Universidad del Rosario | Facultad de Economía

SERIE
DOCUMENTOS
DE TRABAJO

No. 271

Julio de 2021

Forecasting Dynamic Term Structure Models with Autoencoders

Carlos Castro-Iragorri

Julián Ramírez

Universidad del Rosario, Facultad de Economía
Cl. 12c #4-59. Bogotá, D. C., Colombia
PBX: (031) 2970200 ext. 4163

facultadeconomia@urosario.edu.co
https://www.urosario.edu.co/Facultad-de-Economia/Inicio/
https://repository.urosario.edu.co/handle/10336/20291

Forecasting Dynamic Term Structure Models with Autoencoders

Carlos Castro-Iragorri[1a], Julian Ramirez[a]

[a] *Universidad del Rosario, Colombia*

Abstract

Principal components analysis (PCA) is a statistical approach to build factor models in finance. PCA is also a particular case of a type of neural network known as an autoencoder. Recently, autoencoders have been successfully applied in financial applications using factor models, Gu et al. (2020), Heaton and Polson (2017). We study the relationship between autoencoders and dynamic term structure models; furthermore we propose different approaches for forecasting. We compare the forecasting accuracy of dynamic factor models based on autoencoders, classical models in term structure modelling proposed in Diebold and Li (2006) and neural network-based approaches for time series forecasting. Empirically, we test the forecasting performance of autoencoders using the U.S. yield curve data in the last 35 years. Preliminary results indicate that a hybrid approach using autoencoders and vector autoregressions framed as a dynamic term structure model provides an accurate forecast that is consistent throughout the sample. This hybrid approach overcomes in-sample overfitting and structural changes in the data.

Keywords: autoencoders, factor models, principal components, recurrent neural networks.
JEL: C45, C53, C58

1. Introduction

The term structure of interest rates is the relationship between interest rates or bond yields and different terms or maturities. The term structure of interest rates is also known as the yield curve, and it plays a central role in

[1]Corresponding author: carlos.castro@urosario.edu.co.

economic and financial analysis. For example, the term structure reflects expectations of market participants about future changes in interest rates and their assessment of monetary policy conditions beyond the direct relationship between the inflation target rate, the policy rate and economic activity.
Term structure modelling is also important for practitioners for pricing and risk management. Fixed income and derivatives markets rely on the information on the term structure of the most liquid securities in particular sovereign bonds to mark-to-market the value of other (less liquid securities) and portfolios. In risk management term structure models provide the tools to obtain stress scenarios base on the historical and simulated process (Engle et al., 2017).
Term structure models are built with different objectives and it is naive to think that there is an all-purpose approach. For forecasting the academic literature has identified that some models are better than others. The Nelson-Siegel model (Nelson and Siegel, 1987) is a statistical approach that provides a parsimony specification to capture the differences in rates along the curve (for different maturities). Its implementation in one or two stages gives the temporal variation of the factors maintaining the factor loading's constant over time. The specification of the model and the estimation methods provide a simple implementation, which is why it turns out to be a successful model for policymakers and central banks. Although the Nelson-Siegel model is not arbitrage-free, (Christensen et al., 2011) propose a representation of the arbitrage-free model. However, it remains unclear if no-arbitrage restrictions improve statistical validity and therefore, some empirical applications showed that different variations of the Nelson-Siegel model provide a better in-sample and out-of-samples fit of the yields than the class of arbitrage-free affine term structure models.
There is of course a larger class of purely statistical models for modelling the term structure, Diebold and Li (2006) provide a systematic evaluation of different reduced form (non-arbitrage free) approaches for yield curve forecasting. Some of the approaches considered are univariate and multivariate autoregressive models, forward curve regressions, principal component analysis and of course the Nelson-Siegel dynamic three-factor model. Their results indicate that for the U.S. data in the sample from January 1985 to December 2000, the Nelson-Siegel dynamic factor model is more accurate than the other models considered for a one-year ahead forecast. The principal component (PCA) approach to build and forecast the yield curve is a popular approach in the financial industry (Redfern and McLean,2014). Both the Nelson-Siegel

type models and PCA map the maturities to a reduced set of factors (level, slope and curve) that provide a parsimonious set of time series to be forecast using classical time series models such as autoregressive or vector autoregressive models.
Neural network models that have been successful at classification task and natural language processing are stating to be used both in finance de Prado (2018), Dixon et al. (2020), Heaton and Polson (2017) and time series forecasting Hewamalage et al. (2021), Borovykh et al. (2017), Zhang and Berardi (2001).
Autoencoders are a type of neural network model where the input and output variables are the same. When there are fewer neurons in the layers than the number of variables these models may be used for dimension reduction. Gu et al. (2020), proposed a flexible factor model based on autoencoders for asset pricing. This approach overcomes some restrictions in traditional factor models in finance: first, it allows for non-linear relations between the factor loadings and a set of covariates related to the individual asset returns. Second, the previous mapping is estimated jointly with the latent factors and hence these factors are not only a function of the asset returns. Third, the complexity of the model does not impede is use on a large cross-section of assets (30,000).

In this paper, we use autoencoders to forecast the term structure of interest rates. Since PCA is a particular case of a linear autoencoder, it is straight forward to relate this methodology to dynamic term structure models which are a factor model that has been successfully used to model and forecast the yield curve. From the statistical point of view, the dynamic factor models have a state-space representation. In the context of a one-layer autoencoder, the decoder provides the measurement equation: a mapping between the factors and the yields. The encoder provides a linear or non-linear mapping between the yields and the factors. In between the encoder and decoder, a state equation based on a vector aggressive model can be used to provide a forecast of the factors. This hybrid model is a generalization of PCA and can provide many extensions depending on the layers of the neural network and the use of linear or non-linear activation functions. One important difference between autoencoders and PCA, is that the former provides an exact mapping to the level, slope and curvature factors versus choosing the first three principal components.
We empirically test the forecasting accuracy of this model using in- sample and out-of-sample forecasting exercises. The data used is 35 years (1985-

2020) of monthly data from the U.S. synthetic yields build from data provided by CRSP, U.S. Treasury and the Federal Reserve. This historical data is interesting because it captures important structural changes such as the financial crisis and the COVID pandemic. We compare the performance of the proposed model to other reduced-form statistical dynamic factor models such as the Nelson-Siegel three-factor model, recurrent neural networks and the random walk benchmark.
Our preliminary results show that the three-factor autoencoder model provides more accurate results than the Nelson-Siegel model and that more complicated or deep networks do not provide a significant advantage over a linear autoencoder and in particular PCA. We find that recurrent neural networks applied to this use case provide inaccurate forecast and further research is required to design model that is capable of forecasting multivariate related data that is subject to structural changes. Hybrid models, combining neural networks and vector autoregressions such as the one that we proposed within the framework of dynamic factor models are a viable alternative to get the best of both worlds. However, unlike traditional time series models for forecasting like autoregressions and static PCA, neural networks are computationally expensive and hence a cost-benefit analysis still favours traditional time series analysis. These results are consistent at different forecast horizons (1, 6 and 12 months) and also across maturities in the term structure.

Our application also contributes to the literature on term structure modeling using neural networks Suimon et al. (2020), Kirczenow et al. (2018), Kondratyev (2018). The paper closes to our application is Suimon et al. (2020) that apply autoencoders to model the yield curve of Japanese sovereign bonds. They find that the level, slope and curvature obtained from a Nelson-Siegel model, PCA and a non-linear autoencoder have similar behaviour. They do not provide estimates of the errors from an out-of-sample forecasting exercise but build successfully trading strategies based on the signals obtained from the yield curve forecast. The investment simulation shows a higher capital gain form using a recurrent neural network (the long short term memory model, LSTM), than a strategy that used a linear autoencoder or a vector autoregression to get the investment signals. They claim the LSTM provides better prediction accuracy but do not provide the result to support their claims.

The remainder of the paper is structured as follows. Section 2 gives a brief introduction to autoencoders. Section 3 provides an account of the dynamic factor models used for forecasting the term structure and proposed their use

with autoencoders as a generalization of PCA. Section 5 describes the U.S. synthetic yield data. Sections 6.1 and 6.2 presents the forecasting exercise and results using an in-sample and an out-of-sample approach, respectively. Section 7 concludes.

