12	-0.47	-0.45	-0.40	-0.36	-0.34	-0.29	-0.29	-0.30
24	-0.36	-0.33	-0.34	-0.41	-0.47	-0.50	-0.49	-0.46
36	-0.41	-0.40	-0.39	-0.49	-0.60	-0.72	-0.71	-0.64

Maturity

Ensemble, mean standarisation

Horizon	3M	6M	12M	2Y	3Y	5Y	7Y	10Y
3	0.91	0.94	0.96	0.97	0.97	0.96	0.96	0.96
6	0.92	0.94	0.96	0.96	0.96	0.96	0.95	0.95
9	0.93	0.95	0.96	0.97	0.96	0.96	0.95	0.95
12	0.94	0.96	0.97	0.97	0.96	0.96	0.95	0.94
24	0.97	0.97	0.97	0.97	0.96	0.95	0.94	0.93
36	0.96	0.96	0.96	0.95	0.94	0.92	0.91	0.90

Maturity

Source: Author’s calculations.

The ensemble could now be used to make the forecasts, charts and statistics required by an investment committee. The next step would involve fitting each model on the complete data set and generating out of sample forecasts. Once we had all the point forecasts, we multiplied them by each of the computed weights to make the final predictions of the ensemble (see Table 1).

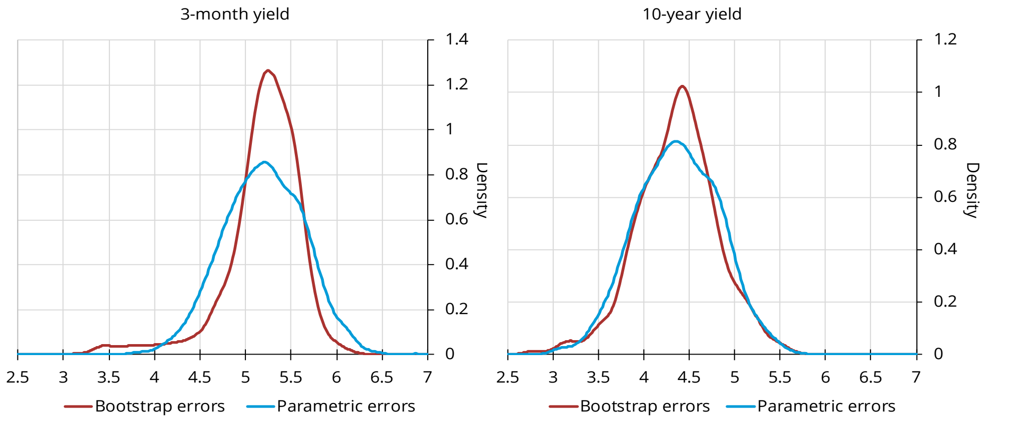
Optimally weighted ensemble forecasts of yield (out of sample) Table 1

Maturity / horizon	3M	6M	12M	2Y	3Y	5Y	7Y	10Y
3	5.19	5.07	4.87	4.63	4.49	4.34	4.34	4.36
6	4.82	4.79	4.73	4.55	4.48	4.33	4.32	4.37
9	4.53	4.57	4.56	4.54	4.50	4.33	4.32	4.36
12	4.38	4.45	4.51	4.56	4.55	4.40	4.36	4.39
24	4.04	4.23	4.34	4.50	4.59	4.61	4.67	4.67
36	3.89	4.09	4.23	4.47	4.60	4.70	4.75	4.66

Source: Author's calculations.