2. Autoencoders

Autoencoders are a particular type of neural network where the input or features and the output variables are the same, therefore we can refer to them as the observable variables. In the simplest case, the network has two layers and if the number of neurons is less than the observed variables then we have a bottleneck effect that implies a dimension reduction from the observable variables[2]. These neurons are non-observable latent variables (figure 1).
If the objective is to obtain a forecast of the observable variables then it is a multivariate regression problem as opposed to a classification problem where there are also multiple outcomes but each of the outcomes represents a class. Let $Y_t(\tau)$ is a matrix of τ time-varying observable variables, where $y_{t,j} \in \mathbb{R}$ where $j = 1, \ldots, \tau$. A simple linear autoencoder model is,

$$Y_t(\tau) = W^{(2)} Z_t + b^{(2)}$$
$$Z_t = \tanh(W^{(1)} Y_t(\tau) + b^{(1)})$$

where $W^{(2)}$ and $W^{(1)}$ denote the weight matrices in the second and first layers, respectively. $b^{(2)}$ and $b^{(1)}$ are the bias parameters. The weights and the biases are dynamically adjusted so as to reconstruct the original data. The neurons are in Z_t a K-dimensional matrix of time-varying latent variables. As shown in figure (1) as long as $K << \tau$ we have by design a mechanism to generate dimension reduction to the neurons (latent variables) from the observable variables. The model is non-linear because the hyperbolic tangent function which also provides the support of the neurons, that it $\tan(x) \in [-1, 1]$.
In this simple autoencoder the first layer (1) acts a encoder transformation from the observable variables to the neurons and the second layer (2) as a decoder transformation from the neurons to the observable variables. In a more general model with a larger number of layers such as a deep neural networks there are as many intermediate encoder transformations as the number of layers.

[2]Neurons also known as hidden units in the neural network literature.

Baldi and Hornik, 1989 show that principal component analysis is a particular case of a linear autoencoder,

$$Y_t(\tau) = W^{(2)} Z_t + b^{(2)}$$
$$Z_t = W^{(1)} Y_t(\tau) + b^{(1)}.$$

The principal components are obtained using the solution to the optimal reconstruction problem,

$$min_{W^{(1)},\mathbf{b}^{(1)},W^{(2)},\mathbf{b}^{(2)}} \; || \; \mathbf{Y} - W^{(2)}(W^{(1)}\mathbf{Y} + \mathbf{b}^{(1)}\mathbf{1}'_N) + \mathbf{b}^{(2)}\mathbf{1}'_N \; ||^2_F$$

This problem can be re-written such that the weights represent a projection matrix that is used to project the original variables and obtain the lower dimensional neurons. In the case of principal components analysis (PCA) the weights have a singular value decomposition that provides the eigenvalues and vectors requires to obtain and select the principal components. There is however important difference between the more general result from linear autoencoders and principal component analysis. First, the principal components are orthogonal but the neurons are not. Second, PCA has a natural ordering where the first component captures the largest variance; the neurons do not have a natural ordering. Finally, the neurons obtained from the linear autoencoder contain all of the information, represented in the variance-covariance matrix. Whereas in PCA, is customary to keep the first principal components given that jointly they represent the majority of the information. Keeping all of the information from the original data exactly would require keeping all of the principal components but this would defeat the goal of dimension reduction. This last point is one of the main reasons to explore linear autoencoders as an alternative to PCA because in principle you can keep all of the information from the original data in just a few latent variables then this would provide ex-ante a better reconstruction than PCA using the in-sample data.

3. Dynamic term structure models

Dynamic term structure models have particular advantages over alternative approaches to derive a functional approximation for the yield curve. These models are introduced within the forecasting context in Diebold and Li (2006). This paper uses a Nelson-Siegel yield curve and combined with standard univariate and multivariate time series models as a mechanism to

perform forecast over one, six and twelve months forecast horizon. The forecast performance compared to standard benchmarks (e.i. random walk, PCA) is particularly promising for the long term yields. These models are also known as purely statistical models of the yield curve because they do not specify any pricing relation. Alternatively, affine term structure models enforce no-arbitrage restrictions within the process of constructing and forecasting the yield curve. However, there are mixed results regarding their benefit in forecasting (Engle et al.,2017). Therefore, since it is unclear whether the no-arbitrage restrictions are sufficiently strong or accurate to improve forecast performance we do not consider these type of models.
We are interested in forecasting and therefore we need a model that provides a representation of the conditional mean of the yields along the curve. We can use the state space representation to provide a generic form of a dynamic term structure model for the purpose of forecasting,

Proposition 1: Every dynamic term structure model has a state-space representation (measurement and state transition equation),

$$y_{t+1}(\tau) = F_t(\tau)B_t + \varepsilon_{t+1}(\tau) \tag{1}$$

$$B_t = \Phi B_{t-1} + v_t \tag{2}$$

where $F_t(\tau)$ are the predetermined or time invariant ($F_t(\tau) = F(\tau)\forall t$) factor loadings. In addition, B_t are lower $K-$dimensional $K << \tau$ time varying factors. These time varying factors, through the state equation, are forecastable and provide a mechanism to forecast the variables of interest $y_{t+1}(\tau)$ given a set of loading factors and $E_{t-1}(B_t) = \hat{\Phi}B_{t-1}$.
There are many advantages to having this representation, in particular, if there is a mechanism to label the factors as observable then it is possible to split the measurement problem and the forecasting problem and hence it is possible to use simpler estimation approaches. A second advantage comes from dimension reduction since the different methods provide the lower-dimensional set of factors that can forecast using the state equation and then recover the higher dimensional observable. In this particular case, we can recover the complete yield curve no only the observable knots.
We can show that most of the models used in the term structure literature and the autoencoders have this state space representation.
In the case of the Nelson-Siegel(1987) three factor model,

$$y_{t+1}(\tau) = \beta_{1,t}1 + \beta_{2,t}(\frac{1-\mathrm{e}^{-\lambda_t\tau}}{\lambda_t\tau}) + \beta_{3,t}(\frac{1-\mathrm{e}^{-\lambda_t\tau}}{\lambda_t\tau} - \mathrm{e}^{-\lambda_t\tau}) + \varepsilon_{t+1}(\tau) \tag{3}$$

where $B_t = (\beta_{1,t}, \beta_{2,t}, \beta_{3,t})$. The Nelson-Siegel is a polynomial approximation using exponential functions for the factor loadings $F_t(\tau) = (1, \frac{1-e^{-\lambda_t \tau}}{\lambda_t \tau}, \frac{1-e^{-\lambda_t \tau}}{\lambda_t \tau} - e^{-\lambda_t \tau})$. These factor loading are usually known as the level, slope and curvature, respectively. λ_t is an exponential decay parameter providing a way to fine tune the yield curve. In particular, low (high) values of λ_t give more weight to a fetter fit of the longer (shorter) part of the curve. The exponential decay parameter can be time-varying or fixed over time $\lambda_t = \lambda \forall t$. A fixed parameter implies that it can be considered as an exogenous tuning parameter, in addition Diebold and Li (2006) mention that by fixing this parameter the factors B_t can be estimated using the cross section of yield at every point in time using ordinary least squares. In a second stage these factors can be modeled as a Vector Autoregressive Model (the state equation) to forecast their value (VAR(1)). Even though can be estimated along with the factors, however in this case the factors become non-observable and estimation must be done recursively using the Kalman-Filter. We prefer to keep as a tuning parameter where we explore through a grid search to determine its effects on forecast performance.

Extensions to a four-factor and two decay parameters is proposed in Svensson (1994) and evaluated by Pooter (2007). The former finds benefits in forecasting performance (in-sample and out-of-sample) when using these extensions to the Nelson-Siegel three-factor model, outperforming some of the usual benchmarks(random walk and vector autoregression models).

Another approach that is very important for practitioners (Redfern and McLean,2014) is to use principal components analysis. In order to obtain the equivalent set of factors as in the Nelson and Siegel three factor model then we only consider the first three principal components $(Z_{i,t})$,

$$y_{t+1}(\tau) \approx Z_{1,t} V_1(\tau) + Z_{2,t} V_2(\tau) + Z_{3,t} V_3(\tau) + \varepsilon_{t+1}(\tau) \tag{4}$$

where the set of factors are determined by the first three principal components $B_t = (Z_{1,t}, Z_{2,t}, Z_{3,t})$. The factor loadings are given by the matrix of the first three eigenvectors $V(\tau) = (V_1(\tau), V_2(\tau), V_3(\tau))$.The latter share similar patterns as the Nelson-Siegel factor loading hence they can also be labeled as the: level, slope and the curvature.

Estimation and forecasting of this model can also be accomplished as a two-step process. In the first step, the variance-co variance of the observed yields is used to derive the eigenvalue decomposition and obtain the principal components. In a second stage these principal components are forecast using a

using a vector autoregressive model (VAR(1)).
If additional principal components are considered then additional factors can be accommodated. For forecasting, the number of principal components or factors to keep can be considered also as an exogenous tuning parameter.

Finally, we can provide a model that uses autoencoders and a state space representation based on the neurons as the set of factors, where we will have a state space representation and an auxiliary equation. As was the case of principal components we restrict to the case of three neurons or hidden units $Z_{i,t}$,

$$y_{t+1}(\tau) = Z_{1,t}W_1^{(2)}(\tau) + Z_{2,t}W_2^{(2)}(\tau) + Z_{3,t}W_3^{(2)}(\tau) + b^{(2)}\mathbf{1}(\tau) + \varepsilon_{t+1}(\tau) \quad (5)$$

$$Z_{i,t} = \tanh(W_i^{(1)}y_t(\tau) + b_i^{(1)}) \quad (6)$$

where the set of factors are determined by the three neurons $B_t = (Z_{1,t}, Z_{2,t}, Z_{3,t})$ that are imposed by design. The factor loadings are given by the weight matrices $W_i^{(2)}$ associated with each of the neurons at the decoder level (2). Again there will be some similarities to the behavior of these weight matrices that will allow us to label them as the level, slope and curvature, however this will not be in any particular order, because as mentioned in the previous section (2), unlike principal components, autoencoders do not have a natural ordering.
Forecasting with this type of model will also be a step procedure. First, we use the neural network to estimate the time-varying neurons, in other words, the factors B_t. Second, we estimate a vector autoregression (VAR(1)) to obtain a forecast of their values. Finally, we use the decoder, in particular, the estimated weights and the bias, part neural network to recover the forecast of the yields. Under this setup, we have a measurement equation given by the decoder 6, as a state equation given by the VAR(1) on the neurons and a deterministic auxiliary equation given by the encoder 6. This model with one layer, three neurons and a non-linear hyperbolic tangent as an activation function is the basic autoencoder model (NA3). A particular case of this model is a linear autoencoder where we replace the non-linear activation function with a linear function (LA3). In addition, we consider a three-layer, deep autoencoder model (DA3). In this model forecasting on the three neurons is performed at the intermediate layer (figure 2), whereas the two additional layers provide the encoder and a layer before the decoder (measurement equation), therefore you have an auxiliary equation in this state-space representation.

Neural networks used in this manner have an important drawback and is the fact that the weights and biases at the decoder level are not updated optimally for the changes in the vector autoregression part of the model. This is unfortunately the cost of estimating and forecasting the model in various steps. As mentioned previously, the dimension reduction implicit in the autoencoders provides a benefit from the lower-dimensional set of factors to be forecast. The forecast performance will provide information regarding the benefits of using this two-step approach for the autoencoders. An alternative approach is to use recurrent neural networks to forecast the observable yields. In the next section, we introduce these alternative models.

4. Recurrent Neural Networks

Recurrent neural networks (RNN) provide an architecture for time series analysis. More recently, these approaches are starting to show significant progress over the more traditional time series (ARIMA, exponential smoothing) in univariate forecasting. Hewamalage et al. (2021) provide a comprehensive survey of the forecasting performance of RNN for different datasets, architectures and the different configurations of these type of models. They find some configurations that successfully capture different characteristics of the data, for example, seasonal patterns. However, they also indicate that the computational cost is much higher than the traditional time series benchmarks. Based on their experiments and unlike previous results they find that these complex models can outperform simple statistical benchmarks within the context of univariate forecasting and leave an open question regarding additional challenges for multivariate forecasting.
The most popular RNN models are the Elman recurrent unit (ERNN), Gated recurrent unit (GRNN) and Long Short Term Memory (LSTM) with or without peephole. To use these models to forecast the term structure of interest rate and more precisely the observed yields we make the following considerations. First, these models can also be considered as encoders since the output and the input variables are the same. However, there is a time lag between the input and the output; this lag is equivalent to the lag order in an AR model. Second, there is always an affine neural layer as a decoder. Independent from the number of layers of the neural network, in other words, the depth of the network, this top layer must be linear to recover the forested yield in the expected support. Finally, these type of models do not have a bottleneck design for dimension reduction, quite the opposite they tend to

have a large number of hidden units and additional units to accommodate long term dependence.
The ERNN(1) has one hidden layer that is a function of the current yields and the lagged hidden unit,

$$y_{t+1}(\tau) = Z_t W^{(2)}(\tau) + b^{(2)}\mathbf{1}(\tau) + \varepsilon_{t+1}(\tau) \tag{7}$$

$$Z_t = \tanh(y_t(\tau)W_y^{(1)} + Z_{t-1}W_z^{(1)} + b^{(1)}) \tag{8}$$

The basic ERNN model suffers from the vanishing gradient problem where the weights are not adjusted properly to account for long term dependence. This is a problem that has been well studied in natural language processing. To overcome such problems the GRNN(1) introduces one latent variable and a gate mechanism applied to a smoothed version of the latent variable Z_t. This smoothing mechanism is analogous to an exponentially weighted moving average, allowing for an autocorrelation across the hidden units that decays over time (Hyndman et al.,2008).

$$y_{t+1}(\tau) = Z_t W^{(2)}(\tau) + b^{(2)}\mathbf{1}(\tau) + \varepsilon_{t+1}(\tau) \tag{9}$$

$$Z_t = u_t \circ \tilde{Z}_t + (1 - u_t) \circ Z_{t-1} \tag{10}$$

$$\tilde{Z}_t = \tanh(y_t(\tau)W_{z,y}^{(1)} + Z_{t-1}W_{z,z}^{(1)}r_t + \mathbf{1}b_z^{(1)}) \tag{11}$$

$$r_t = \sigma(W_{r,y}^{(1)}y_t(\tau) + W_{r,z}^{(1)}Z_{t-1} + \mathbf{1}b_r^{(1)}) \tag{12}$$

$$u_t = \sigma(W_{u,y}^{(1)}y_t(\tau) + W_{u,z}^{(1)}Z_{t-1} + \mathbf{1}b_u^{(1)}) \tag{13}$$

where u_t and r_t denote the update and reset gate respectively. The update gate determines how much of the candidate hidden state contributes to the current hidden state (level of smoothing). The reset gate decides how much of the previous hidden state contributes to the candidate state to the current step (temporal feedback or autocorrelation). $\sigma(x) \in (0, 1)$ is an activation function.
The Long Short Term Memory targets the long term dependence problem by introducing two latent variables the traditional hidden state (present in ERNN and GRNN) for the short memory component Z_t and an additional latent variable unknown as the cell state C_t that corresponds to the long-term memory component.

$$y_{t+1}(\tau) = Z_t W^{(2)}(\tau) + b^{(2)}\mathbf{1}(\tau) + \varepsilon_{t+1}(\tau) \tag{14}$$
$$Z_t = o_t \circ \tanh(C_t) \tag{15}$$
$$C_t = i_t \circ \tilde{C}_t + f_t \circ C_{t-1} \tag{16}$$
$$\tilde{C}_t = \tanh(y_t(\tau)W^{(1)}_{c,y} + Z_{t-1}W^{(1)}_{c,z}r_t + \mathbf{1}b^{(1)}_c) \tag{17}$$
$$f_t = \sigma(W^{(1)}_{f,y}y_t(\tau) + W^{(1)}_{f,z}Z_{t-1} + \mathbf{1}b^{(1)}_f) \tag{18}$$
$$o_t = \sigma(W^{(1)}_{o,y}y_t(\tau) + W^{(1)}_{o,z}Z_{t-1} + \mathbf{1}b^{(1)}_o) \tag{19}$$
$$i_t = \sigma(W^{(1)}_{i,y}y_t(\tau) + W^{(1)}_{i,z}Z_{t-1} + \mathbf{1}b^{(1)}_i) \tag{20}$$

where f_t, o_t and i_t are the forget, output and input gates respectively. The forget modulates how much in of the past information C_{t-1} to retain in the current cell state C_t. The input gate modulates how much of the current context (the candidate cell) to propagate to the current cell state C_t.

These three RNN models provide a dynamic system for forecasting with some similarities and differences to a state-space representation. It is not difficult to see that the affine decoder function is equivalent to the measurement equation. In addition, there is a system of equations that is deterministic (the weights and bias are t-measurable) and with non-linear activation functions. The most important distinction is that these model have only a single source of error. This single source of error is needed to use a backpropagation algorithm to minimize the error and train the neural network. Some state-space models represent a system with a single source of error, for example, an additive exponential moving average that can be represented as a linear non-deterministic dynamic system with a single source of error (Hyndman et al., 2008).

5. Data

We estimate the models using monthly synthetic U.S. yields from November 1985 to December 2020. These synthetic yields are obtained from multiple sources and used together to obtain sixteen observable maturities through the sample and each of them from the same source. The yields from one to six months of maturity are obtained from the Fama CRSP Treasury Bill Files. The one to five-year bonds yields is obtained from the Fama CRSP

zero-coupon files. The seven, ten, twenty and thirty-year bond yields are obtained from the U.S. Treasury constant maturity yields. The fifteen and twenty-five-year maturity zero-coupon yields are obtained from the H.15 data release of the Federal Reserve Board of Governors. These synthetic yields have at least two drawbacks. First, they are not directly observable as opposed to using the bond prices (Andreasen et al., 2019). Second, some of these are the result of interpolation therefore there is an unobservable model risk. However, these synthetic yields are the most common data sources used in empirical studies especially if a long historical time series is required (Doshi et al.,2020).
Figure 3 shows a three-dimensional plot of the observed yields through the sample. There is an important variation throughout the sample that reflects the monetary policy and economic conditions in the U.S. economy in the last thirty-five years. From the figure, it is clear that the historical behaviour indicates a strong variation of different intensities across the level, slope and curvature in the yield surface. The reason for the success of the family of three-factor models that we will be testing in the next section is that in a parsimonious manner the dynamic factor model provides a powerful tool to capture these variations.