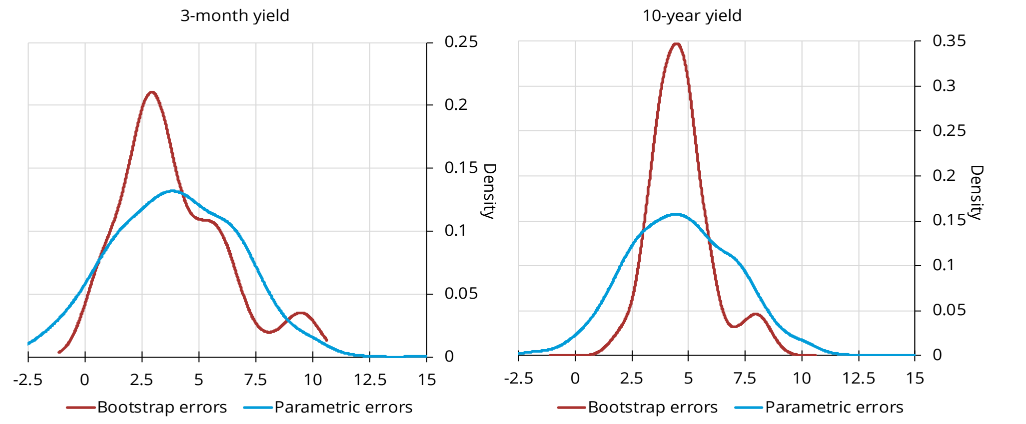
Utilising the weights and historical errors of each model, we proceeded to create a probability distribution of the predictions using a weighted kernel density estimation (KDE). These errors were used to construct model-specific confidence intervals, employing both parametric and non-parametric techniques. On one hand, in the parametric approach, we calculated the standard deviation of the errors to construct a Gaussian distribution with a mean equal to the forecast and a standard deviation equal to the standard deviation of the historical errors (model-maturity-horizon). On the other hand, we employed a bootstrapping approach to generate non-parametric errors. In this method, errors were demeaned and bootstrapped errors were subsequently generated by adding the demeaned errors to the forecasted values. These sub-distributions were then assigned weights based on both equal and optimal weights. By combining the data points and their weights, we computed a weighted KDE, providing insights into the underlying distribution of forecasted yield curve values (see graphs 8 and 9).

Optimally weighted KDE: forecast horizon three months — Graph 8



Source: Author’s calculations.

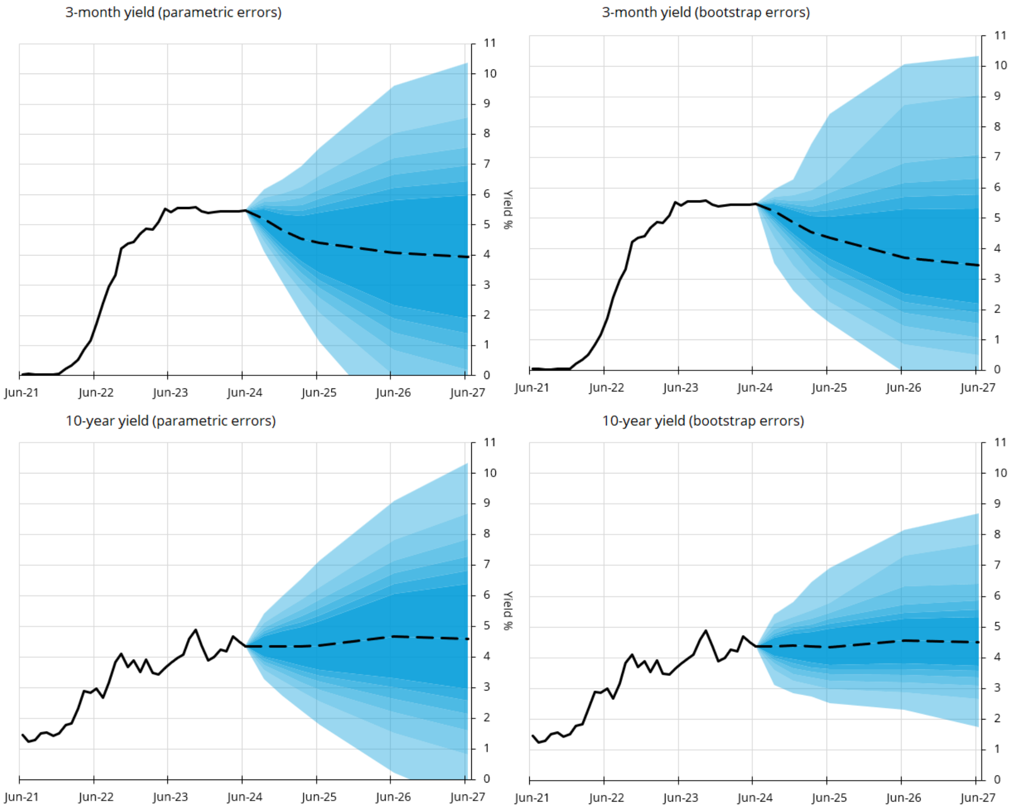
Optimally weighted KDE: forecast horizon 36 months — Graph 9



Source: Author’s calculations.

Following this, we calculated the cumulative density forecast, enabling the assessment of predefined percentiles ranging from 0.01 to 0.99. These percentiles aided in understanding the range of potential outcomes and associated uncertainties in yield curve forecasts. Additionally, we utilised fan charts to visualise the output results across a time slice (varying horizons with just one interest rate) and across a horizon slice (different rates for a single horizon) (see graphs 10 and 11). These visualisations provided a comprehensive view of the forecasted yield curve values and their corresponding levels of uncertainty, thereby facilitating decision-making processes and risk assessment.

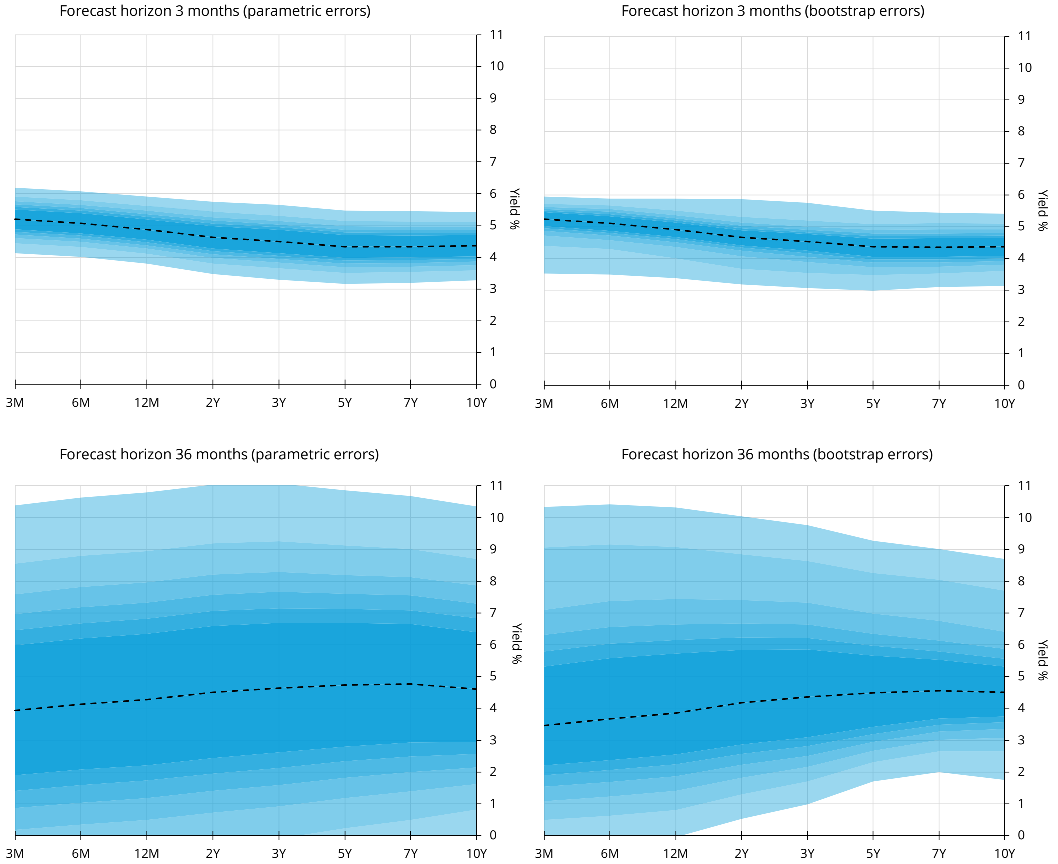
Fan charts: time slice Graph 10



Source: Author's calculations.

Fan charts: horizon slice

Graph 11



Source: Author's calculations.

Concluding remarks

This research successfully implemented and evaluated an ensemble model for forecasting the US Treasury yield curve across various maturities and horizons. The framework allowed for the agile inclusion and exclusion of models, enabling the construction of confidence intervals and probability distributions for ensemble forecasts. The varying performances exhibited by different models, depending on the forecast horizon and maturity, highlight the need for a horizon-maturity-specific ensemble.

Using a Bayesian approach enhanced by a decay factor for ensemble optimisation, we developed an ensemble model that shows promise in improving both the accuracy and robustness of yield curve forecasts. This methodology is well suited for integration into the investment process, providing valuable insights into the uncertainty surrounding predicted yield curve values and supporting financial decision-making.

We acknowledge that the models used in this initial approach may be correlated or lack sufficient diversity. In subsequent steps, we aim to refine the framework by exploring which forecasts should be included in the combination. For instance, an ex ante forecast pooling strategy based on clustering methods could be employed (Kourentzes et al (2019)), or the optimisation process could be adapted to shrink the weights of similar models to zero.