6. Forecast Performance

6.1. In-sample fit

In this section, we present the results regarding the in-sample fit of the sixteen maturities for the two groups of models considered, the dynamic term structure models (Table 1) and the recurrent neural networks (Table 2).
The results indicate that autoencoders, including principal components as a special case, outperform the basic three-factor Nelson and Siegel model at most of the maturities. The linear autoencoder with three hidden units and the dynamic factor model that used the first three principal components have the best overall performance and very similar results. This results empirically confirms the results in section 2 and indicates that the dimension reduction in principal components requires an arbitrary number of principal component to keep as opposed to the autoencoders where the original data is exactly matched to the lower dimensional hidden units. However, in this particular case, it is also important to consider that there is substantial evidence confirming the relevance and sufficiency of the three-factor: level, slope and curvature for modelling the yield curve as opposed to considering

more factors (Diebold et al.,2006). Our results are also similar indicating that more complex models, in this case including a non-linear activation for the autoencoder does not lead to better results compared to the linear autoencoder. We see a reduced in-sample performance from considering a one-layer model with a non-linear activation function (NA3). When considering non-linear activation performance can only be improved by introducing three layers, we consider this model to be a deep autoencoder (DA3) with the architecture determined by the graph in Figure 2. It is important to recall that these autoencoders are used as dynamic factor models, therefore they use an autoencoder for dimension reduction into the three factors (level, slope and curvature) and then these factors are forecast using a vector autoregression.

For the RNN we obtain multivariate estimates of the maturities using the different architectures explained in section 4. In this case there no intermediate factor estimation and hence we consider two variations of the models with a single layer and the different number of hidden units, twenty or two hundred to be exact. Of course, the large number of hidden units increases the computational complexity of the models.
All of the models that only consider twenty hidden units have a performance that is below the Nelson-Siegel model. However, when we increase the ten times the number of units the improvement is quite substantial showing the best overall performance where all of the errors across the maturities are below 10 basis points.
In summary, the in-sample results show significant improvements of using artificial neural networks in forecasting the yield curve, either as a complement and within the context of dynamic term structure model or using techniques design for sequential/time-series data such as recurrent neural networks. There is however important computational cost in using neural networks as opposed to principal component analysis. It is still important to consider in the next section when looking at an out-of-sample exercise whether the increased forecast performance of a method like LSTM could be the result of overfitting.

6.2. Out-of-sample results

For the out-of-sample forecasting exercise, we consider an initial estimation/training window. We consider eight years of data from the start of the sample from November 1985 up to December 1994, this is about one-fourth

of the historical data (thirty-five years of monthly data). For the out-of-sample evaluation, we fix this eight-year rolling window. There is no clear consensus regarding the benefits of a rolling or an expanding window. In Diebold and Li (2006) and Pooter (2007) they use a expanding window in Ang and Piazzesi (2003) the strategy is not clear. We believe that a rolling window gives the models a better chance of adjusting to structural changes. However, this approach is not common in neural network models that are better trained using a larger percentage of the original sample. The advantage of having the short rolling window is the ability to test how the models adapt to the different U.S. economic conditions during the period and its important effects on the yield curve. This is clear when observing figure 3 where there is a lot of variation along the curve and also within; meaning that there are some parts of the curve that are more volatile (Nguyen et al., 2020) or subject to jump dynamics (Dungey et al., 2009) than others. For economic and financial forecasting it is very important to have a model that can quickly capture structural changes.

Forecast evaluation is performed over the sample period starting in January 1994 and ending in December 2020. Once a piece of new monthly information is included the model is re-estimated. The metric used is the root mean square error (RMSE) expressed in basis points using as forecast horizon 1, 6 and 12 months. As a benchmark we consider a random walk (RW) forecast, $E_t(y_{t+1}(\tau)) = y_t(\tau)$. To determine deviations from the benchmark during the out-of-sample evaluation we use the Cumulative square prediction error (CSPE),

$$CSPE_t = \sum_{i=1}^{t}((\hat{y}_i^{RW}(\tau) - y_i(\tau))^2 - (\hat{y}_i^{model}(\tau) - y_i(\tau))^2)$$

where $\hat{y}_i^{model}$ denotes the forecast from the candidate model.

To perform inference on the forecast and compare the results across models we use the Diebold-Mariano test adjusted for small samples (Harvey et al., 1997) for forecast accuracy where the null hypothesis indicates that the pair of models under consideration have equivalent forecast performance.

For the one-month forecast horizon (Table 3) PCA and the linear autoencoders have similar results as expected, their performances are also superior to that of the Nelson-Siegel model and the non-linear autoencoder. The performance of PCA and the linear autoencoders are statistically equivalent because for most maturities we cannot reject the null hypothesis of the

Diebold-Mariano test (at a significance level of 5%). At maturities below one year, seven and ten years they have similar accuracy to that of the random walk (Table 4). The test also indicates that they outperform the Nelson and Siegel model especially in the short and medium part of the curve. As in the in-sample results, there is no significant benefit in considering deep autoencoders, this result is important because deep models have more layers and are computationally more expensive. Figure 4 compares the evolution of the cumulative error along with the sample for the one-month horizon. The linear autoencoder seems to have errors across most maturities, whereas the Nelson and Siegel model has increasing errors, especially at the longer maturities. The figures also indicate that forecasts performance is also related to the sample period (Pooter, 2007). For example after the financial crisis and throughout the last twelve years, the decay in performance compared to the random walk is much smaller than for the first part of the sample. In other words, the slope becomes less negative and in some maturities it is zero, indicating no difference to the random walk.
For the six and twelve-month forecast horizon (Tables 5, 7) the results are similar regarding the performance across models. The non-linear autoencoders have an overall poor performance. After introducing additional layers to the model and obtaining a deep autoencoder (DA3) the errors are reduced substantially sometimes outperforming the Nelson-Siegel model but not PCA or their linear counterparts. The Diebold-Mariano test show that these longer forecast horizons the advantage of the linear autoencoder over the Nelson-Siegel model are not as consistent across maturities since only for some maturities (1-4 months,4,5,10,15 years) the forecast performance is statistically different (Tables 6,8).

Recurrent neural network models have an important number of hyperparameters associated with the training of the model. We tested different combinations of the following parameters: the batch size which denotes the number of lagged time series that are sequentially used in training the model. For the batch size, we find that the best performing models are those where the batch size is equivalent to the forecast horizon. The epochs denote the number of applications of the forward and backward propagation is used on the estimation window. For the epochs, we tested different sizes from 10 to 10000. The number of epochs considered increases the computational cost of estimating the models and performing the out-of-sample forecast evaluation because the model is re-estimated every time the rolling window changes.

We also consider single layer and three-layer models. Finally, we consider a different number of hidden units, in particular 20 or 200 units. It is important to note that the model considered, ERNN, GRNN and LTSM have different numbers of parameters. From expressions 20 and 8 we can observe that LSTM has the greatest number parameters and GRNN the fewest.
Tables 9, 10, and 11 present the out-of-sample performance for the recurrent network models at the one, six and twelve-month forecast horizon respectively. The forecasting performance of these models quite poor. The average error of the RNN is at least six times that of the dynamic factor models. For example, at the one-month horizon (Table 3) the errors for most models (except for the non-linear autoencoder, NA3) are below 45 basis points, whereas in the RNN (Table 9) all of the errors are above 150 basis points. In addition, more complex models like the LTSM do not show a systematic better performance over the simple ERNN model. In fact, at some maturities, the ERNN model with fewer hidden units (20) shows a better performance than the LSTM with 200 units. One possible explanation is that the long term memory of the LSTM does not let the model quickly adapt to the structural changes observed in the data, in this case, a simpler model like the ERNN might be better suited to adapt to these changes. These preliminary results confirm some of the uncertainties regarding the use of these type of models mentioned when discussing the in-sample results, section 6.1. It is possible to tune the hyperparameters in the model at the training stage (in-sample) to get remarkable results in terms of forecasting accuracy. However, the success of hyperparameter tuning is no guarantee that in a properly defined out-of-sample exercise the forecast accuracy will hold.
The results confirm that simpler factor models like the dynamic factor model that used PCA seem to provide the best performance and a reduced computational effort. This is important because practitioners tend to favour PCA for yield curve modelling over other more complex approaches.

7. Conclusions

In this paper, we explore the forecasting performance of neural networks in term structure models. Dynamic term structure models that do not account for no-arbitrage restrictions have provided a useful tool for forecasting and in a more general sense providing a framework for building the yield curve. New tools and increased computational power used to estimate neural networks has increased the application of these tools in economic and

financial forecasting. Autoencoders are a particular type of neural network, that provide tools for dimension reduction. Using the autoencoders we provide a way to interpret and use these tools as dynamic term structure models and therefore propose a hybrid vector autoregression and neural network approach to forecasting the term structure using a factor model (a state-space representation). As it is well known in the literature principal component is a particular case of an autoencoder. In addition to the autoencoders, we also explore the forecasting performance of recurrent neural networks based on the observed yields as an alternative.
Our empirical results based on 35 years of monthly U.S. yields shows that the latent factors estimated using the autoencoders and forecast using a vector autoregression provide a successful tool for forecasting; this is also true of the principal components. Both approaches out-perform the Nelson-Siegel three-factor model and the more complex models based on recurrent neural networks. Recurrent neural networks suffer from over-fitting especially those that design to accommodate long term trends. Their in-sample sample show significantly lower forecast errors than all of the other models but the out-of-sample performance are poor.

Although these autoencoders provide better RMSE than the Nelson-Siegel model, linear version and in particular PCA has the same performance at a much lower computational cost. PCA provides a quick method to estimate and select a reduced number of factors without losing significant information. In these particular case the well-known level, slope and curvature, obtained from the first three principal components provide the core elements to reconstruct the curve and quickly adjust to structural changes like the ones observed on interest rates in the last ten years in the U.S. data.

References

[1] Shihao Gu, Bryan T. Kelly, and Dacheng Xiu. Autoencoder asset pricing models. *Journal of Econometrics*, 2020. ISSN 1556-5068. doi: doi.org/10.1016/j.jeconom.2020.07.009.

[2] J. B. Heaton and Nick Polson. Deep learning for finance: Deep portfolios. *Appl. Stoch. Models Bus. Ind.*, 2017. ISSN 1556-5068. doi: doi.org/10.1002/asmb.2209.

[3] Francis Diebold and Canlin Li. Forecasting the term structure of government bond yields. *J. Econometrics*, 2006. doi: 10.1016/j.jeconom.2005.03.005.

[4] Robert Engle, Guillaume Roussellet, and Emil Siriwardane. Scenario generation for long run interest rate risk assessment. 201:333–347, 2017. ISSN 0304-4076. doi: 10.1016/j.jeconom.2017.08.012.

[5] Charles R. Nelson and Andrew F. Siegel. Parsimonious modeling of yield curves. *Journal of Business*, 60:473, 1987. ISSN 0021-9398. doi: 10.1086/296409.

[6] Jens H. E. Christensen, Francis X. Diebold, and Glenn D. Rudebusch. The affine arbitrage-free class of nelsonsiegel term structure models. *Journal of Econometrics*, 164:4–20, 2011. ISSN 0304-4076. doi: 10.1016/j.jeconom.2011.02.011.

[7] D. Redfern and D. McLean. Principal component analysis for yield curve modelling: Reproduction of out-of-sample yield curves. Technical report, Moodys Analytics, 2014.

[8] Marcos López de Prado. *Advances in Financial Machine Learning:*. Wiley, 2018. doi: 10.2139/ssrn.3270269.

[9] Matthew F. Dixon, Igor Halperin, and Paul Bilokon. *Machine Learning in Finance: From theory to practice*. Springer, 2020. doi: 10.1007/978-3-030-41068-1.

[10] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37:388–427, 2021. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2020.06.008.

[11] Anastasia Borovykh, Sander Bohte, and Cornelis W. Oosterlee. Conditional time series forecasting with convolutional neural networks. March 2017.

[12] G. P. Zhang and V. L. Berardi. Time series forecasting with neural network ensembles: an application for exchange rate prediction. *Journal of the Operational Research Society*, 52:652–664, 2001. ISSN 0160-5682. doi: 10.1057/palgrave.jors.2601133.

[13] Yoshiyuki Suimon, Hiroki Sakaji, Kiyoshi Izumi, and Hiroyasu Matsushima. Autoencoder-based three-factor model for the yield curve of japanese government bonds and a trading strategy. 13:82, 2020. ISSN 1911-8074. doi: 10.3390/jrfm13040082.

[14] G. Kirczenow, A. Fathi, and M. Davison. Machine learning for yield curve feature extraction: Application to illiquid corporate bonds. June 2018.

[15] Alexei Kondratyev. Learning curve dynamics with artificial neural networks. April 2018.

[16] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2:53–58, 1989.

[17] Lars E. O. Svensson. Estimating and interpreting forward interest rates: Sweden 1992 - 1994. NBER Working Paper Series, 1994. 4871.

[18] Michiel De Pooter. Examining the nelson-siegel class of term structure models: In-sample fit versus out-of-sample forecasting performance. 2007. ISSN 1556-5068. doi: 10.2139/ssrn.992748.

[19] R.J. Hyndman, Koehler A.B., J.K. Ord, and R.D. Snyder. *Forescasting with exponential smoothing: The state space approach*. Springer, 2008.

[20] Martin M. Andreasen, Jens H. E. Christensen, and Glenn D. Rudebusch. Term structure analysis with big data: One-step estimation using bond prices. *Journal of Econometrics*, 212:26–46, 2019. ISSN 0304-4076. doi: 10.1016/j.jeconom.2019.04.019.

[21] Hitesh Doshi, Kris Jacobs, and Rui Liu. Information in the term structure: A forecasting perspective. *Management Science*, 2020. ISSN 0025-1909. doi: 10.1287/mnsc.2020.3715.

[22] Francis Diebold, Glenn Rudebusch, and S. Boragan Aruoba. The macroeconomy and the yield curve: A dynamic latent factor approach. *Journal of Econometrics*, (131):309–338, 2006. doi: 10.3386/w10616.

[23] Andrew Ang and Monika Piazzesi. A no-arbitrage vector autoregression of term structure dynamics with macroeconomic and latent variables.

Journal of Monetary Economics, 50:745–787, 2003. ISSN 0304-3932. doi: 10.1016/s0304-3932(03)00032-1.

[24] Giang Nguyen, Robert Engle, Michael Fleming, and Eric Ghysels. Liquidity and volatility in the u.s. treasury market. *Journal of Econometrics*, 217:207–229, 2020. ISSN 0304-4076. doi: 10.1016/j.jeconom.2019.12.002.

[25] Mardi Dungey, Michael McKenzie, and L. Vanessa Smith. Empirical evidence on jumps in the term structure of the us treasury market. *Journal of Empirical Finance*, 16:430–445, 2009. ISSN 0927-5398. doi: 10.1016/j.jempfin.2008.12.002.

[26] David Harvey, Stephen Leybourne, and Paul Newbold. Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, 13:281–291, 1997. ISSN 0169-2070. doi: 10.1016/s0169-2070(96)00719-4.

[27] F. Van Veen and S. Leijnen. The neural network zoo, 2019. URL https://www.asimovinstitute.org/neural-network-zoo.

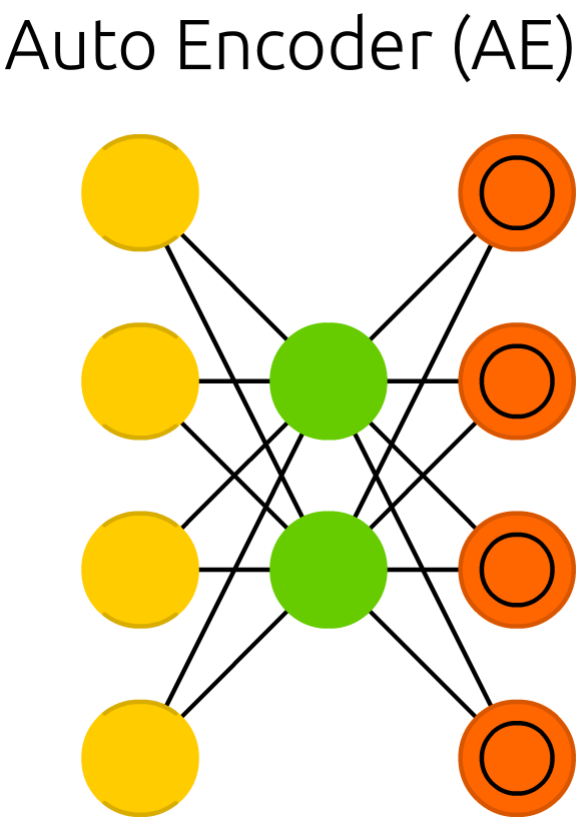


Figure 1: Simple autoencoder (Van Veen and Leijnen, 2019)