Future research can explore additional weighting schemes and optimisation techniques, integrate exogenous variables with more complex dynamics, investigate the use of deep learning models for yield curve forecasting and extend the ensemble framework to other financial forecasting tasks. Addressing these aspects will further refine and improve the ensemble approach, leading to more accurate forecasts, better risk management and enhanced financial decision-making.

References

Aiolfi, M, C Capistrán and A Timmermann (2010): "Forecast combinations", *Banco de México Working Papers*, no 2010-04, June.

Caldeira, J, G Moura and A Santos (2016): "Predicting the yield curve using forecast combinations", *Computational Statistics & Data Analysis*, vol 100, August, pp 79–98, doi.org/10.1016/j.csda.2014.05.008.

Choi, D, W Jiang and C Zhang (2025): "Alpha go everywhere: machine learning and international stock returns", SSRN Electronic Journal, April, doi.org/10.2139/ssrn.3489679.

Côté, P-O, A Nikanjam, N Ahmed, D Humeniuk and F Khomh (2023): "Data cleaning and machine learning: a systematic literature review", *ArXiv*, 2310.01765.

Diebold, F (2024): "Forecasting in Economics, Business, Finance and Beyond", Department of Economics, University of Pennsylvania, www.sas.upenn.edu/~fdiebold/Teaching221/Forecasting.pdf.

Diebold, F and C Li (2006): "Forecasting the term structure of government bond yields", *Journal of Econometrics*, vol 130, no 2, pp 337–64.

Elliott, G and A Timmermann (2016): "Forecasting in economics and finance", *Annual Review of Economics*, vol 8, pp 81–110.

Gu, S, B Kelly and D Xiu (2019): "Empirical asset pricing via machine learning", *Review of Financial Studies*, vol 33, no 5, pp 2223–63.

Hanauer, M and T Kalsbach (2023): "Machine learning and the cross-section of emerging market stock returns", *Emerging Markets Review*, vol 55, June, 101022.

Jansen, S (2020): *Machine learning for algorithmic trading: predictive models to extract signals from market and alternative data for systematic trading strategies with Python* (2nd edition), Packt Publishing.

Joia, A and J Coche (2010): "Framework and process for strategic asset allocation in central bank foreign exchange reserves management", in A Berkelaar, J Coche and K Nyholm (eds), *Central bank reserves and sovereign wealth management*, Palgrave Macmillan, pp 27–52, doi.org/10.1057/9780230250819_2.

Kolanovic, M and Krishnamachari (2017): *Big data and AI strategies: machine learning and alternative data approach to investing*, JP Morgan Chase & Co.

Kourentzes, N, D Barrow and F Petropoulos (2019): "Another look at forecast selection and combination: evidence from forecast pooling", *International Journal of Production Economics*, vol 209, pp 226–35.

Lessmann, S., Baesens, B., Seow, H.-V., and Thomas, L.C. (2015): "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research", *European Journal of Operational Research*, vol. 247, May, pp. 124–136, doi.10.1016/j.ejor.2015.05.030.

Nyholm, K (2015): "A rotated dynamic Nelson-Siegel model with macro-financial applications", *European Central Bank Working Papers*, no 1851 September.

Petropoulos, F, et al (2022): "Forecasting: theory and practice", *International Journal of Forecasting*, vol 38, no 3, pp 705–871.

Rasekhschaffe, K C and R Jones (2019): "Machine learning for stock selection", *Financial Analysts Journal*, vol 75, no 3, pp 70–88.

Sezer, O B, M U Gudelek and A M Ozbayoglu (2020): "Financial time series forecasting with deep learning: a systematic literature review: 2005–2019", *Applied Soft Computing*, vol 90, May, 106181, doi.org/10.1016/j.asoc.2020.106181.

Shah, A, S Paturi and S Chava (2023): "Trillion dollar words: a new financial dataset, task & market analysis", in A Rogers, J Boyd-Graber and N Okazaki (eds), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, vol 1: long papers, Association for Computational Linguistics, pp 6664–79.

Shi, S, R Tse, W Luo, S D'Addona and G Pau (2022): "Machine learning-driven credit risk: a systemic review", *Neural Computing and Applications*, vol 34, pp 14327–39, doi.org/10.1007/s00521-022-07472-2.

Swinkels, L and T Hoogteijling (2022): *Forecasting stock crash risk with machine learning*, white paper, Robeco.

West, J and M Bhattacharya (2016): "Intelligent financial fraud detection: a comprehensive review", *Computers & Security*, vol 57, March, pp 47–66.