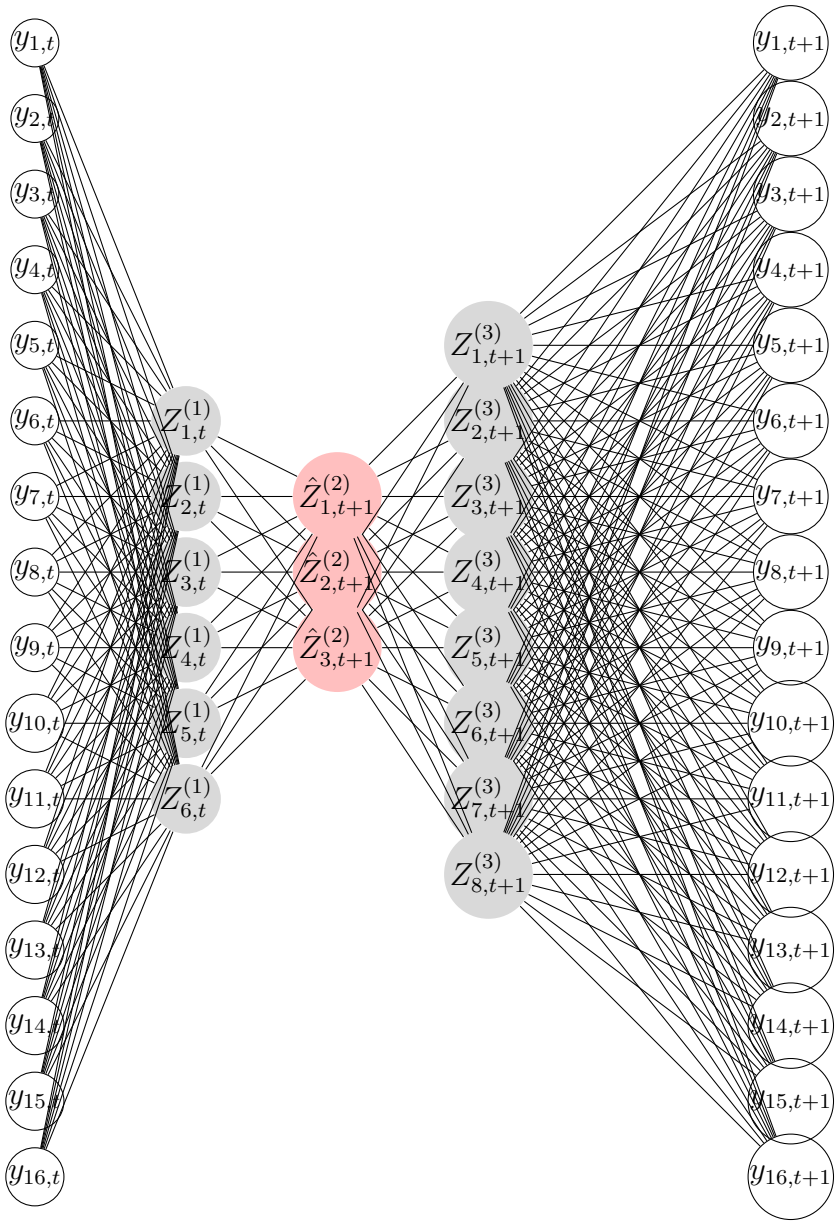


Figure 2: Deep autoencoder

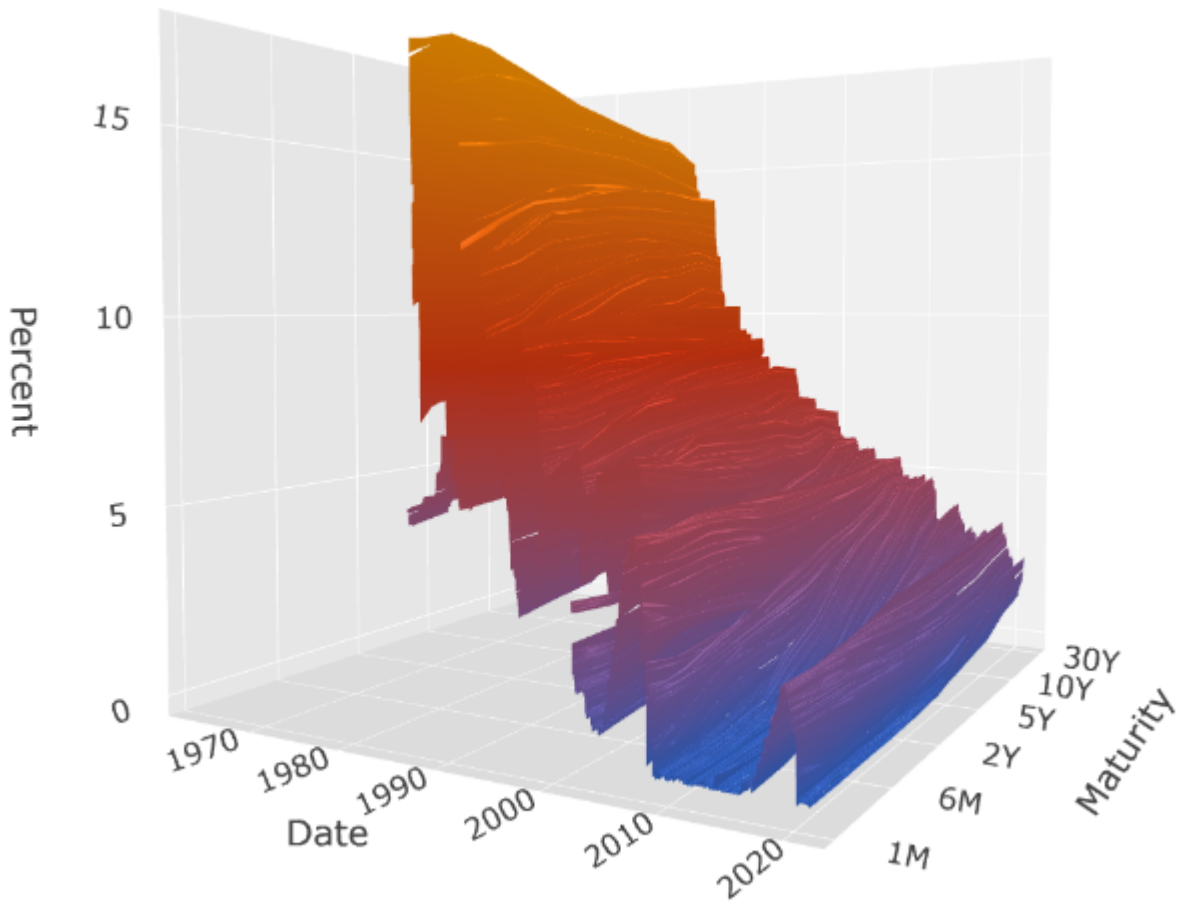
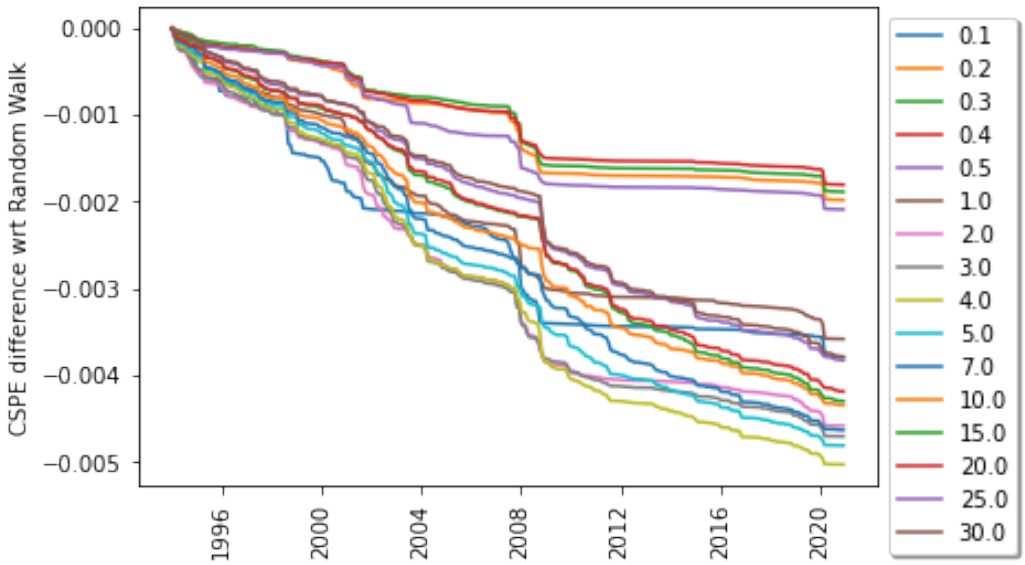
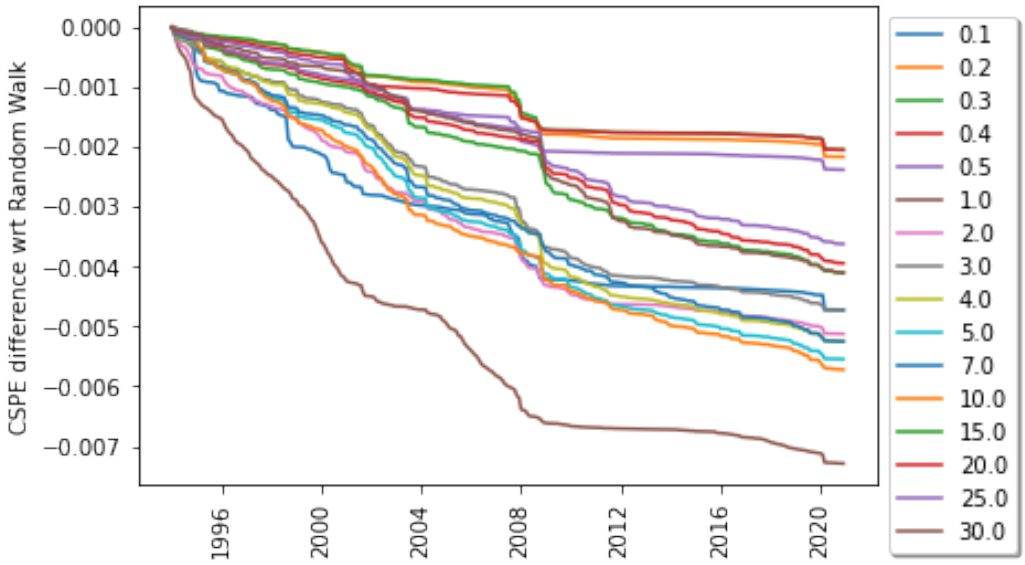


Figure 3: The figure shows a three dimensional plot of the panel of monthly U.S. zero coupon yields from November 1985 to December 2020.



(a) Linear autoencoder



(b) Nelson-Siegel

Figure 4: Cumulative square prediction error compared to the random walk, forecast horizon 1 month.

Maturity	NS3	PCA	LA3	NA3	DA3
0.1	17.78	12.50	12.62	19.97	17.25
0.2	6.78	4.72	4.73	11.344	9.98
0.3	5.92	4.05	4.07	9.76	8.14
0.4	7.53	4.45	4.48	8.68	7.55
0.5	10.39	5.55	5.55	8.70	7.43
1.0	38.93	8.59	8.64	10.99	9.70
2.0	20.03	5.56	5.54	13.18	10.72
3.0	7.15	3.98	4.02	13.08	11.11
4.0	5.99	5.56	5.57	13.06	11.24
5.0	11.86	5.86	5.91	12.24	10.72
7.0	11.76	5.99	5.98	10.09	8.22
10.0	16.63	5.75	5.75	7.46	5.87
15.0	9.98	5.29	5.52	9.82	7.97
20.0	8.37	8.13	8.17	11.45	8.58
25.0	10.96	5.70	5.71	11.43	10.36
30.0	9.05	7.37	7.48	11.44	10.92

Table 1: The table shows the in-sample fit based on the RMSE statistic for the full sample (1985:11-2020:12). The statistics are expressed in basis points (10 basis points is 0.10%). The statistic is given for the three factor Nelson-Siegel model (NS3), principal components (PCA), linear autoencoder (LA3), non-linear autoencoder (NA3) and a deep autoencoder (DA3).

Maturity	ERNN20	GRNN20	LSTM20	ERNN200	GRNN200	LSTM200
0.1	25.59	24.93	23.38	8.37	12.54	3.76
0.2	15.77	12.91	11.86	5.46	6.25	3.38
0.3	15.87	11.40	10.99	5.17	5.88	3.14
0.4	16.07	10.54	10.71	4.64	6.36	3.51
0.5	17.31	11.18	11.57	5.67	6.84	3.03
1.0	21.24	12.81	13.77	6.20	7.60	3.80
2.0	21.30	13.62	14.03	5.83	6.82	3.65
3.0	21.52	13.66	13.65	5.13	5.94	3.53
4.0	21.94	14.90	13.40	5.03	6.18	3.14
5.0	21.62	14.47	13.15	5.08	6.23	3.09
7.0	21.44	14.83	12.47	4.16	5.62	2.96
10.0	21.33	17.28	12.37	4.95	5.98	2.95
15.0	20.85	13.83	12.52	4.95	5.89	2.97
20.0	21.22	13.24	12.12	5.15	6.04	3.01
25.0	19.80	14.80	12.25	5.04	6.51	3.11
30.0	19.78	19.01	11.95	5.10	6.56	2.82

Table 2: The table shows the in-sample fit based on the RMSE statistic for the full sample (1985:11-2020:12). The statistics are expressed in basis points (10 basis points is 0.10%). The statistic is given for simple recurrent neural network (ERNN), gated recurrent network (GRNN) and the Long-Short Memory model(LSTM). All of the models are estimated using 20 or 200 hidden units.

Maturity	RW	NS3	PCA	LA3	NA3	DA3
0.1	25.63	28.28	22.25	22.65	36.44	24.25
0.2	17.28	19.25	17.33	17.71	40.66	19.66
0.3	16.69	18.80	17.02	17.44	48.78	19.12
0.4	16.29	19.17	16.89	17.09	55.09	18.75
0.5	17.79	20.47	17.56	18.14	59.96	19.61
1.0	21.57	42.23	24.42	25.30	97.89	26.10
2.0	24.12	31.61	27.95	28.85	126.31	29.23
3.0	25.63	28.34	27.69	28.19	131.61	29.17
4.0	26.84	29.96	28.41	28.83	136.09	29.92
5.0	26.52	31.74	27.66	27.95	139.56	29.53
7.0	26.28	30.49	27.23	27.17	140.50	29.95
10.0	25.55	33.35	26.04	26.23	146.72	29.14
15.0	24.10	26.19	26.89	27.31	131.67	31.39
20.0	23.54	25.74	26.46	27.16	131.03	29.92
25.0	22.27	24.94	25.37	26.18	128.60	31.32
30.0	22.47	27.58	24.83	25.76	133.91	31.15

Table 3: The table shows the out-of-sample perfomance of the RMSE statistic for 1 month forecast horizon. The statistics are expressed in basis points (10 basis points is 0.10%). The statistic is given for the three factor Nelson-Siegel model (NS3), principal components (PCA), linear autoencoder (LA3), non-linear autoencoder (NA3) and a deep autoencoder (DA3).

Maturity	PCA/LA3	RW/LA3	NS/LA3	DA3/LA3	RW/DA3
0.1	0.454	0.071	0.000	0.054	0.413
0.2	0.091	0.588	0.000	0.007	0.010
0.3	0.045	0.292	0.000	0.032	0.008
0.4	0.367	0.213	0.000	0.035	0.003
0.5	0.023	0.686	0.000	0.062	0.072
1.0	0.008	0.000	0.000	0.382	0.000
2.0	0.007	0.000	0.006	0.675	0.000
3.0	0.101	0.000	0.768	0.165	0.000
4.0	0.058	0.010	0.031	0.187	0.000
5.0	0.100	0.003	0.000	0.056	0.001
7.0	0.639	0.173	0.000	0.006	0.003
10.0	0.152	0.222	0.000	0.023	0.012
15.0	0.083	0.000	0.255	0.006	0.000
20.0	0.038	0.000	0.027	0.026	0.000
25.0	0.091	0.000	0.152	0.001	0.000
30.0	0.030	0.001	0.050	0.001	0.000

Table 4: The table shows the p-values from the Diebold-Mariano test for forecast accuracy at the one month horizon. The statistic compares the accuracy between pair of models.

Maturity	RW	NS3	PCA	LA3	NA3	DA3
0.1	63.52	69.06	58.66	59.57	1.81e+08	60.34
0.2	61.61	68.10	62.13	62.72	2.49e+08	62.66
0.3	61.52	70.30	65.32	65.70	1.42e+08	65.38
0.4	61.67	71.69	67.01	67.40	2.47e+07	66.92
0.5	62.32	72.96	68.55	69.05	6.56e+07	67.97
1.0	74.81	82.18	87.88	88.53	5.61e+08	86.97
2.0	74.51	84.00	88.58	89.00	1.15e+09	87.90
3.0	73.01	84.73	84.46	85.08	1.30e+09	84.74
4.0	72.11	86.15	82.22	82.53	1.40e+09	82.73
5.0	70.62	86.87	80.39	80.68	1.51e+09	81.23
7.0	66.70	80.50	76.10	76.43	1.52e+09	77.71
10.0	64.15	79.95	72.77	73.03	1.55e+09	74.86
15.0	57.50	63.68	69.31	69.67	1.50e+09	71.59
20.0	56.59	64.02	67.82	68.32	1.43e+09	69.78
25.0	53.95	60.65	65.11	65.81	1.39e+09	68.32
30.0	55.00	64.68	63.25	63.88	1.44e+09	67.52

Table 5: The table shows the out-of-sample perfomance of the RMSE statistic for 6 month forecast horizon. The statistics are expressed in basis points (10 basis points is 0.10%). The statistic is given for the three factor Nelson-Siegel model (NS3), principal components (PCA), linear autoencoder (LA3), non-linear autoencoder (NA3) and a deep autoencoder (DA3).

Maturity	PCA/LA3	RW/LA3	NS/LA3	DA3/LA3	RW/DA3
0.1	0.035	0.413	0.000	0.697	0.473
0.2	0.242	0.820	0.000	0.977	0.815
0.3	0.464	0.387	0.001	0.892	0.378
0.4	0.484	0.232	0.013	0.841	0.229
0.5	0.422	0.160	0.083	0.674	0.197
1.0	0.363	0.023	0.281	0.615	0.028
2.0	0.496	0.012	0.062	0.701	0.008
3.0	0.079	0.035	0.692	0.895	0.011
4.0	0.266	0.066	0.001	0.939	0.016
5.0	0.213	0.077	0.005	0.832	0.020
7.0	0.161	0.088	0.054	0.593	0.020
10.0	0.099	0.116	0.006	0.486	0.031
15.0	0.030	0.018	0.011	0.483	0.003
20.0	0.013	0.013	0.047	0.594	0.001
25.0	0.026	0.019	0.091	0.406	0.003
30.0	0.007	0.038	0.735	0.200	0.008

Table 6: The table shows the p-values from the Diebold-Mariano test for forecast accuracy at the six month horizon. The statistic compares the accuracy between pair of models.

Maturity	RW	NS3	PCA	LA3	NA3	DA3
0.1	104.54	124.07	108.36	109.29	1.31e+10	104.82
0.2	105.63	125.54	115.59	116.29	1.62e+10	109.81
0.3	105.55	128.01	119.61	120.31	1.81e+10	112.75
0.4	105.25	129.26	121.68	122.41	1.96e+10	114.40
0.5	105.60	130.50	123.59	124.45	2.06e+10	115.64
1.0	121.92	136.52	153.04	153.64	2.90e+10	141.31
2.0	112.28	137.22	146.94	147.39	3.21e+10	135.32
3.0	104.17	135.27	136.33	136.77	3.12e+10	127.64
4.0	98.44	132.60	128.05	128.25	3.04e+10	121.50
5.0	94.46	129.92	122.07	122.36	2.99e+10	116.97
7.0	88.33	118.76	113.38	113.66	2.89e+10	111.58
10.0	83.35	113.12	106.01	106.60	2.88e+10	107.47
15.0	74.42	89.98	98.29	98.85	2.47e+10	103.73
20.0	72.85	88.58	94.81	95.56	2.44e+10	100.56
25.0	68.90	83.52	91.36	92.19	2.33e+10	99.63
30.0	69.63	87.85	88.41	89.45	2.43e+10	97.16

Table 7: The table shows the out-of-sample perfomance of the RMSE statistic for 12 month forecast horizon. The statistics are expressed in basis points (10 basis points is 0.10%). The statistic is given for the three factor Nelson-Siegel model (NS3), principal components (PCA), linear autoencoder (LA3), non-linear autoencoder (NA3) and a deep autoencoder (DA3).

Maturity	PCA/LA3	RW/LA3	NS/LA3	DA3/LA3	RW/DA3
0.1	0.083	0.545	0.002	0.438	0.294
0.2	0.327	0.226	0.001	0.478	0.192
0.3	0.340	0.131	0.001	0.558	0.171
0.4	0.329	0.106	0.009	0.590	0.168
0.5	0.266	0.093	0.056	0.722	0.161
1.0	0.508	0.060	0.056	0.720	0.140
2.0	0.509	0.069	0.051	0.638	0.147
3.0	0.453	0.085	0.537	0.471	0.154
4.0	0.660	0.103	0.020	0.372	0.175
5.0	0.585	0.120	0.007	0.331	0.183
7.0	0.630	0.129	0.113	0.290	0.193
10.0	0.424	0.147	0.028	0.275	0.210
15.0	0.318	0.054	0.046	0.277	0.174
20.0	0.215	0.062	0.160	0.281	0.190
25.0	0.169	0.049	0.142	0.262	0.173
30.0	0.125	0.061	0.742	0.268	0.194

Table 8: The table shows the p-values from the Diebold-Mariano test for forecast accuracy at the twelve month horizon. The statistic compares the accuracy between pair of models.

Maturity	ERNN20	GRNN20	LSTM20	ERNN200	GRNN200	LSTM200
0.1	173.52	171.96	171.96	185.83	176.92	177.19
0.2	185.93	194.31	182.31	191.18	185.40	186.52
0.3	191.14	200.68	185.58	191.67	188.61	188.67
0.4	193.32	208.67	192.57	192.33	190.77	191.06
0.5	191.83	208.40	192.61	192.74	191.67	192.01
1.0	244.05	273.30	246.16	236.09	238.86	237.81
2.0	257.33	281.97	256.88	246.32	253.73	251.10
3.0	253.63	266.66	253.84	250.62	251.66	252.00
4.0	261.19	275.55	264.36	253.40	260.20	259.35
5.0	260.69	268.47	257.13	254.38	260.13	259.75
7.0	260.56	272.74	257.86	254.20	258.63	259.32
10.0	255.46	255.79	247.35	245.18	250.81	252.00
15.0	255.57	291.27	253.35	252.04	254.43	256.53
20.0	257.90	242.50	254.61	239.78	254.95	254.52
25.0	256.35	275.83	256.27	254.92	255.84	257.45
30.0	236.37	232.55	235.30	236.65	236.36	236.97

Table 9: The table shows the out-of-sample perfomance of the RMSE statistic for 1 month forecast horizon. The statistics are expressed in basis points (10 basis points is 0.10%). The statistic is given for simple recurrent neural network (ERNN), gated recurrent network (GRNN) and the Long-Short Memory model(LSTM). All of the models are estimated using 20 or 200 hidden units.

Maturity	ERNN20	GRNN20	LSTM20	ERNN200	GRNN200	LSTM200
0.1	187.38	213.83	183.82	185.17	176.81	177.18
0.2	197.26	188.47	205.77	192.41	186.48	186.52
0.3	199.21	191.40	211.11	192.78	188.44	189.21
0.4	200.17	193.02	212.70	192.69	190.32	190.43
0.5	201.71	192.71	204.08	193.23	191.40	191.22
1.0	245.81	247.05	257.22	234.44	235.64	234.45
2.0	258.37	266.79	278.80	249.62	247.31	244.45
3.0	267.66	280.24	300.52	262.10	251.23	247.76
4.0	274.04	289.68	318.31	271.25	251.57	248.66
5.0	276.21	294.68	322.50	274.53	252.58	249.74
7.0	281.29	299.17	335.28	280.81	252.28	248.96
10.0	271.72	290.73	327.99	271.45	244.38	242.55
15.0	282.33	295.57	330.00	283.06	250.36	247.93
20.0	260.92	278.90	311.38	260.20	239.91	236.18
25.0	283.11	296.14	329.62	285.06	253.26	251.15
30.0	267.93	282.76	326.19	272.91	235.59	234.57

Table 10: The table shows the out-of-sample perfomance of the RMSE statistic for 6 month forecast horizon. The statistics are expressed in basis points (10 basis points is 0.10%). The statistic is given for simple recurrent neural network (ERNN), gated recurrent network (GRNN) and the Long-Short Memory model(LSTM). All of the models are estimated using 20 or 200 hidden units.

Maturity	ERNN20	GRNN20	LSTM20	ERNN200	GRNN200	LSTM200
0.1	184.54	183.31	186.75	181.02	174.59	175.97
0.2	195.68	189.01	189.07	191.94	186.09	186.15
0.3	197.96	189.89	191.68	195.15	188.21	188.56
0.4	199.59	191.21	193.44	197.22	189.76	189.89
0.5	200.31	192.18	194.66	197.94	190.47	191.15
1.0	249.20	236.17	245.08	248.42	236.11	235.59
2.0	272.03	248.93	279.63	272.23	252.42	249.77
3.0	289.57	258.13	303.08	290.93	263.35	259.32
4.0	304.01	264.04	320.33	307.27	271.20	265.43
5.0	309.17	267.43	327.12	313.31	276.42	269.89
7.0	320.54	273.49	339.97	326.53	282.24	275.16
10.0	313.10	265.47	332.64	320.77	274.28	267.61
15.0	325.05	277.91	341.09	333.41	285.53	279.11
20.0	299.85	255.21	316.22	307.94	264.03	259.02
25.0	323.34	279.90	337.39	331.59	288.35	282.18
30.0	315.86	267.82	330.43	323.54	278.27	270.66

Table 11: The table shows the out-of-sample perfomance of the RMSE statistic for 12 month forecast horizon. The statistics are expressed in basis points (10 basis points is 0.10%). The statistic is given for simple recurrent neural network (ERNN), gated recurrent network (GRNN) and the Long-Short Memory model(LSTM). All of the models are estimated using 20 or 200 hidden units.